Contribution to the improvement of meshless methods applied to continuum mechanics

Gabriel Fougeron

université PARIS-SACLAY

CentraleSupélec

NNT: 2018SACLC068

Thèse de doctorat

Contribution to the improvement of meshless methods applied to continuum mechanics

Thèse de doctorat de l'Université Paris-Saclay
préparée à CentraleSupélec

Ecole doctorale n°579 Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)
Spécialité de doctorat: Mécanique des matériaux

Thèse présentée et soutenue à Gif-sur-Yvette, le 2 Octobre 2018, par

Gabriel Fougeron

Composition du Jury:

Francisco Chinesta Arts et Métiers ParisTech (PIMM UMR CNRS 8006)	Président
Piotr Breitkopf Université de Technologie de Compiègne (Roberval UMR CNRS 6066)	Rapporteur
Nicolas Moës École Centrale de Nantes (Gem UMR CNRS 6183)	Rapporteur
Ludovic Chamoin École Normale Supérieure Paris-Saclay (LMT UMR CNRS 8535)	Examinateur
Denis Aubry École CentraleSupélec (MssMat UMR CNRS 8579)	Directeur de thèse
Guillaume Pierrot Icon CFD	Co-directeur de thèse
Anargiros Kamoulakos ESI Group	Co-directeur de thèse

À Tristan et Raphaëlle,
mes anciens co-locataires.

Remerciements

Ce travail a été conjointement financé par ESI Group et par l'Association Nationale de la Recherche et de la Technologie dans le cadre d'une Convention Industrielle de Formation par la REcherche portant la référence 2015/0280.

En tout premier lieu, j'aimerais remercier les personnes qui, au sein d'ESI Group, ont eu la charge de définir, financer et encadrer mes travaux de thèse. Anargiros Kamoulakos m'a accueilli dès 2014 au sein de son équipe de recherche, le CoE Techno. Il a toujours soutenu la tournure très théorique qu'ont rapidement pris mes travaux, en les inscrivant dans une stratégie à long terme de développement des solutions sans maillage d'ESI Group. Cet engagement se manifeste aussi par mon embauche récente au sein de la même équipe dont l'objectif principal est de traduire les contributions de ces quatre années de recherche en un produit industriel.

Rien de ce projet n'aurait été possible sans Guillaume Pierrot, qui m'a pris sous son aile dès la fin 2013 pendant ma troisième année d'études à l'École Centrale Paris. C'est réellement lui qui m'a fait découvrir le monde de la simulation numérique, ce sont ses idées qui forment la base du présent travail, et c'est sa vision que je m'applique à faire perdurer. Je suis extrêmement reconnaissant tant à Argiris qu'à Guillaume pour m'avoir laissé une liberté quasi-totale pendant la préparation de cette thèse, pour la bienveillance de leur encadrement, ainsi que pour leur sens de la loyauté.

Je tiens aussi à remercier Denis Aubry, professeur désormais émérite du laboratoire MSSMat. J'ai appris récemment que nos discussions hebdomadaires pouvaient dérouter les étudiants travaillant aux alentours, tant la passion du débat – dont le seul but était parfois le débat lui-même – savait nous emporter. Son expérience de l'encadrement m'a été précieuse, en particulier lors de la structuration et de mon travail dans le présent manuscrit, et surtout pour sa mise en valeur le jour de la soutenance.

Je remercie vivement Piotr Breitkopf et Nicolas Moës d'avoir volontiers accepté de lire le présent manuscrit et de rédiger un rapport à son sujet. Je remercie également Ludovic Chamoin d'avoir accepté de prendre part à mon jury de thèse, et à Francisco Chinesta de l'avoir présidé.

Je souhaite encore remercier ceux qui ont partagé mon quotidien à ESI Group: Georges Eleftheriou, Fatima-Zohra Daïm, Benoit Laussat et Mattia Oriani. Je réserve une mention spéciale à ce dernier en souvenir de quatre années de déjeuners partagés.

J'adresse également une pensée amicale à mes compagnons d'infortune du laboratoire MSSMat, parmis eux Yu Liu, Michael Callahan, Matthieu Bonneric, Gilles Lenoir, Solenne Mondésert et Hélène Moustacas. Ils ont su trouver les mots du réconfort dans mes quelques moments de doute, et n'ont jamais rechigné à me guider dans les méandres administratifs de l'école doctorale.

Je remercie enfin ma famille, en particulier mes parents, mes sœurs Claire et Blandine, mes (par anticipation pour l'un) beaux-frères Pierre et Baudouin, et mes neveux Tristan, Raphaëlle et Marin pour leur soutien et leur patience. Je suis aussi reconnaissant de l'intérêt inattendu qu'ont pris mon oncle Étienne et ma tante Marie dans mes travaux de recherche, ayant eux-mêmes réclamé la tâche de relire et corriger (en un temps record !) mes publications.

Résumé

Cette thèse présente un cadre général pour l'étude de schémas de discrétisation nodaux sans maillage formulé en termes d'opérateurs discrets définis sur un nuage de points : intégration volumique et de bord, gradient et opérateur de reconstruction. Ces définitions dotent le nuage de points d'une structure plus faible que celle définie par un maillage, mais partageant avec elle certain concepts fondamentaux. Le plus important d'entre eux est la condition de compatibilité intégro-différentielle. Avec la consistance linéaire du gradient discret, cet analogue discret de la formule de Stokes constitue une condition nécessaire à la consistance linéaire des opérateurs elliptiques en formulation faible. Sa vérification, au moins de manière approchée, permet d'écrire des discrétisations dont le taux de convergence est optimal. La construction d'opérateurs discrets compatibles est si difficile que nous conjecturons – sans parvenir à le démontrer – qu'elle nécessite soit la résolution d'un système linéaire global, soit la construction d'un maillage : c'est "la malédiction sans-maillage". Trois grandes approches pour la construction d'opérateurs discrets compatibles sont étudiées. Premièrement, nous proposons une méthode de correction permettant de calculer l'opérateur gradient compatible le plus proche – au sens des moindres carrés – sans mettre à mal la consistance linéaire. Dans le cas particulier des gradients DMLS, nous montrons que le gradient corrigé est en réalité globalement optimal. Deuxièmement, nous adaptons l'approche SFEM au cadre opérateur et constatons qu'elle définit des opérateurs consistants à l'ordre un et compatibles. Nous proposons une méthode d'intégration discrète exploitant la relation topologique entre les cellules et les faces d'un maillage qui préserve ces caractéristiques. Troisièmement, nous montrons qu'il est possible de générer tous les opérateurs sans maillage rien qu'avec la donnée d'une formule d'intégration volumique nodale en exploitant la dépendance fonctionnelle des poids volumiques nodaux par rapport à la position des nœuds du nuage, l'espace continu sous-jacent et le nombre de nœuds. Les notions de consistance des différents opérateurs sont caractérisées en termes des poids volumiques initiaux, formant un jeu de recommandation pour la mise au point de bonnes formules d'intégration. Dans ce cadre, nous réinterprétons les méthodes classiques de stabilisation de la communauté SPH comme cherchant à annuler l'erreur sur la formule de Stokes discrète. L'exemple des opérateurs SFEM trouve un équivalent en formulation volume, ainsi que la méthode d'intégration discrète s'appuyant sur un maillage. Son écriture nécessite toutefois une description très précise de la géométrie des cellules du maillage, en particulier dans le cas où les faces ne sont pas planes. Nous menons donc à bien une caractérisation complète de la forme de telles cellules uniquement en fonction de la position des nœuds du maillage et des relations topologiques entre les cellules, permettant une définition sans ambigüité de leur volume et centre de gravité. Enfin, nous décrivons des schémas de discrétisation d'équations elliptiques utilisant les opérateurs sans-maillage et proposons plusieurs possibilités pour traiter les conditions au bord tout en imposant le moins de contraintes sur la position des nœuds du nuage de points. Nous donnons des résultats numériques confirmant l'importance capitale de vérifier les conditions de compatibilité, au moins de manière approchée. Cette simple recommandation permet dans tous les cas d'obtenir des discrétisations dont le taux de convergence est optimal.

Abstract

This thesis introduces a general framework for the study of nodal meshless discretization schemes. Its fundamental objects are the discrete operators defined on a point cloud : volume and boundary integration, discrete gradient and reconstruction operator. These definitions endow the point cloud with a weaker structure than that defined by a mesh, but share several fundamental concepts with it, the most important of them being integration-differentiation compatibility. Along with linear consistency of the discrete gradient, this discrete analogue of Stokes's formula is a necessary condition to the linear consistency of weakly discretized elliptic operators. Its satisfaction, at least in an approximate fashion, yields optimally convergent discretizations. However, building compatible discrete operators is so difficult that we conjecture – without managing to prove it – that it either requires to solve a global linear system, or to build a mesh. We dub this conjecture the "meshless curse". Three main approaches for the construction of discrete meshless operators are studied. Firstly, we propose a correction method seeking the closest compatible gradient – in the least squares sense – that does not hurt linear consistency. In the special case of MLS gradients, we show that the corrected gradient is globally optimal. Secondly, we adapt the SFEM approach to our meshless framework and notice that it defines first order consistent compatible operators. We propose a discrete integration method exploiting the topological relation between cells and faces of a mesh preserving these characteristics. Thirdly, we show that it is possible to generate each of the meshless operators from a nodal discrete volume integration formula. This is made possible with the exploitation of the functional dependency of nodal volume weights with respect to node positions, the continuous underlying space and the total number of nodes. Consistency of the operators is characterized in terms of the initial volume weights, effectively constituting guidelines for the design of proper integration formulae. In this framework, we re-interpret the classical stabilization methods of the SPH community as actually seeking to cancel the error on the discrete version of Stokes's formula. The example of SFEM operators has a volume-based equivalent, and so does its discrete mesh-based integration. Actually computing it requires a very precise description of the geometry of cells of the mesh, in particular in the case where its faces are not planar. We thus fully characterize the shape of such cells, only as a function of nodes of the mesh and topological relations between cells, allowing unambiguous definition of their volumes and centroids. Finally, we describe meshless discretization schemes of elliptic partial differential equations. We propose several alternatives for the treatment of boundary conditions with the concern of imposing as few constraints on nodes of the point cloud as possible. We give numerical results confirming the crucial importance of verifying the compatibility conditions, at least in an approximate fashion. This simple guideline systematically allows the recovery of optimal convergence rates of the studied discretizations.

Contents

Introduction

Industrial motivations behind the project

ESI Group defines itself as the leading innovator in virtual prototyping softwares and services. And indeed, this company specialized in material physics has over the years developed a unique proficiency in helping industrial manufacturers replace physical prototypes by virtually replicating the fabrication, assembly and performance testing of products (with an "as manufactured as tested" philosophy) in different environments, including extreme and transient dynamics. Hence, the simulation of the manufacturing processes used in the aerospace, automotive and rail transportation industries – like stamping, forming and welding – is a core part of ESI Group's activity. The class of associated "finishing" processes – like trimming, riveting, milling, drilling and clinching – pose very special demands in terms of predictive simulation capabilities.

Even though the Finite Element Method (FEM) based numerical algorithms are built upon strong theoretical foundations and have over the years had time to mature, these methods still exhibit serious accuracy and stability weaknesses as soon as the regime of application involves excessive local deformations, including topological changes (transition to multi-connectedness, *etc* ...). This is mainly due to the degradation of the Jacobian transformation / mapping between the "parent" state and the "real" state of the elements in question inherent to the topological requirement of a fixed inter-element connectivity, fundamental to the method. Local mesh repair in the form of rezoning / morphing or remeshing is in general at best extremely cumbersome (often necessitating human intervention) and at worst impossible. Additionally, simulation results most often exhibit a high dependency with respect to the geometry of the mesh, lowering the confidence level in the accuracy of the results.

Further on, the transition from a simply to a multi-connected domain, which is the outcome of fracture, involves the creation of a new surface within the continuum, a process that is extremely difficult to treat in a classical FEM approach. Indeed, it necessitates either sophisticated "element-splitting"-class algorithms that are extremely tedious to generalize in a robust way for industrial 3D applications, or very complex and CPU costly "multi-scale" approaches (global-local). A practical alternative is to use "element elimination", a process that is extremely robust and widely used in Virtual Performance Solution (ESI Group's leading software) and other competitor softwares. However, not only does this process require excessively fine meshes, its predominant drawback is to removes material from the simulation model, jeopardizing the fidelity of the physical modeling and with it the residual stress prediction.

One of the most natural ideas to alleviate these difficulties in the simulation of demanding manufacturing processes consists in relaxing the fixed connectivity constraint and use a meshless method instead of a more classical FEM approach. This was the driving principle behind the first introduction of the Smooth Particle Hydrodynamics (SPH) method within

the PAM-SHOCK code of ESI Group as soon as 1998 for hypervelocity impact simulations of space debris upon the double shield of the European Module of the International Space Station (see [124, 125, 126]).

The promising capability revealed by this method regarding the simulation of strongly hydrodynamic processes involving extreme material strains and associated phase changes led to its further development in modelling Fluid-Structure Interaction (FSI) phenomena like tank sloshing, birdstrike and aircraft ditching. It was then further developed with options pertaining to naval engineering and in particular large ship flotation in heavy seas, wave slamming and coastal security assessment under tsunami events. However, its application in the defense sector in the domain of ultra compressible gas dynamics (explosions, ...) revealed serious weaknesses in the accuracy prediction of the associated pressure waves. This gave rise to a more specialized variant of the method called the Finite Pointset Method (FPM), a method that solves the strong form of the associated equations (as opposed to the weak form traditionally used by the SPH method). Finally similar limitations were discovered by the SPH method when applied on quasi-static to mildly dynamic solicitations where once more the accuracy of the full stress tensor (hydrostatic and deviatoric) was put into question.

One other fundamental difficulty with the classical SPH method is the correct representation of an interior Neumann boundary condition which is crucial in the predictive simulation of fracture as we do not a priori know the kinematics of the crack surface (where it opens, in what direction and how much). The ability to incorporate a proper dynamic description of such “free surfaces” is one of the missing ingredients to allow the appearance of the crack to appear and evolve according to the material constitutive law and hopefully not according to the details of the discretization structure (the underlying mesh in the case of the classical FEM).

These limitations are the foundation of the motivation for this PhD thesis: our aim is to provide significant advances in the representation of the stress state by SPH-like methods and tackle the so-called tension instability phenomenon identified by Swegle et al. [220]. To that end, we concentrate on the consistency of variational formulations, which amounts to simultaneously satisfy a linear "primal" consistency condition as well as a "dual" conservation property which takes the form of a discrete version of Stokes's formula.

In the next section, we give a quick literature review of how these problems have been identified and approached in the past.

Historical overview of meshless methods

Meshless methods for the simulation of Partial Differential Equations (PDEs) are defined in contrast with mesh-based methods: the term "meshless" refers to any numerical method designed to approach the solution of a PDE without resorting to any kind of mesh structure. Going further, the real aspiration of the field is to be able to define flexible self-reorganizing structures, *i.e.* computation methods that allow the dynamic restructuring of communication between the discretization entities. This definition tolerates gradation: since the mesh serves several purposes in the modeling and simulation, it might be set aside for one function (*e.g.* integration), and kept for another. Hence, some methods are "more meshless" than others.

Forsaking the rigid structure borne by a mesh has several major consequences on the meshless discretization principles. First of all, the geometric definition of the computational domain is traditionally assumed given at the discrete level by the mesh. Obviously, meshless methods need a substitute object to represent geometry. Secondly, the degrees of freedom may no longer be attached to well-defined regions of space (which would amount to building

a tessellation of space, *i.e.* a mesh). As a substitute, the most common practice is to attach data to nodes of a point cloud. Each node is most often not responsible for a well-defined sub-domain, but rather a fuzzy, blurred area in its vicinity. The very notions of "vicinity" and "neighborhood" of a node can no longer be inherited from the definition of cells in a mesh, they need to be explicitly provided by the meshless methods, for instance with the definition of a smoothing length. The third valuable aspect of meshes for which we need to find an equivalent is more subtle. The organization of cells within a mesh is itself sufficient to define a locally oriented abstract cell complex structure whose co-homology determines a differentiation operator. This operator bears in its very definition the main property mesh-based methods strive for: compatibility. This discrete analogue of Stokes's formula is essential to the good behavior of simulation methods, and its verification for meshless methods is notoriously difficult. All these reasons can – and indeed have in the past – justify the use of meshes even in so-called meshless methods.

The history of meshless methods may roughly be divided into three phases. At their very debut (≈1977-1995), meshless methods were applied to very difficult multi-physics problems such as those occurring in astrophysics. Methods such as Smoothed Particle Hydrodynamics (SPH) were among the first to tackle such complex problems (see [161] for a review article of SPH theory and application during this period), at a time when most other methods failed to even provide an answer. During this period, several *approximation* methods were adapted into *simulation* methods, with special emphasis on either conservation or consistency. For instance, the adaptation of the Kernel Density Estimation ideas and the definition of an approximate delta function gave birth to SPH, the theory of reproducing kernel Hilbert spaces translated into the Reproducing Kernel Particle Methods (RKPM) of Liu and Zhang (see [151]) and Moving Least Squares (MLS) approximation were used as a basis in weak Galerkin formulations yielding the Element-Free Galerkin (EFG) method (see [27]). This period is very well detailed in the review article by Belytschko et al. [26] and the more recent book by Li and Liu [141].

It is only during a subsequent consolidation phase (≈1995-2010) that the two main challenges of meshless methods were really identified. First, the imposition of boundary conditions is particularly difficult in meshless methods (compared to mesh-based methods), mainly because of the difficulties related to the incorporation of the geometrical modeling of the continuous boundary into the discretization.

Approaches for the imposition of boundary conditions in meshless methods can be classified into essential and natural methods. Essential methods are those which integrate the boundary conditions *a priori* into the space of solution candidates, often via the definition of Sobolev H_0 spaces. Numerically, this often translates into the inconvenient necessity to place discretization nodes directly on the boundary for direct collocation (see [170, 237, 239] for instance), or to couple the meshless simulation with a mesh-based finite element procedure (as described in [119, 191] for example). The other alternative is to use natural boundary enforcement methods, namely procedures which integrate the boundary conditions into the continuous operator rather than into the function space. Among them, let us refer to the method of Lagrange multipliers [28, 114] which requires the solution of a saddle point-type linear system, the penalty method [213, 239], which often requires high values of the penalty parameter (with detrimental consequences on the condition number of the discretized system), and the Nitsche symmetric and non-symmetric methods (see [13, 110, 236]), which can be seen as consistent improvements of the penalty method. The most influential paper recapitulating the state of the art at that time is arguably that of Fernández-Méndez and Huerta [97], which overall praises the practical benefits of using Nitsche-type formulations.

Secondly during this period, several facets of a common fundamental problem were identified. Some authors saw its manifestation in the lack of accuracy or stability, and proposed

several remedies to improve these aspects like renormalization (see for instance the contributions of Randles and Libersky [192] and Lanson and Vila [135]), the use of MLS shape functions in SPH by Dilts [77, 78], or the use of additional integration nodes (the so-called "stress points" by Belytschko et al. [25] for instance). Other interpreted it as the manifestation of a fundamental duality between consistency and conservation, the two aspects needing to be addressed. This for instance gave rise to the C-SPH formulation of Bonet and Kulasegaram [36, 37, 39]. Finally, other authors incriminated the properties of the numerical integration procedure, and proposed improvements or corrections like the Stabilized Conforming Nodal Integration (SCNI) by Chen et al. [53], or corrections to ensure the so-called " zero row sum condition" by Babuška et al. [12, 14]. Most authors eventually agreed that the source of their difficulties stemmed from the fact that conditions for first order consistency of their final discretization were not easily met. It was realized that in general, both exactness on linear functions and a discrete analogue of Stokes's formula are necessary to ensure patch test condition and optimal convergence. In a sense, they require that the discrete differentiation and integration procedures be mindful of each other, and for this reason, we call them "compatibility conditions". They are notoriously tough to meet in a fully meshless context and constitute the main concern of the present work.

For the past decade, meshless methods have entered a third "renaissance" phase. The difficulties stated above have been widely recognized and several propositions were made to tackle them. Some of them rely on a mesh built on the point cloud like the power diagram method by de Goes et al. [67]. Others like Chiu [60] solve a global linear system to generate adequate discrete operators. Some authors also completely circumvent the problem by resorting to either strong or non-symmetric Petrov-Galerkin formulations like Chen [54] and his assumed strain variationally consistent integration, or the Meshless Local Petrov-Galerkin (MLPG) approach by Atluri and Zhu [11].

In a nutshell, any currently known means of ensuring exact compatibility within a weak formulation can be classified into one of two categories: either a mesh or some weakened version of a mesh is used, or a global system of (possibly linear) equations needs to be solved. While the first alternative betrays the initial intent of the meshless methods, the second can be extremely costly. In the author's opinion, the next milestone in the history of meshless methods will be reached when this "meshless curse" is either finally broken or explained.

Our contribution to the state of the art

This work develops several original contributions to the theory of meshless methods. For the most part, we restrict ourselves to nodal meshless methods, namely methods that only require the storage of discrete data and evaluation of operators at node locations. This choice is both general enough to encompass many of the most classical meshless methods like SPH and its variants, and restrictive enough to allow the development of a common framework. This thesis proposes an original foundation for nodal meshless methods, with a focus on discrete operators and a precise choice of reduction map, allowing unambiguous notations for discrete operators and definitions of their consistency and compatibility properties. In agreement with recent literature, we give novel numerical evidence that the compatibility error of classical meshless operators like SPH and Diffuse Moving Least Squares (DMLS) gradients quickly increases with point cloud refinement, and numerically demonstrate the accuracy benefits in satisfying these conditions in weak discretizations of elliptic problems.

We propose several novel compatible meshless sets of operators, especially in the vein of Smoothed Finite Element Method (SFEM)-type discretizations. Our search for compatible sets of meshless operators leads us to realize that the rationale behind Lagrangian-based meshless discretizations of the hydrodynamics equations extends far beyond SPH and

Voronoi-based discretizations. To that end, we widely generalize the underlying theory of "volume-based" meshless nodal discretizations, providing characterizations of local and global consistency of meshless operators arising from the dependency of nodal volume weights with respect to geometric and discretization parameters. It is important to note that this theory does not shy away from the proper treatment of the boundary integration operator, which has otherwise always been ignored by the meshless community. In addition, we prove a novel order limitation result pertaining to volume-based gradient operators, mitigating the hope of designing volume-based discretization schemes with consistency properties of arbitrarily high order.

Furthermore, we remark that no currently known numerical method is able to fulfill both first order consistency and compatibility requirements exactly without resorting during their practical construction to some kind of mesh structure or solving a global linear system (*i.e.* a linear system whose number of degrees of freedom scales with the number of nodes in the point cloud). We suggest that such methods might simply not exist, and dub this conjecture the "meshless curse". Studying the repercussions of the meshless curse leads us to design several novel meshless methods achieving compatibility. In particular, we develop in this manuscript a general least-square correction method aiming at restoring compatibility of a first order consistent set of discrete meshless operators, and prove that this method achieves some kind of global optimality when applied to classical DMLS gradients.

We then design several mesh-based integration procedures that manage to retain the exact compatibility properties of several SFEM-like discrete meshless operators. In this process, the particular case of volume-based operators requires a precise understanding of the evolution of the geometry of cells in a mesh during the advection of its nodes. We thus develop an algorithm to compute volumes and centroids of polygonal cells. This algorithm is specifically tailored to the topological description of a mesh in terms of a known combinatorial structure which we call Brisson's cell-tuple structure. Even though this structure was developed by Brisson in the late 80s [45, 46], our approach is extremely original since we first study it independently of a geometrical embedding instead of considering that it ensues from a given subdivided manifold. This allows us to show that the positions of nodes of the mesh fully specify its geometry, even if nodes of the same cell are not constrained to lie in the same hyperplane (such cells are called non-planar), which is particularly useful since this property cannot be assumed for general advection. This original result allows us to effectively compute the needed derivatives of cell volumes and centroids with respect to general variation of node positions.

Finally, we develop and compare several variants of weak nodal meshless discretizations of elliptic equations. All these variants share the same "bulk" consistency conditions, namely first order consistency and compatibility of the discrete operators are sufficient to ensure satisfaction of linear patch test conditions. The different formulations mainly differ by the choice of a method to handle boundary conditions either in a natural or an essential way, and their strategy to circumvent the need to place discretization nodes directly on the boundary of the computational domain.

Even though all numerical applications present in this thesis are two-dimensional, and we can safely assume that most industrial applications of our work are limited to three-dimensional cases, every theoretical aspect is treated for an arbitrary finite dimension d. This follows our firm belief that any dimension-specific "trick" only yields short-lived benefits, especially compared to the impossibility to generalize.

This also translates to a rigorous programming ethics sustained throughout our work. Our numerical results were all obtained after implementation into an in-house prototyping library called Fortran Template Library (FTL) [181] in a non-standard extension to Fortran 90, allowing template meta-programming. This capability is fully exploited since the dimen-

sion of the problem at hand is present in the code as an integer template argument, only specified at compile time. This includes both low-level subroutines typically designed for complex algebraic purposes like second order symmetric tensors or fourth order tensors with minor and major symmetries (pertaining to linear elastic constitutive equations), multidimensional modular arithmetic, exterior algebra operations: wedge product, interior product, Hodge operator, ... This is also combined with more complex constructions like pointwise evaluation of basis polynomials of arbitrary order for the computation of Moving Least Squares (MLS) operators for instance, and also concerns more high level structures and algorithms like point cloud and mesh handling, including volume and centroid computation and the evaluation and storage of meshless operators.

Outline of the thesis

The present document is structured as follows. The first chapter is dedicated to the theoretical study of meshes. More than just providing a clearer definition of meshless methods by a proper definition of what a mesh really is, we detail the topological properties allowing the natural definition of a mesh-based compatible structure. Then, we design an algorithm to compute the volume and centroids of planar polygonal cells in a mesh, which we then generalize to the case of non-planar cells. Incidentally, we not only show that the knowledge of incidence relationship and nodal positions is sufficient to fully characterize the geometry of a mesh, but we also enable the computation of the derivatives of cell volumes and centroids with respect to these nodal positions. This is essential for the mesh-based integration of meshless volume-based SFEM operators. Finally, several first order consistent mesh-based integration formulae are exposed and proved.

The second chapter introduces notions relative to meshless operators: point cloud, reduction map, boundary and volume integration operator, gradient operator, reconstruction operator, their consistency properties and compatibility. The compatibility conditions are interpreted, their importance stressed and the difficulty to achieve them is summarized in the "meshless curse". Then, several examples of meshless operators are detailed: the SPH operators, which are – in their original formulation – neither consistent nor compatible, and their renormalized versions which are up to first order consistent. Then, a general quadratic optimization method for the recovery of compatibility in a first order consistent set of meshless operators is presented. Moreover, we prove that the application of this method to first order DMLS gradients satisfies global quadratic optimality requirements, providing an efficient means of computing such operators. Finally, we generalize the SFEM concepts to the construction of consistent compatible discrete meshless operators and show that these properties can be exactly retained using a mesh-based discrete integration procedure.

The third chapter builds on the idea that the specification of an appropriate volume integration operator (through its associated nodal weights) as a function of geometric parameters is sufficient to define all other discrete operators. We develop an original general theory of such operators, widely extending previous constructions and characterizations. Furthermore, we prove an impossibility result concerning first order consistency of volume-based primal gradient operators. This theory also allows a novel interpretation of the effectiveness of several classical stabilization procedures: they actually improve the consistency of the dual gradient operator. Finally, we are able to exhibit compatible sets of meshless operators built within this volume method, as well as adapt the mesh-based procedure of the previous chapter to their compatibility-preserving approximation, giving yet another illustration of the meshless curse.

The fourth and last chapter uses the meshless framework and operators developed earlier to solve systems of elliptic differential equations. The importance of satisfying the

compatibility conditions – at least in an approximate fashion – is numerically illustrated on several examples: in a nutshell, we are only able to achieve second order convergence of elliptic discretizations using first order consistent compatible operators. Two choices of formulations for the imposition of boundary conditions are compared: on the one hand, the Nitsche formulation imposes Dirichlet boundary conditions weakly, which fits perfectly to our operator-based meshless framework. On the other hand, we show how this framework can be extended to treat Dirichlet boundary conditions essentially, without the need to place boundary nodes directly on the boundary. The two approaches are then compared on a common stress intensity factor estimation test case.

Finally, several examples and miscellaneous results completing the exposition are moved to the appendices at the end of the document.

Chapter 1

Concerning Meshes

Contents

This work focuses on the study of meshless methods, namely numerical methods for the approximation of the solution to PDEs that do not (or as little as possible) require the use of a mesh. Since the main challenge faced by meshless discretization methods is to recover what is lost when we decide to avoid using a mesh in the discretization process, we consider that a clear understanding of the different aspects of meshes is a prerequisite to the construction of innovative meshless methods. The aim of this chapter is to introduce the relevant mesh-based notions for the study and development of meshless methods.

Depending on the context (numerical simulation, computer graphics, ...) and the specific use case, different authors give different definitions of a mesh or stress different aspects. Consequently, there is not even a unique definition of a polygon (or polyhedron, *etc*) as discussed by Grünbaum [112]. Our choice of combinatorial structure to represent meshes is Brisson's cell-tuple structure, which is both general enough to represent a large class of geometrical shapes, and convenient enough to provide an adequate theoretical framework for their study as well as an easy enough interpretation in terms of simplices. In section 1.1, we develop mesh-related topological notions, namely all aspects of the mesh that are independent of a specific embedding: incidence relation between cells, orientations and boundary operator. These notions are fundamental for the theoretical study of PDEs and are unfortunately the first to be left behind when going from mesh-based to meshless methods.

We realize in subsequent chapters that retaining a weakened version of the duality between the boundary operator and the exterior differentiation through integration is of the utmost importance.

Contrary to the original exposition by Brisson [45, 46], we choose to define the combinatorial structure *a priori* instead of deriving its properties from those of general subdivided manifolds. This original change of perspective allows us to strip the geometrical requirements down to their bare minimum. In particular, we show in section 1.2 that the knowledge of a cell-tuple structure and nodal positions is enough to unambiguously define the mesh geometry, with minimal restrictions. This is the most important result of this chapter: it allows us to transport nodes of a mesh with the guarantee that cells naturally follow. This feature is highly desirable as it ensures that the mesh-based approximations of integrals developed in section 1.3 can be used to define adequate computation procedures for the meshless operators as explored later in this thesis.

1.1 Topological aspects of meshes

In this section, we address the question of choosing an appropriate theoretical framework for the discrete representation of space, independently of their embedding. Namely, we describe incidence relations between the fundamental constituents of a mesh, regardless of their precise shapes or geometric position of their points of interest (nodes, integration points, etc ...). This non-metric structure is at the center of the development of several mesh-based simulation techniques. Some methods necessitate the use of simple configurations: for instance, most Finite Differences (FD) schemes are defined in terms of logically Cartesian grids (nodes have right/left neighbors in 1-D, top/bottom neighbors in 2-D, front/back neighbors in 3-D, etc ...). Similarly, many FEM schemes rely on a simplicial decomposition of space for the definition of shape functions. Several simulation methods also exploit the topological structure of the discrete space: for instance, the Finite Volume Method (FVM) (see [226] for an introduction to FVMs) heavily relies on the notion of faces separating cells to define fluxes of conserved quantities. Generalizing these notions to multiple dimensions have allowed the definition of Compatible Discrete Operator (CDO) (also called *mimetic*, *compatible* or *structure-preserving*, see [52]) discretization schemes such as the Virtual Element Method (VEM) (see [22, 23] for quick introductions to the VEM), Discrete Exterior Calculus (DEC) (see [72, 117] and [71] for a practical introduction), the Finite Element Exterior Calculus (FEEC) by Arnold et al. (see [10] for an in-depth introduction). The underlying reasons for the success of these schemes are briefly explained at the end of section 1.1.2. The review work of Bonelle [35] constitute a well-detailed presentation of CDO schemes in a common framework.

The following sections introduce a few notions of algebraic topology that are required to fully understand the role of meshes in the discretization of PDEs. Section 1.1.1 describes incidence relationships in a mesh with a particular focus on cell-tuple structures and section 1.1.2 recalls a few known results concerning the orientation of such structures.

1.1.1 Organizing incidence relations in a topological structure

In this section, we characterize the hierarchical organization of cells within a mesh and detail two particular types of such organizations: simplicial complexes and cell-tuple structures. The incidence relations in a mesh are naturally described by the notion of abstract cell complex (see [9] paragraph 1.7 and [222] paragraph 2.1):

Definition 1.1 (Abstract cell complex)**.**
An *abstract cell complex* $\mathcal{A}_{CC} = (C, <, \dim)$ of dimension d is made of a set C of abstract elements called cells, an irreflexive, asymmetric and transitive binary relation $<$ and a mapping $\dim : C \to [\![0, d]\!]$ compatible with $<$, *i.e.* such that $\forall\ \not{c}, \not{c}' \in C,\ (\not{c} < \not{c}') \Rightarrow \dim(\not{c}) < \dim(\not{c}'))$.

In what follows, an exponent on the cell symbol always refers to the dimension of the cell ($\not{c}^p$ is a cell of dimension p or *p-cell*), and the set of p-cells in a cell complex is usually denoted C^p. Cells of maximal or *full* dimension d are called volume cells or simply cells and cells of dimension $d-1$ are called faces. At the other end of the dimension spectrum, 1-cells are called edges and 0-cells are called nodes. Informally, the binary relation $<$ represents the fact that a cell is in the boundary of another cell. For this reason, in following developments we prefer to denote $\not{c}_1 \in \partial\not{c}_2$ instead of $\not{c}_1 < \not{c}_2$.

The definition of abstract cell complexes is very broad and needs to be restricted to fit our idea of a mesh. Many combinatorial definitions have been proposed to this effect, either for a specific dimension of abstract cell complex [19, 20, 66] or more recently for arbitrary dimensions [33, 46, 50, 65, 142]. Numerous similarities and equivalency results relating these discrete models have been proved [7, 8, 108, 222], so that there is no real wrong choice. The construction of a boundary operator on such structures was recently investigated by Alayrangues et al. [6], which greatly inspired our developments. In this work, we only consider two (related) structures: simplicial complexes and the cell-tuple structure of Brisson [46].

First, let us introduce abstract simplicial complexes, which generalize the 2-D and 3-D definitions of conformal triangular and tetrahedral meshes independently of their embedding:

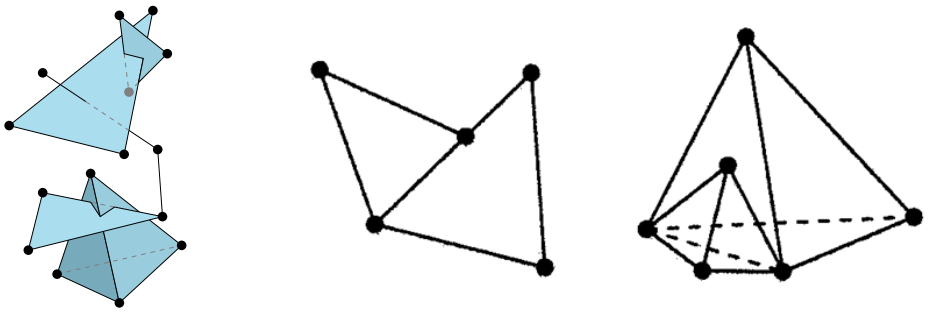
Definition 1.2 (Abstract simplicial complex)**.**
An *abstract simplicial complex* of dimension d is an abstract cell complex $\mathcal{A}_{CC} = (C, <, \dim)$, such that:

- C is a set of finite sets of indices of cardinal at most d such that if $\not{c} \in C$ then $\forall\ \not{c}' \neq \varnothing \subset \not{c}, \not{c}' \in C$.
- The boundary relation $<$ is the "natural order" (see [9] page 121), *i.e.* $\not{c}' < \not{c}$ if and only if $\not{c}' \subsetneq \not{c}$.
- $\dim(\not{c})$ is given by the cardinal of $\not{c}$ minus one.

Simplicial complexes are very handy to manipulate, especially because of their natural relationship with affine embeddings as we explore in section 1.2.3. Their definition readily implies that p-cells of a simplicial complex can be unequivocally described by the set of their $p+1$ incident nodes. Example of valid and non-valid cell complexes are given in figure 1.1.

Dealing with simplicial meshes can be too restrictive: for instance, their dual (see definition 1.8) is not a simplicial complex, and they can be ill suited for adaptive local refinement. The above definition is also too permissive since it allows the creation of objects of non-homogeneous dimensions (see figure 1.1 a)) for instance. Even though this particular issue can be fixed with an additional assumption (see definition 1.3) these issues have led to the development of unstructured meshes. In [43], Brezzi et al. also mention the development of numerical methods for non-matching meshes, and the necessity to simulate regions with complex solution transitions as a motivation for the more and more widespread usage of unstructured meshes with non-planar, degenerate and non-convex elements.

For these reasons, we also consider another restriction of abstract cell complexes called cell-tuple structures (definition 1.7, introduced in [46]). The choice of this particular combinatorial structure is explained in section 1.2 where we study embeddings of cell complexes.



a) b) c)

Figure 1.1 – *a) An example of simplicial complex that does not verify the maximal chain property. It is not properly embedded in 3-dimensional space since it self-intersects. Image courtesy of [225].*
b) c) 2 and 3-dimensional examples of non-compatible simplicial meshes: even if the cells have simplicial shapes, every subset of nodes of a cell does not define a cell. However, these can be considered as valid cell-tuple structures. Image courtesy of [157].

The following definition constraints the dimensions of sequences of incident cells in an abstract cell structure.

Definition 1.3 (Maximal chain property)**.**
An abstract cell complex $\mathcal{A}_{\mathcal{CC}} = (C, \in \partial, \dim)$ satisfies the *maximal chain property* if for every ordered sequence of cells $\not{c}_0 \in \partial\not{c}_1 \cdots \in \partial\not{c}_k$, there exists a maximal sequence of $d+1$ cells $\not{c}^0 \in \partial\not{c}^1 \cdots \in \partial\not{c}^d$ such that:

- $\forall\ i \in [\![0, k]\!],\ \exists\ p \in [\![0, d]\!],\ \ | \not{c}_i = \not{c}^p$
- $\forall\ p \in [\![0, d]\!],\ \dim(\not{c}^p) = p$

Maximal ordered sequences of cells (*i.e.* paths of incident cells of consecutive dimension linking a 0-cell to a d-cell) are called cell-tuples or more simply tuples, and the set of tuples in an abstract cell complex $\mathcal{A}_{\mathcal{CC}}$ is denoted $\mathcal{T}(\mathcal{A}_{\mathcal{CC}})$. Similarly, we call "partial tuple" of length p or p-tuple an ordered sequence of $p+1$ cells whose dimension is at most p, and denote $\mathcal{T}^p(\mathcal{A}_{\mathcal{CC}})$ the set of p-tuples. An exponent on the tuple symbol denotes the size of the tuple: t^p is a p-tuple. The cell of dimension $p \in [\![0, d]\!]$ in a tuple t is denoted $\not{c}^p(t)$.

The best way to picture tuples and partial tuples is probably to realize that abstract cell complexes satisfying the maximal chain property can be represented as d nested bipartite graphs as pictured on figure 1.2. Indeed, definition 1.3 ensures that there is no dimensional jump between incident cells: two incident cells $\not{c}^q \in \partial\not{c}^p$ of and abstract cell complex satisfying the maximal chain property are always linked through a path of incident cells of consecutive increasing dimensions. Another way to understand tuples is to consider the *barycentric subdivision* defined here:

Definition 1.4 (Barycentric subdivisions of an abstract cell complex)**.**
Let $\mathcal{A}_{\mathcal{CC}}$ be an abstract cell complex $\mathcal{A}_{\mathcal{CC}} = (C, \in \partial, \dim)$ satisfying the maximal chain property. Then $\mathcal{T}(\mathcal{A}_{\mathcal{CC}})$ can be considered as a set of d-dimensional simplices with nodes in C. In other words, tuples are the d-cells of a simplicial complex whose nodes are the cells of $\mathcal{A}_{\mathcal{CC}}$. This simplicial complex is called the *barycentric subdivision* of a cell complex (see Lemma 1 of [46] and [9] pages 131-132).

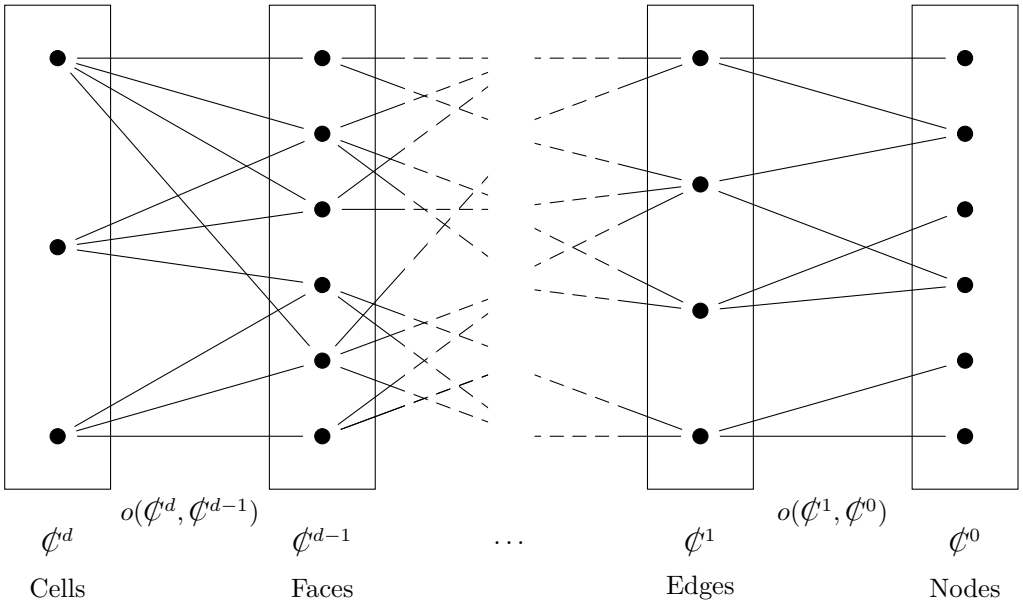


Figure 1.2 – *Incidence relations in an abstract cell complex satisfying the maximal chain property can be encoded as d nested bipartite graphs. In this graph, a tuple can be thought of as a path joining a 0-cell to a d-cell and a p-tuple as a path joining a 0-cell to a p-cell . Relative orientations map the edges of the graph to* $\{-1, +1\}$.

The barycentric subdivision turns out to be an efficient tool to study the properties of an abstract cell complex. It is at the center of several discretization methods and we use it several times in our subsequent developments. The next definitions detail two combinatorial restrictions introduced by Brisson in the late 1980s to restrict the category of representable objects with abstract cell complexes. Their relevance is best understood in the light of proposition 1.6 (see section 1.2.1).

Definition 1.5 (Relaxed Brisson's switch condition)**.**
An abstract cell complex $\mathcal{A}_{\mathcal{CC}}$ satisfies the *relaxed Brisson's switch condition* if for all dimension $p \in [\![0, d-1]\!]$ and tuple t, there exists a unique tuple t' that agrees with t on all cells except that of dimension p:

$$\forall\ t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}), \exists\ !t' \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}) \mid \forall\ q \in [\![0, d]\!], q \neq p, \mathcal{C}^q(t') = \mathcal{C}^q(t) \text{ and } \mathcal{C}^p(t') \neq \mathcal{C}^p(t) \quad (1.1)$$

Unicity allows the definition of the "switch" functions acting on tuples: the unique tuple t' satisfying the above condition is denoted $\text{switch}_p(t)$. The symmetric roles of t and t' in this definition readily entails that switch_p is an involution: $\forall\ p \in [\![0, d-1]\!], \forall\ t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}), \text{switch}_p(\text{switch}_p(t)) = t$
Additionally, for the "full" dimension $p = d$, at most one such tuple may exist.

The switch condition is the natural generalization to arbitrary dimensions of the statements "every edge separates exactly two nodes" and "every face separates exactly two cells". It is readily seen that the latter should not hold for boundary faces, which are incident to an interior cell and the exterior. With this definition, boundary faces are faces that belong to tuples t such that $\text{switch}_d(t)$ is not defined. Boundary cells of dimension $p < d-1$ are defined as incident cells to boundary faces.

Definition 1.6 (Local path-connectedness)**.**
An abstract cell complex $\mathcal{A}_{\mathcal{CC}}$ satisfying the relaxed Brisson's switch condition is said "locally path-connected" if two tuples t, t' sharing all cells of dimensions $i \in \mathbb{I} \subset [\![0, d]\!]$ can be transformed into each other with repeated application of switch$_p$ for $p \notin \mathbb{I}$:
$\forall\, \mathbb{I} \subset [\![0, d]\!], \forall\, t, t' \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}) \mid \forall\, i \in \mathbb{I}, \phi^i(t) = \phi^i(t')$,

$$\exists\, (p_1, \ldots, p_m) \in ([\![0, d]\!] \backslash \mathbb{I})^m \mid t' = \text{switch}_{p_m} \circ \cdots \circ \text{switch}_{p_1}(t) \tag{1.2}$$

The local path-connectedness condition forbids the construction of pathological cells. Figure 1.3 b) gives an example of an embedded cell structure that is not a subdivided manifold since its unique cell is not homeomorphic to a disc. The corresponding abstract cell structure does not satisfy local path-connectedness.

Actually, assuming local path-connectedness (or a close enough substitute) allows a major change of perspective: instead of founding the combinatorial structure on cells, some authors (see [65] for instance) prefer to first introduce objects called *darts* (tantamount to our definition of tuples) and several functions acting on darts (similar to Brisson's switch). Cells are only then defined as the repeated action (orbits) of these functions on darts. Local path-connectedness ensures the equivalence of the two approaches. We use it in the next section to show that properties defined on tuples readily translate to properties on cells.

We now have all the relevant tools at our disposal to define a proper combinatorial structure for the representation of meshes.

Definition 1.7 (Cell-tuple structures)**.**
Abstract cell complexes that satisfy the maximal chain property (definition 1.3), the relaxed Brisson's switch conditions (definition 1.5) and local path connectedness (definition 1.6) are called *cell-tuple structures*. Additionally, a cell-tuple structure verifying the Brisson's switch condition even for the full dimension $p = d$ is called a cell-tuple structure without boundary.

A 2-D example of a simplicial complex, which is a cell-tuple structure as well is given in figure 1.4.

Proposition 1.1 (Barycentric subdivisions of cell-tuples are cell-tuples)**.**
The barycentric subdivision of a cell-tuple structure is itself a cell-tuple structure.

Proof. Let $\mathcal{A}_{\mathcal{CC}} = (C, <, \dim)$ denote a cell-tuple structure. We want to show that its barycentric subdivision is endowed with a cell-tuple structure, namely that it satisfies the maximal chain property, the relaxed Brisson's switch condition and local path-connectedness. The set of tuples of the barycentric subdivision of $\mathcal{A}_{\mathcal{CC}}$ is denoted $\mathcal{T}^2(\mathcal{A}_{\mathcal{CC}})$ and is comprised of ordered sequences of the form $t^\sigma = (\phi^{\sigma(0)}, \phi^{\sigma(1)}, \ldots \phi^{\sigma(d)})$ where $\sigma \in Sym([\![0, d]\!])$ and $t = (\phi^0, \phi^1, \ldots \phi^d) \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}})$. By construction, the maximal chain property is satisfied. The switch operator switch$^{\mathcal{T}^2(\mathcal{A}_{\mathcal{CC}})}$ acting on $\mathcal{T}^2(\mathcal{A}_{\mathcal{CC}})$ is naturally defined as:

$$\begin{cases} \text{switch}_d^{\mathcal{T}^2(\mathcal{A}_{\mathcal{CC}})}(t^\sigma) & = \text{switch}_{\sigma(d)}(t)^\sigma & \text{Where switch is defined} \\ \text{switch}_p^{\mathcal{T}^2(\mathcal{A}_{\mathcal{CC}})}(t^\sigma) & = t^{\pi_p \circ \sigma} & \text{For } p < d \end{cases}$$

Where π_p is the adjacent transposition exchanging p and $p+1$. Since π_p is an involution, the relaxed Brisson's switch condition on the barycentric subdivision is a consequence of its being satisfied on the original cell-tuple structure. Similarly, local path-connectedness on the barycentric subdivision is a consequence of the fact that adjacent transpositions generate the symmetric group and of local path-connectedness on the original cell-tuple structure. □

A 2-D example of barycentric subdivision is sketched on figure 1.4.

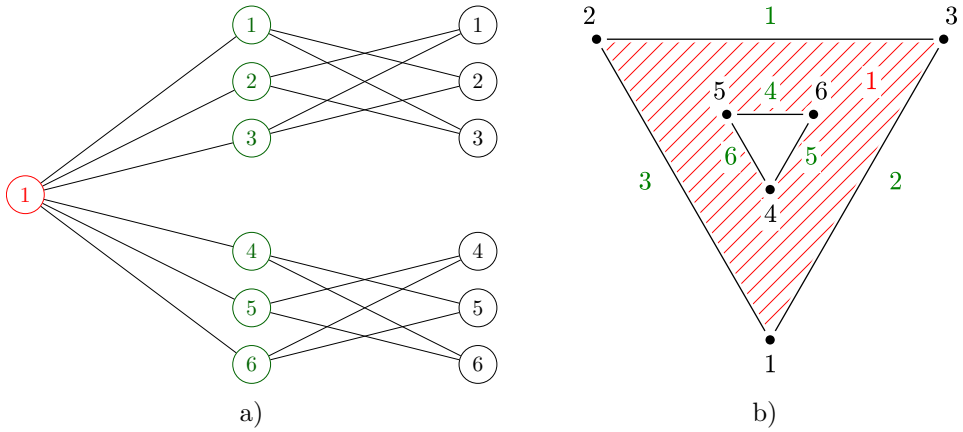


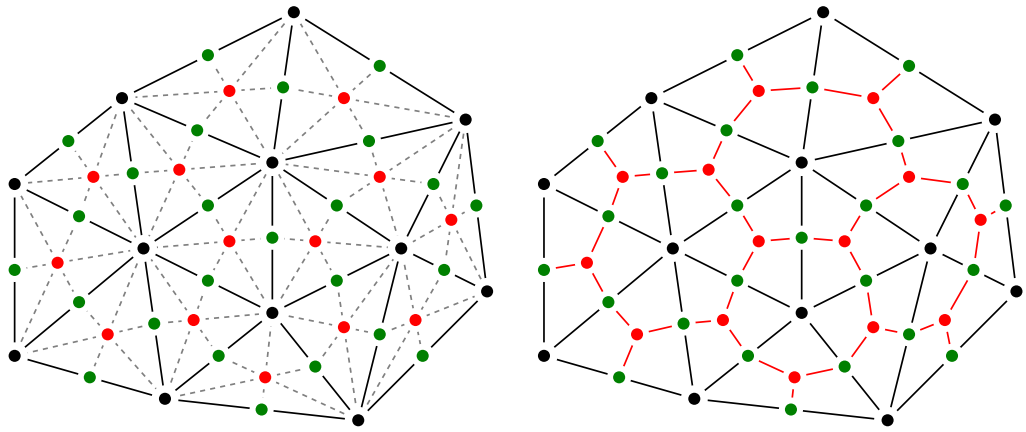
Figure 1.3 – *Local path-connectedness forbids the construction of cells with pathological shapes.*
a) A 2-D abstract cell structure with one cell which does not satisfy local path-connectedness (definition 1.6). For instance the tuple $(1, 2, 1)$ *cannot be transformed into* $(4, 5, 1)$ *by means of* switch_0 *and* switch_1 *only.*
b) A possible corresponding embedding in $\mathbb{R}^2$*, which is not homeomorphic to a disc.*

Definition 1.8 (Dual cell-tuple structure)**.**
Let $\mathcal{A}_{\mathcal{CC}} = (C, <, \dim)$ be a cell-tuple structure without boundary. Inverting the incidence relation defines a new "dual" cell-tuple structure ${\mathcal{A}_{\mathcal{CC}}}^* = (C, >, d - \dim)$. This procedure effectively maps p-cells of the initial "primal" structure to $d - p$-cells of the dual one. This procedure is involutive: the dual of the dual is the initial structure.

The definition of the dual structure is especially convenient for the definition of staggered schemes, which express physical quantities for the next iteration on the dual mesh (see [179] for instance). They also provide an ideal tool for the discretization of the Hodge operator, which maps p-differential forms to $d - p$ forms (see [116, 173] for instance).

Remark. The dual cell-tuple structure can also be defined for cell-tuple structures with boundary, but the definition is slightly more involved, and the process is no longer an involution.

In the next section, we describe the meaning of relative orientation between cells, define a boundary operator and sketch a few relevant results in homology and cohomology theory to the discretization of PDEs.



a) b)

Figure 1.4 – *Simplicial complexes and cell-tuple structure are highly related.*
a) In black: an example of simplicial complex which is also a cell-tuple structure. Its barycentric subdivision defined in proposition 1.1 is explicitly built (in light grey). Black nodes correspond to nodes, green nodes to edges and red nodes to cells of the original cell-tuple structure.
b) Inverting the orientation of the incidence relation maps p-cells of the cell-tuple structure to $(d-p)$-cells of the "dual" structure, which is not a simplicial complex.

1.1.2 Orientation and boundary operator of a cell-tuple structure

In order to make sense to physical signed values like the *outward* of *inward* flux of a quantity into a cell or its flow *from* one cell *to* another, we need to define a notion of orientation on the cell-tuple structure. However, if we still want to be able to represent non-orientable objects (like the Möbius band or higher dimensional equivalents), orientation may not always be defined globally.

Definition 1.9 (Orientability of a cell-tuple structure)**.**
A cell-tuple structure $\mathcal{A}_{\mathcal{CC}}$ is said *locally orientable* if there exists a map $o : \mathcal{T}(\mathcal{A}_{\mathcal{CC}}) \to \{-1, +1\}$ such that:

$$\forall\ t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}), \forall\ p \in [\![0, d-1]\!], \quad o(\text{switch}_p(t)) = -o(t) \tag{1.3}$$

Furthermore, it is said *globally orientable* or simply *orientable* if the above condition is met for the full dimension $p = d$ as well (where the switch operator is defined).

Definition 1.10 (Orientation of a cell-tuple structure)**.**
A *local orientation* of a cell-tuple structure $\mathcal{A}_{\mathcal{CC}}$ is a set of $d+1$ maps $o_i : \mathcal{T}(\mathcal{A}_{\mathcal{CC}}) \to \{-1, +1\}$ for $i \in [\![0, d]\!]$ such that:

$$\forall\ t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}), \forall\ p \in [\![0, d]\!], \begin{cases} \text{if } p < i \text{ then } o_i(\text{switch}_p(t)) = -o_i(t) \\ \text{if } p > i \text{ then } o_i(\text{switch}_p(t)) = \quad o_i(t) \end{cases} \tag{1.4}$$

Furthermore, the orientation is said *global* if the above condition is met for the full dimension $i = p = d$ as well where the switch operator is defined. Global orientations are local orientations, but the converse is not true in general. A concrete example of oriented cell-tuple structure is sketched on figure 1.5.

Remark. The orientation map o_i can equivalently be defined on i-tuples as the second condition suggests. Indeed, for $p \geq i$, we can consistently denote $o_i(t^p) = o_i(t)$ for all tuples t agreeing with t^p on the $p+1$ first cells.

The choice of a (local) orientation is non-unique and eventually a matter of convention. Indeed, if $(o_i)_{i\in[\![0,d]\!]}$ is a valid (local) orientation, then the opposite orientation $(-o_i)_{i\in[\![0,d]\!]}$ also is. In general, orientation is chosen cell-wise as illustrated in the proofs of proposition 1.3 and 1.4.

Proposition 1.2 (Orientability is a *sine qua non* condition to admit an orientation)**.**
A cell-tuple structure admits a (local) orientation if and only if it is (locally) orientable.

Proof. The proofs for local and global orientations are almost identical. Let us for instance consider local orientations. First, suppose a cell-tuple structure $\mathcal{A}_{\mathcal{CC}}$ admits a local orientation $(o_i)_{i\in[\![0,d]\!]}$. Then $o = o_d$ satisfies the local orientability conditions.
Conversely, suppose a cell-tuple structure $\mathcal{A}_{\mathcal{CC}}$ is locally orientable, *i.e.* that there exists an orientation map $o : \mathcal{T}(\mathcal{A}_{\mathcal{CC}}) \to \{-1, +1\}$ such that:

$$\forall\ t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}), \forall\ p \in [\![0, d-1]\!], \quad o(\text{switch}_p(t)) = -o(t)$$

Then defining $o_d = o$ satisfies the criteria of definition 1.10. Using descending induction, suppose we have found suitable o_i maps for all $j < i \leq d$. Then for $i = j$, let us consider the set of pairs of the form $(\not{\phi}^i, \not{\phi}^{i+1})$. From local path-connectedness (definition 1.6), considering such a pair $(\not{\phi}^i(t), \not{\phi}^{i+1}(t))$ amounts to considering all tuples of the form $\text{switch}_{p_m} \circ \cdots \circ \text{switch}_{p_1}(t)$ where $p_k \in [\![0, i-1]\!] \cup [\![i+1, d]\!]$. Following the Brisson switch condition (definition 1.5), the switch_{i+1} operator is a permutation of such pairs, its repeated action thus defines an equivalence relation whose equivalence classes – called cycles – have cardinal 2. We pick a pair $(\not{\phi}^i, \not{\phi}^{i+1})$ from each class, and define $o_i(t) = o_{i+1}(t)$ for all corresponding tuples. We then define the orientation of members of the other class as $o_i(\text{switch}_{i+1}(t)) = o_{i+1}(t)$. This definition verifies: $\forall\ t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}})$, $\forall\ 0 \leq p < i$, $o_i(\text{switch}_p(t)) = -o_i(t)$ inherited from o_{i+1}. By construction, we have $o_i(t) = o_i(\text{switch}_{i+1}(t))$. Moreover, $\forall\ i+2 \leq p < d$, $o_i(\text{switch}_p(t)) = o_i(t)$ inherited from o_{i+1}, which proves the result by descending induction. □

Proposition 1.3 (Orientability of barycentric subdivisions)**.**
The barycentric subdivision of a cell-tuple structure is locally orientable. Furthermore, it is globally orientable if and only if the original cell-tuple structure is globally orientable.

Proof. Keeping the notations of the proof of proposition 1.1, defining $o(t^\sigma) = \text{sign}(\sigma)$ proves local orientability. Moreover, given a global orientation of the original cell-tuple structure, we can define $o(t^\sigma) = o(t) \times \text{sign}(\sigma)$, which in turn proves global orientability of the barycentric subdivision. Now let us assume that there exists a global orientation on the barycentric subdivision. Then, the value of $o(t^\sigma) \times \text{sign}(\sigma)$ is independent of the chosen permutation $\sigma \in Sym([\![0, d]\!])$ and defines a proper global orientation on the original cell-tuple structure. □

In fact, we can even state the following result:

Proposition 1.4 (Cell-tuples structures are locally orientable)**.**
A cell-tuple structure is always locally orientable.

Proof. Let $\mathcal{A}_{\mathcal{CC}}$ be a cell-tuple structure. Let us inductively build a local orientation on partial tuples to prove the result. For $p = 0$, defining $o_0(t^0) = 1\ \forall\ t^0 \in \mathcal{T}^0(\mathcal{A}_{\mathcal{CC}})$ satisfies every condition of definition 1.10. Let us suppose we have built local orientation maps o_i

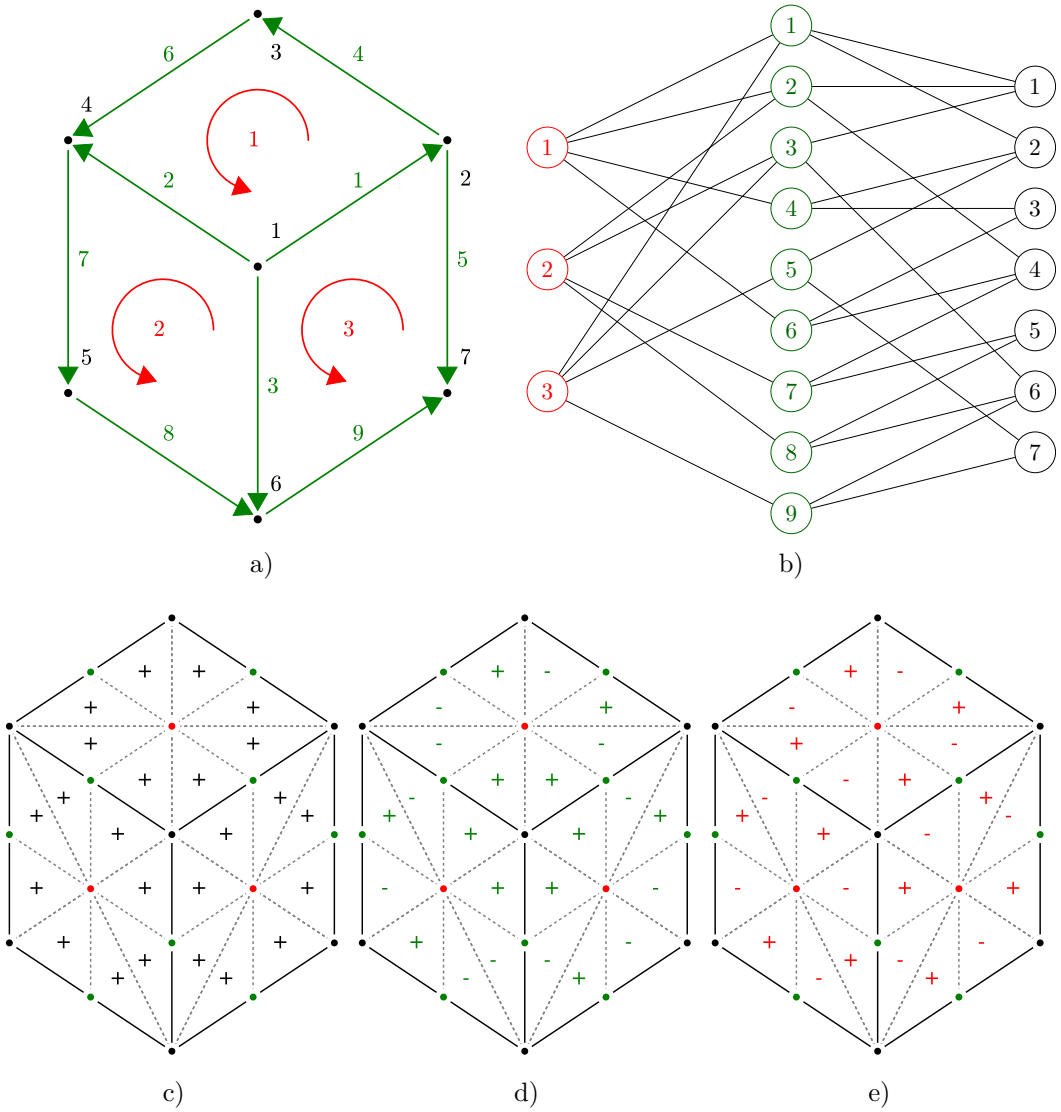


Figure 1.5 – *Relative orientation of cell-tuple structures allow the definition of a boundary operator on formal sums of cells.*
a) The orientation of 2-cells is positive relative to the conventional orientation of the plane. The relative orientation of 1-cells within them is positive if the arrows go in the same direction, negative otherwise. 0-cells are negatively oriented within 1-cells if the arrow leaves the 0-cells.
b) The values of $o(\mathcal{C}^{p-1}, \mathcal{C}^{p})$ *are attached to the edges of the incidence graph.*
c)d)e) Consistency of the relative orientation is a consequence of the fact that the switch *operation reverses orientation of tuples of lower dimensions, and conserves that of tuples of higher dimensions. A possible orientation of tuples consistent with the relative orientation of a) is proposed here.*

for $i < p < d$. In order to define o_p, we need to choose a relative orientation of $(p-1)$-cells into p-cells.

Let $t^p \in \mathcal{T}^p(\mathcal{A}_{\mathcal{CC}})$. From local path-connectedness, considering the pair $(\phi^{p-1}(t^p), \phi^p(t^p))$ amounts to considering all p-tuples of the form $\text{switch}_{p_m} \circ \cdots \circ \text{switch}_{p_1}(t^p)$ where $p_i \in [\![0, p-1]\!]\ \forall\ i \in [\![1, m]\!], m \in \mathbb{N}$. Following the Brisson switch condition (definition 1.5), the switch_p operator is a permutation of such pairs, its repeated action thus defines an equivalence relation whose equivalence classes – called cycles – have cardinal 2, and possibly 1 when $p = d$. Pick a pair (ϕ^{p-1}, ϕ^p) from each class, and define $o_p(t^p) = o_{p-1}(t^p)$ for all corresponding tuples. Then, define $o_p(\text{switch}_p(t^p)) = -o_{p-1}(t^p)$ on the other one (where switch_p is defined). Let us check that this procedure effectively defines a local orientation on $\mathcal{A}_{\mathcal{CC}}$: For $q < p-2$, for all $t^p \in \mathcal{T}^p(\mathcal{A}_{\mathcal{CC}})$, we have: $o_p(\text{switch}_q(t^p)) = \pm o_{p-1}(\text{switch}_q(t^p)) = \mp o_{p-1}(t^p) = -o_p(t^p)$. Furthermore, by construction we have $o_p(\text{switch}_{p-1}(t^p)) = -o_p(t^p)$, thus concluding the proof. □

As we have seen in the proof of proposition 1.4, a local orientation of an abstract cell-tuple structure follows from the choice of a (consistent) relative orientation for pairs of incident cells of consecutive dimensions. The next definition affirms that the converse is also true:

Definition 1.11 (Relative orientation in a locally oriented cell-tuple structure)**.**
For $p \in [\![1, d]\!]$, we call orientation of a $(p-1)$-cell ϕ^{p-1} relative to an incident p-cell ϕ^p and denote $o(\phi^{p-1}, \phi^p)$ the number defined as: $\forall\ t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}}) \mid \phi^{p-1}(t) = \phi^{p-1}, \phi^p(t) = \phi^p$,

$$o(\phi^{p-1}, \phi^p) = o_{p-1}(t) o_p(t)$$

This definition does not depend on a specific tuple t.

Proof. For $i \in [\![1, p-1]\!]$, we have $o_{p-1}(\text{switch}_i(t)) o_p(\text{switch}_i(t)) = -o_{p-1} \times -o_p(t) = o_{p-1}(t) o_p(t)$, and similarly for $i \in [\![p+1, d]\!]$, we have $o_{p-1}(\text{switch}_i(t)) o_p(\text{switch}_i(t)) = -o_{p-1} \times -o_p(t) = o_{p-1}(t) o_p(t)$. Hence, switch_i conserves $o_{p-1}(t) o_p(t)$ for all $i \neq p-1, p$. Using local path-connectedness of the cell-tuple structure shows that all tuples incident to both ϕ^{p-1} and ϕ^p agree on the value of $o_{p-1}(t) o_p(t)$. □

If $o(\phi^p, \phi^{p+1}) = +1$, the two cells are oriented in the same way, and if $o(\phi^p, \phi^{p+1}) = -1$, they are oriented in opposite ways. Relative orientations are attached to pairs of incident cells of consecutive dimensions, namely to edges of the nested bipartite graph of the cell-tuple structure (see figure 1.2). It is customary to choose orientation of tuples such that boundary faces ϕ^{d-1} are positively oriented relative to the unique cell ϕ^d they are attached to: $o(\phi^{d-1}, \phi^d) = 1$. This choice is equivalent to the convention that vector surface areas (see section 1.2.2 for the computation of vector surface areas in the planar case) should point towards the exterior of the domain.

Relative orientations are best understood when considering boundary operators, which we now introduce. For $p \in [\![0, d]\!]$, we consider C^p as the free abelian group generated by the p-cells of a cell-tuple structure. In other words, we study formal sums (also called p-chains) of the form $\sum_{\phi^p \in C} a_{\phi^p} \phi^p$ where $a_{\phi^p} \in \mathbb{Z}$. The boundary operator is defined on these formal sums as:

Definition 1.12 (Boundary operator on a locally oriented cell-tuple structure)**.**
For $p \in [\![1, d]\!]$, the p-th boundary operator ∂_p is defined on C^p with values in C^{p-1} as the group homomorphism with the following action on the canonical basis: $\forall\ \phi^p \in C^p$,

$$\begin{cases} \partial_p : C^p \to C^{p-1} \\ \partial_p \phi^p \overset{\text{def}}{=} \displaystyle\sum_{\phi^{p-1} \in \partial \phi^p} o(\phi^{p-1}, \phi^p) \phi^{p-1} \end{cases} \tag{1.5}$$

Similarly to the boundary operator defined on subsets of $\mathbb{R}^n$, our discrete boundary operator defined on cell-tuple structures is such that the boundary of a boundary vanishes:

Proposition 1.5 (Boundaries have vanishing boundaries)**.**
For $p \in [\![1, d-1]\!]$, we have $\partial_p \circ \partial_{p+1} = 0$.

Proof. Let compute the boundary of the boundary of a cell: $\forall\, p \in [\![1, d-1]\!], \forall\, \phi^{p+1} \in C^{p+1}$,

$$\partial_p \partial_{p+1} \phi^{p+1} = \sum_{\phi^p \in \partial \phi^{p+1}} o(\phi^p, \phi^{p+1}) \sum_{\phi^{p-1} \in \partial \phi^p} o(\phi^{p-1}, \phi^p) \phi^{p-1} \tag{1.6}$$

$$= \sum_{\phi^{p-1} \in \partial \phi^{p+1}} \left(\sum_{\substack{\phi^p \in \partial \phi^{p+1} \\ \phi^{p-1} \in \partial \phi^p}} o(\phi^{p-1}, \phi^p) o(\phi^p, \phi^{p+1}) \right) \phi^{p-1}$$

As a consequence of the relaxed Brisson's switch condition, the sum between parentheses on the second line of equation (1.6) is made of exactly two terms part of switch_p-related pairs of tuples. Thus, there is a tuple $t \in \mathcal{T}(\mathcal{A}_{\mathcal{CC}})$ such that this term reads:
$o_{p-1}(t) o_p(t) o_p(t) o_{p+1}(t) + o_{p-1}(\text{switch}_p(t)) o_p(\text{switch}_p(t)) o_p(\text{switch}_p(t)) o_{p+1}(\text{switch}_p(t))$
$= o_{p-1}(t) o_{p+1}(t) - o_{p-1}(t) o_{p+1}(t) = 0$ □

This proves that C has a chain complex structure, adequate to define a homology theory.

$$C^0 \xleftarrow{\partial_1} C^1 \xleftarrow{\partial_2} \dots \xleftarrow{\partial_{d-1}} C^{d-1} \xleftarrow{\partial_d} C^d \tag{1.7}$$

In a nutshell, cellular homology is the study of the *shape* of cellular complexes independently of their embeddings through the characterization of their homology spaces: $H^p = \text{Ker}(\partial_p) \backslash \text{Im}(\partial_{p+1})$. In particular, the dimension of H^p as a $\mathbb{Z}$-module is called the p-th Betti number and is loosely speaking the number of p-dimensional holes in the complex. It can be shown (see for instance theorem 2.35 of [115]) that cellular homology agrees with singular homology defined on smooth manifolds, namely that cellular decompositions of smooth manifolds appropriately capture the shape of smooth manifolds.

These questions are particularly relevant to the discretization of PDEs. Indeed, numerous PDEs can be re-phrased and better studied in the language of differential forms. The central operator of differential geometry is the exterior derivative, which unifies the vector calculus notions of divergence, curl and gradient within a co-chain structure. The study of the corresponding graded ring is called the De Rham cohomology theory.

$$\Lambda^0 \xrightarrow{\mathrm{d}^0} \Lambda^1 \xrightarrow{\mathrm{d}^1} \dots \xrightarrow{\mathrm{d}^{d-2}} \Lambda^{d-1} \xrightarrow{\mathrm{d}^{d-1}} \Lambda^d \tag{1.8}$$

The exterior derivative acting on differential forms is related to the boundary operator defined on manifolds by Stokes's formula (ι denotes the canonical injection or inclusion map from $\partial\Omega_p$ to Ω_p, see theorem 8.2.8 of [2] for more information):

$$\int_{\Omega^p} \mathrm{d}^{p-1} \alpha^{p-1} = \int_{\partial^p \Omega^p} \iota \alpha^{p-1} \tag{1.9}$$

The De Rham theorem (see [94]) states that integration is actually an isomorphism between De Rham cohomology and singular cohomology (defined on simplicial complexes on smooth manifolds). Hence, the singular homology (shape) of a smooth manifold conditions the space of solutions of PDEs. The success of CDO discretization methods is due to the fact that they define discrete differential forms as co-chains on a cellular cell complex: using the duality pairing between chains and cochains, the boundary operator of definition 1.12 translates

to a discrete exterior derivative operator. Stokes's formula acts here as a definition of the differential operator, and is therefore naturally satisfied at the discrete level. This feature is of the utmost importance to achieve optimal convergence.

In a meshless context, we need to build this discrete exterior derivative from scratch (see section 2.2.2). One of the greatest challenges of meshless methods is precisely to build a discrete differential operator that is both precise enough (more precisely, consistent of order one) and that satisfies Stokes's formula. The importance of this feature is explained in greater length in sections 2.4 and demonstrated numerically in chapter 4.

1.2 The geometry of general meshes

In the last section, we introduced the "cell-tuple structure" and studied its topological properties. The present section is dedicated to the study of the geometrical aspects of meshes. In section 1.2.1, we recall a result by Brisson stating that the topological structure of a wide category of geometrical objects can be described with the cell-tuple structure introduced in the previous section. In sections 1.2.2 and 1.2.3, we develop an algorithm for the computation of the volume and centroid of cells in polygonal meshes and of moments of any order in simplicial meshes. In section 1.2.4, we describe the geometrical sub-cell structure of non-planar cells and extend the validity of previous results to meshes without the planarity assumption.

1.2.1 Subdivided manifolds can be represented by cell-tuple structures

In this section, we first justify our choice of working with cell-tuple structures, with the introduction of the notion of subdivided manifold. Then, we study embeddings of subdivided manifolds and define what we mean by polygonal mesh.

Following Brisson [46] section 3, we give a definition of subdivided manifold, namely a manifold adequately cut into cells.

Definition 1.13 (Subdivided manifold)**.**
A *subdivision* of a topological manifold $\mathbb{M}$ is a set of embedding maps $(\phi_{\mathcal{C}})_{\mathcal{C}\in C}$ where C is a set of disjoint subsets of $\mathbb{M}$ whose union is $\mathbb{M}$ and such that for each $\mathcal{C}$ in C,

- The corresponding embedding map $\phi_{\mathcal{C}} : B^p \to \overline{\mathcal{C}}$ is continuous and injective. It transforms the unit open ball of some dimension p onto $\mathcal{C}$, and the unit closed ball onto $\overline{\mathcal{C}}$.
- The boundary of $\mathcal{C}$ relative to the topology defined by the metric on $\mathbb{M}$: $\partial\mathcal{C} = \overline{\mathcal{C}}\backslash\mathcal{C}$ can be written as the union of (finitely many) cells from C.

The notion of subdivided manifold matches the intuitive idea of decomposition of a domain into cells. The shape of these cells can be complicated (non-convex cells, non-star shaped cells, thin cells, curved cells, ...), but their topology is constrained to be that of a simple ball. The link between cell-tuple structures and subdivided manifolds is given by the following result by Brisson (see [46] for more information).

Proposition 1.6 (Subdivided manifolds define cell-tuple structures)**.**
A subdivision $(\phi_{\mathcal{C}})_{\mathcal{C}\in C}$ of a manifold defines a cell-tuple structure $\mathcal{A}_{\mathcal{CC}} = (C, <, \dim)$ where:

- $\dim(\mathcal{C})$ is the dimension of the domain (and the co-domain) of $\phi_{\mathcal{C}}$.

- $\not{C}^q < \not{C}^p$ if $\not{C}^q \subset \overline{\not{C}^p}$

Proof. See Brisson [46]. □

Hence, the results concerning cell-tuple structures directly apply to subdivided manifolds.

Up until this point, we have considered manifolds independently of a possible embedding. From now on, we study these manifolds $\mathbb{M}$ through a locally flat embedding into $\mathbb{R}^n$ with $d \leq n$. We do not impose that the embedding dimension n fit the intrinsic dimension d of the mesh in order to accommodate lower dimensional objects (meshing of shells for instance).

Definition 1.14 (Polygonal mesh)**.**
A *polygonal mesh* is a subdivided manifold embedded in an affine space $\mathbb{R}^n$ such that each p-cell $\not{C}^p$ spans an affine space of dimension p, which we denote $\mathbb{H}(\not{C}^p)$.

Supposing a proper planar embedding allows the definition of a metric and an orientation on every cell given an orthogonal basis of the spanning affine space. Computing cell volumes and centroids in this case is the object of the next section.

1.2.2 Computation of metric quantities in a polygonal mesh

The goal of this section is to compute the oriented volume $V(\not{C}^p)$ and centroid $\mathbf{x}(\not{C}^p)$ of cells $\not{C}^p$ of a polygonal mesh embedded in $\mathbb{R}^n$. These two notions require the definition of a metric on $\mathbb{R}^n$, which we subordinate to the choice of an oriented frame $(O, \mathbf{e}_1, \ldots, \mathbf{e}_n)$ without choice of generality. This choice translates to a metric on every affine subspace of $\mathbb{R}^n$, and fully determines musical operators $\sharp$ and $\flat$, the Hodge operator $*$ and the different volume forms. Using the affine properties of $\mathbb{R}^n$ and subspaces, we identify points with vectors of the tangent and forms of the cotangent spaces: if $\mathbf{x}$ denotes a point, then there is a corresponding vector field with the same coordinates in a given orthogonal affine frame which we also denote $\mathbf{x}$. Expressions such as $\mathbf{x} - \mathbf{y}$ might either refer to the vector joining point $\mathbf{y}$ to point $\mathbf{x}$, or to the corresponding 1-form.

The computation of volumes and centroids of polygons, and more generally of moments of polygons has been an active research topic for several years. One of the reasons for the wide variety of existing methods is that their computational complexity often depends on how the mesh is represented as analyzed in [81, 127]. A review of some existing methods with associated complexity computed on a specific example is given in [47].

Most industrial simulation codes build explicit decompositions of polygons into simple shapes (often simplices, sometimes orthogonal parallelograms when possible), but these methods can be criticized because of their relative computational inefficiency and the arbitrary nature of the decomposition. One of the best known methods for the computation of volumes of convex polytopes is that of Lasserre [137], extended in [136] for the computation of the integral of homogeneous functions. Our method (see proposition 1.7) is a direct extension of Lasserre's method to non-convex polygons. It also very closely resembles the hybrid method of Büeler et al. [47]. Other famous methods for volume computation are built either around Brion's formula (see [18, 44, 139, 188] for its independent discoveries, and [109] for a more practical approach), which we exploit in section 1.2.3, or more anecdotally around the Lasserre-Avrachenkov formula for the integration of symmetric functions on a simplex (see [128, 130, 138]). These questions are closely linked to the count of integer lattice points inside a given polytope, which is the subject of very recent research (see [16, 68, 69]).

Let us explain what we mean by "oriented volume of a p-cell": if the cell has full embedding dimension $p = n$, its oriented volume is a real number, namely the measure of the open set it bounds. In codimension one however ($p = n - 1$), $V(\not{C}^{n-1})$ denotes the oriented area

of the face $\not{C}^p$, and not only characterizes the $d-1$-th dimensional *measure* of the cell, but its *direction* and orientation in space as well.

In general, we want $V(\not{C}^p)$ to simultaneously encode the direction of the cell $\not{C}^p$ (an element of the Grassmannian $\mathrm{Gr}_p(\mathbb{R}^n)$) and its oriented measure (a real number). A classical result of projective geometry (see section 2.4 of [111] for more information and [194] for a more basic introduction to Grassmannians) states that $\mathrm{Gr}_p(\mathbb{R}^n)$ is naturally embedded into the projective space $\mathrm{P}(\Lambda^p(\mathbb{R}^n))$ via the Plücker coordinates. Accounting for orientation and measure, we encode volumes as elements of $\Lambda(\mathbb{R}^n)$ the exterior algebra of $\mathbb{R}^n$. More precisely, in the planar case $V(\not{C}^p) \in \Lambda^p(\mathbb{R}^n)$ is co-linear to the volume form on $\mathbb{H}(\not{C}^p)$ denoted $\mathrm{vol}_{\mathbb{H}(\not{C}^p)}$, and $V(\not{C}^p) \cdot \mathrm{vol}_{\mathbb{H}(\not{C}^p)}$ denotes the p-th dimensional measure of $\not{C}^p$.

Remark. Note that $V(\not{C}^{n-1})$ is not strictly speaking what is commonly called the vector area of face $\not{C}^{n-1}$, which we denote $\mathbf{\Gamma}(\not{C}^{n-1})$. Indeed, $V(\not{C}^{n-1})$ is a $(n-1)$-form and not a vector (which we identify to a 1-form in our affine setting). This gap is bridged by the relationship $\mathbf{\Gamma}(\not{C}^{n-1}) = *^{-1}V(\not{C}^{n-1})$ where $*^{-1}$ is the inverse Hodge operator.

Proposition 1.7 (Expression of volumes and centroids of cells of a polygonal mesh)**.** The zeroth and first moment of a p-cell ($p \geq 1$) of an embedded polygonal mesh can be computed from those of lower dimensional cells with the following formulae:

$$V(\not{C}^p) = \frac{1}{p} \sum_{\not{C}^{p-1} \in \partial \not{C}^p} o(\not{C}^{p-1}, \not{C}^p)(\mathbf{x}(\not{C}^{p-1}) - \mathbf{x}(\not{C}^p)) \wedge V(\not{C}^{p-1}) \tag{1.10}$$

$$V(\not{C}^p) \otimes \mathbf{x}(\not{C}^p) = \frac{1}{p} \sum_{\not{C}^{p-1} \in \partial \not{C}^p} o(\not{C}^{p-1}, \not{C}^p)(\mathbf{x}(\not{C}^{p-1}) - \mathbf{x}(\not{C}^p)) \wedge V(\not{C}^{p-1}) \otimes \mathbf{x}(\not{C}^{p-1}) \tag{1.11}$$

Moreover, the volumes satisfy the following *closure* property:

$$\mathbf{0} = \sum_{\not{C}^{p-1} \in \partial \not{C}^p} o(\not{C}^{p-1}, \not{C}^p) V(\not{C}^{p-1}) \tag{1.12}$$

Proof. For $p \in [\![0, d]\!]$ and $\not{C}^p \in C$, we denote $\mathbb{H}(\not{C}^p)$ the affine hyperplane of dimension p that contains $\not{C}^p$ and $\mathbf{u}$ a uniform vector field. In $\mathbb{H}(\not{C}^p)$, we have $\mathrm{d}(\iota_{\mathbf{u}} \mathrm{vol}_{\mathbb{H}(\not{C}^p)}) = 0$. Thus,

$$\begin{aligned} 0 &= \int_{\not{C}^p} \mathrm{d}(\iota_{\mathbf{u}} \mathrm{vol}_{\mathbb{H}(\not{C}^p)}) = \int_{\partial \not{C}^p} \iota_{\mathbf{u}} \mathrm{vol}_{\mathbb{H}(\not{C}^p)} \\ &= \sum_{\not{C}^{p-1} \in \partial \not{C}^p} o(\not{C}^{p-1}, \not{C}^p) \int_{\not{C}^{p-1}} \iota_{\mathbf{u}} \mathrm{vol}_{\mathbb{H}(\not{C}^p)} \end{aligned}$$

An consequently,

$$\sum_{\not{C}^{p-1} \in \partial \not{C}^p} o(\not{C}^{p-1}, \not{C}^p) V(\not{C}^{p-1}) = \mathbf{0}$$

Which is the sought closure property.

The rest of the proof consists in applying Stokes's theorem to the integration of homogeneous monomials of increasing order. The affine structure of $\mathbb{R}^n$, more specifically the identification of $\mathbb{R}^n$ with its tangent spaces allows the definition of a vector field $\mathbf{x}$ given an affine frame, with point values corresponding to point position. Contraction with this vector field (denoted $\iota_{\mathbf{x}}$ in this work, sometimes κ in the relevant literature, see [214] and [10] for instance) is called the Koszul operator. The key point of the proof is to use the fact that if ω is a r-homogeneous p-form, *i.e.* if it satisfies $\omega(\alpha \mathbf{u}_1, \ldots, \alpha \mathbf{u}_p) = \alpha^r \omega(\mathbf{u}_1, \ldots, \mathbf{u}_p)$ for all $\alpha \in \mathbb{R}^+$ and $\mathbf{u}_i \in \mathbb{R}^n$, then $\mathcal{L}_{\mathbf{x}} = \mathrm{d}\iota_{\mathbf{x}} + \iota_{\mathbf{x}}\mathrm{d}$, the Lie derivative with respect to $\mathbf{x}$ satisfies:

$$(\mathrm{d}\iota_{\mathbf{x}} + \iota_{\mathbf{x}}\mathrm{d})\omega = (p + r)\omega \tag{1.13}$$

Let us first consider homogeneous polynomial p-forms of order zero. In $\mathbb{H}(\not{C}^p)$, we have $p \operatorname{vol}_{\mathbb{H}(\not{C}^p)} = (\mathrm{d}\iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} + \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)}\mathrm{d}) \operatorname{vol}_{\mathbb{H}(\not{C}^p)} = \mathrm{d}(\iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)})$. Thus, using Stokes's theorem, we have:

$$\begin{aligned} V(\not{C}^p) \cdot \operatorname{vol}_{\mathbb{H}(\not{C}^p)} &= \int_{\not{C}^p} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} = \frac{1}{p} \int_{\not{C}^p} \mathrm{d}\iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \\ &= \frac{1}{p} \int_{\partial \not{C}^p} \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \\ &= \frac{1}{p} \sum_{\not{C}^{p-1} \in \partial \not{C}^p} o(\not{C}^{p-1}, \not{C}^p) \int_{\not{C}^{p-1}} \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \end{aligned}$$

Since each $\not{C}^{p-1}$ is assumed to be embedded in a $p-1$-dimensional space, the term $\iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)}$ is constant on $\mathbb{H}(\not{C}^{p-1}) \supset \not{C}^{p-1}$, and its value is $\iota_{\mathbf{x}(\not{C}^{p-1})-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)}$. The integral hence reads:

$$\begin{aligned} \int_{\not{C}^{p-1}} \iota_{\mathbf{x}(\not{C}^{p-1})-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} &= \iota_{\mathbf{x}(\not{C}^{p-1})-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \cdot V(\not{C}^{p-1}) \\ &= (\mathbf{x}(\not{C}^{p-1}) - \mathbf{x}(\not{C}^p)) \wedge V(\not{C}^{p-1}) \cdot \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \end{aligned}$$

And finally, we have:

$$\begin{aligned} V(\not{C}^p) &= \left(\int_{\not{C}^p} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \right) \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \\ &= \frac{1}{p} \sum_{\not{C}^{p-1} \in \partial \not{C}^p} o(\not{C}^{p-1}, \not{C}^p)(\mathbf{x}(\not{C}^{p-1}) - \mathbf{x}(\not{C}^p)) \wedge V(\not{C}^{p-1}) \end{aligned}$$

Now we consider first order homogeneous monomials in the variable $\mathbf{x} - \mathbf{x}(\not{C}^p)$. They have the general form $\mathbf{u} \cdot (\mathbf{x} - \mathbf{x}(\not{C}^p))$, where $\mathbf{u}$ denotes a constant vector field. Using the Koszul operator contracting property $(\mathrm{d}\iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} + \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)}\mathrm{d})(\mathbf{u} \cdot (\mathbf{x} - \mathbf{x}(\not{C}^p)) \operatorname{vol}_{\mathbb{H}(\not{C}^p)}) = (p+1)\mathbf{u} \cdot (\mathbf{x} - \mathbf{x}(\not{C}^p)) \operatorname{vol}_{\mathbb{H}(\not{C}^p)}$, we get:

$$\begin{aligned} \mathbf{0} &= (p+1) \int_{\not{C}^p} \mathbf{u} \cdot (\mathbf{x} - \mathbf{x}(\not{C}^p)) \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \\ &= \int_{\not{C}^p} (\mathrm{d}\iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} + \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)}\mathrm{d})(\mathbf{u} \cdot (\mathbf{x} - \mathbf{x}(\not{C}^p)) \operatorname{vol}_{\mathbb{H}(\not{C}^p)}) \\ &= \int_{\partial \not{C}^p} \mathbf{u} \cdot (\mathbf{x} - \mathbf{x}(\not{C}^p)) \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} + \underbrace{\int_{\not{C}^p} \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} (\mathbf{u} \wedge \operatorname{vol}_{\mathbb{H}(\not{C}^p)})}_{=0} \\ &= \int_{\partial \not{C}^p} \mathbf{u} \cdot \mathbf{x} \; \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} - \int_{\partial \not{C}^p} \mathbf{u} \cdot \mathbf{x}(\not{C}^p) \; \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \\ &= \int_{\partial \not{C}^p} \mathbf{u} \cdot \mathbf{x} \; \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \; - \mathbf{u} \cdot \mathbf{x}(\not{C}^p) \int_{\partial \not{C}^p} \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \\ &= \int_{\partial \not{C}^p} \mathbf{u} \cdot \mathbf{x} \; \iota_{\mathbf{x}-\mathbf{x}(\not{C}^p)} \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \; - p\mathbf{u} \cdot \mathbf{x}(\not{C}^p) V(\not{C}^p) \cdot \operatorname{vol}_{\mathbb{H}(\not{C}^p)} \end{aligned}$$

Hence, we can express the centroid of $\not{C}^p$ as a function of the centroids of its boundary sub-cells:

$$\begin{aligned}
\mathbf{u}\cdot\mathbf{x}(\mathcal{C}^p)V(\mathcal{C}^p)\cdot\mathrm{vol}_{\mathbb{H}(\mathcal{C}^p)} &= \frac{1}{p}\int_{\partial\mathcal{C}^p}\mathbf{u}\cdot\mathbf{x}\;\iota_{\mathbf{x}-\mathbf{x}(\mathcal{C}^p)}\,\mathrm{vol}_{\mathbb{H}(\mathcal{C}^p)}\\
&= \frac{1}{p}\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\int_{\mathcal{C}^{p-1}}\mathbf{u}\cdot\mathbf{x}\;\underbrace{\iota_{\mathbf{x}-\mathbf{x}(\mathcal{C}^p)}\,\mathrm{vol}_{\mathbb{H}(\mathcal{C}^p)}}_{=\text{cst on }\mathcal{C}^{p-1}}\\
&= \frac{1}{p}\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\mathbf{u}\cdot\mathbf{x}(\mathcal{C}^{p-1})\iota_{\mathbf{x}(\mathcal{C}^{p-1})-\mathbf{x}(\mathcal{C}^p)}\,\mathrm{vol}_{\mathbb{H}(\mathcal{C}^p)}\cdot V(\mathcal{C}^{p-1})\\
&= \frac{1}{p}\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\mathbf{u}\cdot\mathbf{x}(\mathcal{C}^{p-1})(\mathbf{x}(\mathcal{C}^{p-1})-\mathbf{x}(\mathcal{C}^p))\wedge V(\mathcal{C}^{p-1})\cdot\mathrm{vol}_{\mathbb{H}(\mathcal{C}^p)}
\end{aligned}$$

Hence, we finally get the sought result:

$$V(\mathcal{C}^p)\otimes\mathbf{x}(\mathcal{C}^p) = \frac{1}{p}\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)(\mathbf{x}(\mathcal{C}^{p-1})-\mathbf{x}(\mathcal{C}^p))\wedge V(\mathcal{C}^{p-1})\otimes\mathbf{x}(\mathcal{C}^{p-1})$$

□

Remark. Proposition 1.7 shows that in the planar case, the first moment of a p-cell decomposes into $\mathbf{m}_1(\mathcal{C}^p) = V(\mathcal{C}^p)\otimes\mathbf{x}(\mathcal{C}^p)$. Moreover, since $V(\mathcal{C}^p)$ is proportional to $\mathrm{vol}_{\mathbb{H}(\mathcal{C}^p)}$, it is a decomposable p-form. These decomposability properties of zeroth and first moment no longer stand in the non-planar case as developed in section 1.2.4.

The main shortcoming of formulae (1.10) and (1.11) is that they do not provide an explicit way of computing volumes and centroids of p-cells knowing those of $(p-1)$-cells. For this reason, we use the following iterative algorithm for practical computations:

Proposition 1.8 (Computation of volumes and centroids of cells of a polygonal mesh)**.** The following algorithm computes the volumes and centroids of p-cells in a polygonal mesh: For $p=0$, the centroid of a node is the (supposed known) position of the node $\mathbf{x}(\mathcal{C}^0)$, and its 0-th dimensional volume is 1. For $p>0$, we proceed iteratively: At step $k=0$, initialize $\mathbf{x}^0(\mathcal{C}^p)$ with any point of $\mathbb{H}(\mathcal{C}^p)$. For instance:

$$\mathbf{x}^0(\mathcal{C}^p) = \frac{\displaystyle\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p}\|V(\mathcal{C}^{p-1})\|\;\mathbf{x}(\mathcal{C}^{p-1})}{\displaystyle\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p}\|V(\mathcal{C}^{p-1}\|}$$

Then, iterate on k as:

$$\begin{cases}
V^k(\mathcal{C}^p) = \dfrac{1}{p}\displaystyle\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)((\mathbf{x}(\mathcal{C}^{p-1})-\mathbf{x}^{k-1}(\mathcal{C}^p))\wedge V(\mathcal{C}^{p-1})) & (1.14)\\
\mathbf{x}^k(\mathcal{C}^p)-\mathbf{x}^{k-1}(\mathcal{C}^p) = \dfrac{1}{(p+1)\|V^k(\mathcal{C}^p)\|^2}\displaystyle\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p) & (1.15)\\
\qquad\big((\mathbf{x}(\mathcal{C}^{p-1})-\mathbf{x}^{k-1}(\mathcal{C}^p))\wedge V(\mathcal{C}^{p-1})\big)\cdot V^k(\mathcal{C}^p)(\mathbf{x}(\mathcal{C}^{p-1})-\mathbf{x}^{k-1}(\mathcal{C}^p))
\end{cases}$$

The proof of proposition 1.7 straightforwardly adapts to show that at the end of iteration $k=1$, the correct values are already computed. Nevertheless, the formula still has a practical

purpose as we experienced that further iterations could enhance the numerical accuracy of the formula in some ill-conditioned cases. Indeed, only differences of coordinates appear in formula (1.15), which can alleviate massive cancellation issues.

The formulae and their proofs developed in this section are valid as long as we deal with polygonal meshes. However, if p-cells of the mesh do not span a p-dimensional affine subspace, the formulae are still computable, but their interpretation is unclear. In spite of this difficulty, usage of non-planar cells is a widespread practice in many areas of numerical simulation, and not having to make the planar assumption would give more flexibility to generate meshes. Furthermore, we develop in section 3.7.2 a set of numerical methods requiring the availability of quantities such as the derivative of volumes and centroids of cells of the mesh with respect to node positions. Since nodes should in these methods not be constrained to move in a particular strict subspace of $\mathbb{R}^n$, we need to make sense of the notion of volume and centroid of cells, even in the non-planar case. Section 1.2.4 address this issue using the barycentric subdivision of the cell-tuple structure. The next section states a few preliminary results.

1.2.3 Integration on simplicial complexes

This section gives closed form expressions of volumes and moments of any order of simplices.

Proposition 1.9 (Expression of the volume a simplex)**.**
If S^p is a p-simplex with nodes $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$, then its p-dimensional volume $V(S^p)$ is given by:

$$V(S^p) = \frac{1}{p!} \, (\mathbf{x}_{p-1} - \mathbf{x}_p) \wedge (\mathbf{x}_{p-2} - \mathbf{x}_{p-1}) \wedge \cdots \wedge (\mathbf{x}_1 - \mathbf{x}_0) \tag{1.16}$$

Proof. Geometrically speaking, S^p is the convex hull of its nodes $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$. Hence, by definition, $\mathbb{H}(S^p)$ is the affine space consisting of points of the form $\mathbf{x} + \mathbf{v}$ where $\mathbf{x} \in S^p$ and $\mathbf{v}$ is a linear combination of $(\mathbf{x}_1 - \mathbf{x}_0, \cdots, \mathbf{x}_p - \mathbf{x}_0)$. Hence, its volume form is proportional to $(\mathbf{x}_1 - \mathbf{x}_0) \wedge (\mathbf{x}_2 - \mathbf{x}_0) \wedge \cdots \wedge (\mathbf{x}_p - \mathbf{x}_0) = (\mathbf{x}_{p-1} - \mathbf{x}_p) \wedge (\mathbf{x}_{p-2} - \mathbf{x}_{p-1}) \wedge \cdots \wedge (\mathbf{x}_0 - \mathbf{x}_1)$. Given a choice of base of $\mathbb{H}(S^p)$, the classical formula for the volume of a simplex readily gives the sought result. □

Remark. This formula generalizes the well-known three dimensional formula for the volume of tetrahedra: Area $= \frac{1}{6}$ Height $\times$ Area of base . Indeed, if S^{p-1} is the face of S^p with nodes $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{p-1})$, then the formula for the volume of S^p can be recast using the volume of S^{p-1} as:

$$V(S^p) = \frac{1}{p}(\mathbf{x}_{p-1} - \mathbf{x}_p) \wedge V(S^{p-1}) \tag{1.17}$$

This effectively gives a recurrence relation to compute the volume of a simplex from the volume of one of its faces. Moreover, the formula remains true if we substitute $\mathbf{x}_{p-1}$ for any point of $\mathbb{H}(S^{p-1})$ because of the properties of the wedge product.

Definition 1.15 (Barycentric coordinates in a simplex)**.**
Any point $\mathbf{x}$ of a non-degenerate p-simplex S^p with nodes $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$ can be uniquely written as a non-negative linear combination of nodes with coefficients summing to unity. In other words, there exists $p+1$ affine functions $N_k : \mathbb{H}(S^p) \to \mathbb{R}$ called *barycentric coordinates* satisfying the following properties:

Partition of unity or consistency of order zero:

$$\forall \, \mathbf{x} \in \mathbb{H}(S^p), \qquad \sum_{k=0}^{p} N_k(\mathbf{x}) = 1 \tag{1.18}$$

Consistency of order one:

$$\forall\ \mathbf{x} \in \mathbb{H}(S^p), \qquad \sum_{k=0}^{p} N_k(\mathbf{x})\mathbf{x}_i = \mathbf{x} \tag{1.19}$$

Non-negativeness:

$$\forall\ \mathbf{x} \in S^p, \forall\ k \in [\![0,p]\!], \qquad N_k(\mathbf{x}) \geq 0 \tag{1.20}$$

Barycentric coordinates play a huge role in the development of FEMs, where they are used piecewise to generate discrete functions spaces on simplicial meshes. We use these shape functions in section 2.7.1 to define a set mesh-based compatible operators.

Proposition 1.10 (Expression of the barycentric coordinates)**.**
Let S^{p-1} be a face of a p-dimensional simplex S^p with nodes $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$. Then, for all $\mathbf{x}_F \in \mathbb{H}(S^{p-1})$ and all point $\mathbf{x}$ inside the convex hull of S^p, the barycentric coordinates of node p has the following expression:

$$N_p(\mathbf{x}) = \frac{1}{p}\left|\frac{\big((\mathbf{x}_F - \mathbf{x}) \wedge V(S^{p-1})\big) \cdot V(S^p)}{\|V(S^p)\|^2}\right| \tag{1.21}$$

Proof. This function is indeed affine. For $i \in [\![0, p-1]\!]$, if we have $N_p(\mathbf{x}_i) = 0$ since $\mathbf{x}_i$ is in $\mathbb{H}(S^{p-1})$. For $i = p$, expression (1.17) of the volume of a simplex proves that $N_p(\mathbf{x}_i) = 1$. □

The next proposition gives an explicit formula for the integration of product of powers of barycentric coordinates over a simplex, which extend long known results in the finite element community for low dimensions (see [86] for instance).

Proposition 1.11 (Integration of products of barycentric coordinates over a simplex)**.**
Given a non-degenerate p-simplex S^p with nodes $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$ and exponents $(i_0, \dots, i_p) \in \mathbb{N}^{p+1}$, the integral of the barycentric monomial $\prod_{k=0}^{p} N_k(\mathbf{x})^{i_k}$ over S^p can be expressed in the following closed-form:

$$\int_{S^p} \prod_{k=0}^{p} N_k(\mathbf{x})^{i_k}\ \mathrm{vol}_{\mathbb{H}(S^p)} = \frac{p!\prod_{k=0}^{p} i_k!}{(p + \sum_{k=0}^{p} i_k)!}\|V(S^p)\| \tag{1.22}$$

Proof. The proof is a consequence of a special case of Brion's formula discovered independently in [18, 44, 139, 188]. Let us first consider the special case where $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$ is an orthonormal affine frame of $\mathbb{R}^p$. Then, from corollary 3 of [109], we know that the moments on S^p generate the following function:

$$\begin{aligned} F_{S^p}(\mathbf{u}) &\overset{\text{def}}{=} \sum_{I=(i_1,\cdot,i_p)\in\mathbb{N}^p} \frac{(p + \sum_{k=1}^{p} i_k)!}{\prod_{k=1}^{p} i_k!}\left(\int_{\mathbf{x}\in S^p} \prod_{k=1}^{p} x_k^{i_k}\ \mathrm{vol}_{\mathbb{R}^p}\right)\prod_{k=1}^{p} u_k^{i_k} \\ &= \frac{1}{\prod_{k=1}^{p}(1-u_k)} \end{aligned}$$

Several differentiations of the above formula with respect to $\mathbf{u}$ and evaluation at $\mathbf{u} = \mathbf{0}$ yield:

$$\int_{\mathbf{x}\in S^p} \prod_{k=1}^{p} x_k^{i_k}\ \mathrm{vol} = \frac{p!\prod_{k=1}^{p} i_k!}{(p + \sum_{k=1}^{p} i_k)!}$$

The result then follows from a change of variable in the integral. □

This result can be used to compute the integral of any polynomial over a simplex. For instance, let us give the following important result:

Proposition 1.12 (Centroid of a simplex)**.**
Let S^p denote a p-simplex with nodes $(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p)$. Its centroid can be expressed as:

$$\mathbf{x}(S^p) = \frac{1}{p+1} \sum_{k=0}^{p} \mathbf{x}_k \tag{1.23}$$

Proof. Using proposition 1.11, the proof is a simple computation:

$$\begin{aligned} \|V(S^p)\| \mathbf{u} \cdot \mathbf{x}(S^p) &= \int_{S^p} \mathbf{u} \cdot \mathbf{x} \, \mathrm{vol}_{\mathbb{H}(S^p)} \\ &= \sum_{k=0}^{p} \mathbf{u} \cdot \mathbf{x}_k \int_{S^p} N_k(\mathbf{x}) \, \mathrm{vol}_{\mathbb{H}(S^p)} \\ &= \frac{\|V(S^p)\|}{p+1} \mathbf{u} \cdot \sum_{k=0}^{p} \mathbf{x}_k \end{aligned}$$

In the next section, we use the formulae for the computation of metric quantities on simplices to generalize the algorithm given in proposition 1.8 to the non-planar case.

1.2.4 Nodal positions fully define the geometry of meshes with non-planar cells

In this section, we generalize formulae (1.10) and (1.11) to the computation of volumes and centroids of non-planar cells. In particular, we show that the geometry of such meshes can be unequivocally and intrinsically defined as long as node positions are prescribed and a simple verifiable non-degeneracy condition is met. This representation is made explicit through a cell-wise affine immersion of the barycentric subdivision of the cell-tuple structure, and an efficient algorithm to compute it is described.

We suppose given the position $\mathbf{x}(\not{c}^0)$ of every node $\not{c}^0$ but we ***do not*** assume that nodes of a same cell $\not{c}^p$ with $p \in [\![1, p]\!]$ lie in the same hyperspace of dimension p. From section 1.2.3, we know that simplicial embeddings are easy to work with, since any position of the $p+1$ nodes of a p-simplex unambiguously defines an affine endomorphism of $\mathbb{R}^p$. Moreover, the formulae for the volume and centroid of simplices are quite simple. We recall from definition 1.4 that tuples have the following combinatorial interpretation: they are cells of a simplicial complex, whose nodes correspond to cells of the original cell-tuple structure. Geometrically speaking, it is reasonable to define the embedding of our mesh piecewise on tuples as sketched on the right hand side of figure 1.4: let us simply have tuple node positions match cell centroids. Using propositions 1.9 and 1.12, this choice translates into the following relations yielding partial tuple volumes and centroids in terms of cell volumes and centroids:

$$V(t^p) = \frac{1}{p!} \left(\mathbf{x}(\not{c}^{t^{p-1}}) - \mathbf{x}(\not{c}^{t^p})\right) \wedge \left(\mathbf{x}(\not{c}^{t^{p-2}}) - \mathbf{x}(\not{c}^{t^{p-1}})\right) \wedge \cdots \wedge \left(\mathbf{x}(\not{c}^{t^1}) - \mathbf{x}(\not{c}^{t^0})\right) \tag{1.24}$$

$$\mathbf{x}(t^p) = \frac{1}{p+1} \sum_{k=0}^{p} \mathbf{x}(\not{c}^{t^k}) \tag{1.25}$$

Now, we use the linearity of the integral to express both the zeroth and first order moments of $\not{c}^p$ in terms of those of its partial tuples:

$$\mathbf{m}_0(\mathcal{C}^p) = V(\mathcal{C}^p) = \sum_{t^p \supset \mathcal{C}^p} o_p(t^p) V(t^p) \tag{1.26}$$

$$\mathbf{m}_1(\mathcal{C}^p) = \sum_{t^p \supset \mathcal{C}^p} o_p(t^p) V(t^p) \otimes \mathbf{x}(t^p) \tag{1.27}$$

In the non-degenerate case, the cell centroid is then expressed as:

$$\mathbf{x}(\mathcal{C}^p) = \frac{\mathbf{m}_0(\mathcal{C}^p) \cdot \mathbf{m}_1(\mathcal{C}^p)}{\|\mathbf{m}_0(\mathcal{C}^p)\|^2} \tag{1.28}$$

The next theorem establishes the consistency of these relations and provides a practical algorithm to compute their values:

Proposition 1.13 (Computation of volumes and centroids of non-planar cells)**.**
In a cell-tuple structure $\mathcal{A}_{\mathcal{CC}}$ with specified nodal coordinates, expressions (1.24) to (1.28) are consistent in $\Lambda(\mathbb{R}^n)$. They always uniquely specify cell volumes and first order moments. In cases where the non-degeneracy condition $V(\mathcal{C}^p) \neq \mathbf{0}$ is met, they uniquely specify cell centroids. Moreover, these metric quantities can effectively be computed by the following iterative algorithm, in which the notion of cell-tuple does not appear: For $p = 0$, the centroid of a node is the (supposed known) position of the node $\mathbf{x}(\mathcal{C}^0)$, and its 0-th dimensional volume is 1. For $p > 0$, we proceed iteratively: At step $k = 0$, initialize $\mathbf{x}^0(\mathcal{C}^p)$ with any point of $\mathbb{R}^n$. For instance:

$$\mathbf{x}^0(\mathcal{C}^p) = \frac{\sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} \|V(\mathcal{C}^{p-1})\| \, \mathbf{x}(\mathcal{C}^{p-1})}{\sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} \|V(\mathcal{C}^{p-1})\|}$$

Then, iterate on k as:

$$\begin{cases} V^k(\mathcal{C}^p) = \dfrac{1}{p} \displaystyle\sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} o(\mathcal{C}^{p-1}, \mathcal{C}^p)((\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}^{k-1}(\mathcal{C}^p)) \wedge V(\mathcal{C}^{p-1})) & (1.29) \\ \mathbf{x}^k(\mathcal{C}^p) - \mathbf{x}^{k-1}(\mathcal{C}^p) = \dfrac{1}{(p+1)\|V^k(\mathcal{C}^p)\|^2} \displaystyle\sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} o(\mathcal{C}^{p-1}, \mathcal{C}^p) & (1.30) \\ \qquad V^k(\mathcal{C}^p) \cdot (\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}^{k-1}(\mathcal{C}^p)) \wedge ((\mathbf{m}_1(\mathcal{C}^{p-1}) - V(\mathcal{C}^{p-1}) \otimes \mathbf{x}^{k-1}(\mathcal{C}^p))) \end{cases}$$

This procedure is the generalization of that presented in the planar case (proposition 1.8). It converges as $k \to +\infty$, and the limit values are consistent with expressions (1.24) to (1.27). Finally, the volumes still satisfy the closure property:

$$\mathbf{0} = \sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} o(\mathcal{C}^{p-1}, \mathcal{C}^p) V(\mathcal{C}^{p-1}) \tag{1.31}$$

Proof. We proceed inductively on the cell dimension p. For $p = 0$, there is nothing to prove. For $p = 1$, edges have exactly two nodes, which can always be embedded in a 1-dimensional affine subspace, everything thus follows from the planar case (proposition 1.8). Let us assume that the proposition holds for all dimensions until rank $p - 1 \geq 1$. Then, let us first prove the closure property:

$$
\begin{aligned}
&\sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)V(\phi^{p-1}) \\
&= \frac{1}{p}\sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)\sum_{\phi^{p-2}\in\partial\phi^{p-1}} o(\phi^{p-2},\phi^{p-1})((\mathbf{x}(\phi^{p-2})-\mathbf{x}(\phi^{p-1}))\wedge V(\phi^{p-2})) \\
&= \frac{1}{p}\Bigg(\sum_{\phi^{p-2}\in\partial\phi^p}\mathbf{x}(\phi^{p-2})\wedge V(\phi^{p-2})\underbrace{\sum_{\substack{\phi^{p-1}\mid \phi^{p-1}\in\partial\phi^p\\ \phi^{p-2}\in\partial\phi^{p-1}}} o(\phi^{p-2},\phi^{p-1})o(\phi^{p-1},\phi^p)}_{=\mathbf{0}\text{ since }\partial_{p-1}\circ\partial_p=0\text{ see proposition 1.5}} \\
&\qquad -\sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)\mathbf{x}(\phi^{p-1})\wedge\underbrace{\sum_{\phi^{p-2}\in\partial\phi^{p-1}} o(\phi^{p-2},\phi^{p-1})V(\phi^{p-2})}_{=\mathbf{0}\text{ by induction hypothesis}}\Bigg) = \mathbf{0}
\end{aligned}
$$

Hence, the closure property holds for a p-cell ϕ^p. Let us now examine the convergence of the sequence $V^k(\phi^p)$:

$$
\begin{aligned}
p\left(V^k(\phi^p)-V^{k-1}(\phi^p)\right) &= \sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)(\mathbf{x}^{k-1}(\phi^p)-\mathbf{x}^{k-2}(\phi^p)\wedge V(\phi^{p-1})) \\
&= (\mathbf{x}^{k-1}(\phi^p)-\mathbf{x}^{k-2}(\phi^p))\wedge\sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)V(\phi^{p-1}) \\
&= \mathbf{0}
\end{aligned}
$$

Hence, the sequence $V^k(\phi^p)$ is constant. It trivially converges towards its initial value, which we now denote $V(\phi^p)$. Let us now examine the convergence of the sequence $\mathbf{x}(\phi^p)$:

$$
\begin{aligned}
&(p+1)\|V(\phi^p)\|^2\left(\mathbf{x}^k(\phi^p)-\mathbf{x}^{k-1}(\phi^p)\right) \\
&= \sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)V(\phi^p)\cdot(\mathbf{x}(\phi^{p-1})-\mathbf{x}^{k-1}(\phi^p))\wedge(\mathbf{m}_1(\phi^{p-1})-V(\phi^{p-1})\otimes\mathbf{x}^{k-1}(\phi^p)) \\
&= \sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)V(\phi^p)\cdot(\mathbf{x}^{k-2}(\phi^p)-\mathbf{x}^{k-1}(\phi^p))\wedge(\mathbf{m}_1(\phi^{p-1})-V(\phi^{p-1})\otimes\mathbf{x}^{k-2}(\phi^p)) \\
&+ \sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)V(\phi^p)\cdot(\mathbf{x}(\phi^{p-1})-\mathbf{x}^{k-2}(\phi^p))\wedge(\mathbf{m}_1(\phi^{p-1})-V(\phi^{p-1})\otimes\mathbf{x}^{k-2}(\phi^p)) \\
&+ \sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)V(\phi^p)\cdot(\mathbf{x}(\phi^{p-1})-\mathbf{x}^{k-1}(\phi^p))\wedge V(\phi^{p-1})\ (\mathbf{x}^{k-2}(\phi^p)-\mathbf{x}^{k-1}(\phi^p)) \\
&= \|V(\phi^p)\|^2(\mathbf{x}^{k-1}(\phi^p)-\mathbf{x}^{k-2}(\phi^p))+ \\
&\qquad V(\phi^p)\cdot\left((\mathbf{x}^{k-2}(\phi^p)-\mathbf{x}^{k-1}(\phi^p))\wedge\sum_{\phi^{p-1}\in\partial\phi^p} o(\phi^{p-1},\phi^p)\mathbf{m}_1(\phi^{p-1})\right)
\end{aligned}
$$

Hence as a result of Hadamard-Schwartz inequalities (see Corollary 4.1 of [122] in particular), we have the following bound:

$$\|\mathbf{x}^k(\mathcal{C}^p) - \mathbf{x}^{k-1}(\mathcal{C}^p)\| \leq$$
$$\frac{1}{(p+1)}\left(1 + \frac{1}{\|V(\mathcal{C}^p)\|}\left\|\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\mathbf{m}_1(\mathcal{C}^{p-1})\right\|\right)\|\mathbf{x}^{k-1}(\mathcal{C}^p) - \mathbf{x}^{k-2}(\mathcal{C}^p)\|$$

We want to tightly bound the oriented sum of first order boundary moments. Using the consistency with the barycentric decomposition at dimension $p-1$, we have:

$$\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\mathbf{m}_1(\mathcal{C}^{p-1}) = \sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\sum_{t^{p-1}\ni\mathcal{C}^p} o(t^{p-1})V(t^{p-1})\otimes\mathbf{x}(t^{p-1})$$

Its projection onto $\Lambda^{p-1}(\mathbb{R})$ is simply expressed using the fact that $\mathbf{x}(t^{p-1}) - \mathbf{x}(\mathcal{C}^{p-1}) \in \mathbb{H}(t^{p-1})$:

$$\begin{aligned}
&\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\sum_{t^{p-1}\ni\mathcal{C}^p} o(t^{p-1})V(t^{p-1})\wedge\mathbf{x}(t^{p-1})\\
&\qquad = \pm\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\sum_{t^{p-1}\ni\mathcal{C}^p} o(t^{p-1})\mathbf{x}(t^{p-1})\wedge V(t^{p-1})\\
&\qquad = \pm\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\sum_{t^{p-1}\ni\mathcal{C}^p} o(t^{p-1})\mathbf{x}(\mathcal{C}^{p-1})\wedge V(t^{p-1}) = \pm V(\mathcal{C}^p)
\end{aligned}$$

Hence, we have $\|V(\mathcal{C}^p)\| \leq \left\|\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\mathbf{m}_1(\mathcal{C}^{p-1})\right\|$ and finally,

$$\|\mathbf{x}^k(\mathcal{C}^p) - \mathbf{x}^{k-1}(\mathcal{C}^p)\| \leq \frac{2}{(p+1)}\|\mathbf{x}^{k-1}(\mathcal{C}^p) - \mathbf{x}^{k-2}(\mathcal{C}^p)\|$$

Since $p \geq 2$, we have $\frac{2}{p+1} < 1$, which proves the convergence of the iterative procedure, and the proper definition of the limit value $\mathbf{x}(\mathcal{C}^p)$. This value is characterized by the following relationship:

$$\mathbf{x}(\mathcal{C}^p) = \frac{1}{p\|V(\mathcal{C}^p)\|^2}V(\mathcal{C}^p)\cdot\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)(\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}(\mathcal{C}^p))\wedge\mathbf{m}_1(\mathcal{C}^{p-1})$$

We now prove that volumes and first order moments computed with this methods agree with the barycenter decomposition:

$$\begin{aligned}
\sum_{t^p\ni\mathcal{C}^p} o_p(t^p)V(t^p) &= \frac{1}{p}\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)\sum_{t^{p-1}\ni\mathcal{C}^{p-1}} o_{p-1}(t^{p-1})(\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}(\mathcal{C}^p))\wedge V(t^{p-1})\\
&= \frac{1}{p}\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)(\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}(\mathcal{C}^p))\wedge\sum_{t^{p-1}\ni\mathcal{C}^{p-1}} o_{p-1}(t^{p-1})V(t^{p-1})\\
&= \frac{1}{p}\sum_{\mathcal{C}^{p-1}\in\partial\mathcal{C}^p} o(\mathcal{C}^{p-1},\mathcal{C}^p)(\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}(\mathcal{C}^p))\wedge V(\mathcal{C}^{p-1})\\
&= V(\mathcal{C}^p)
\end{aligned}$$

Similarly for the first order moments:

$$\begin{aligned}
\mathbf{m}_1(\mathcal{C}^p) &= \sum_{t^p \ni \mathcal{C}^p} o_p(t^p) V(t^p) \otimes \mathbf{x}(t^p) \\
&= \frac{1}{p(p+1)} \sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} o(\mathcal{C}^{p-1}, \mathcal{C}^p) \\
&\qquad \sum_{t^{p-1} \ni \mathcal{C}^{p-1}} o_{p-1}(t^{p-1}) (\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}(\mathcal{C}^p)) \wedge V(t^{p-1}) \otimes (p\mathbf{x}(t^{p-1}) + \mathbf{x}(\mathcal{C}^p)) \\
&= \frac{1}{p+1} \left(V(\mathcal{C}^p) \otimes \mathbf{x}(\mathcal{C}^p) + \sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} o(\mathcal{C}^{p-1}, \mathcal{C}^p) (\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}(\mathcal{C}^p)) \wedge \mathbf{m}_1(\mathcal{C}^{p-1}) \right)
\end{aligned}$$

Consequently,

$$\begin{aligned}
\frac{V(\mathcal{C}^p) \cdot \mathbf{m}_1(\mathcal{C}^p)}{\|V(\mathcal{C}^p)\|^2} &= \frac{1}{p+1} \Bigg(\mathbf{x}(\mathcal{C}^p) + \\
&\qquad \frac{1}{\|V(\mathcal{C}^p)\|^2} V(\mathcal{C}^p) \cdot \sum_{\mathcal{C}^{p-1} \in \partial \mathcal{C}^p} o(\mathcal{C}^{p-1}, \mathcal{C}^p) (\mathbf{x}(\mathcal{C}^{p-1}) - \mathbf{x}(\mathcal{C}^p)) \wedge \mathbf{m}_1(\mathcal{C}^{p-1}) \Bigg) \\
&= \mathbf{x}(\mathcal{C}^p)
\end{aligned}$$

Which completes the proof by induction. □

To the author's knowledge, proposition 1.13 constitutes an original result, and its consequences are numerous. First of all, it means that cell connectivities and node positions fully specify a natural (possibly self-intersecting) mesh geometry, in a consistent fashion with its cell-tuple decomposition. This geometry is unambiguously defined and is computable in practice without any reference to the sub-cell structure (simplices of the barycentric decomposition) it is shown to describe.

Even assuming we can efficiently build a mesh with planar faces, the result keeps a practical interest since it allows the unequivocal definition of the derivative of these metric quantities with respect to *any* nodal displacement, whether it conserve the planarity assumption or not. And indeed, we use this fact in section 3.7.2, where these derivatives naturally appear in the context of background integration of meshless operators. These derivatives can be computed with a very similar iterative algorithm, or a non-iterative version akin to corollary 1.14.

Such a result has long been missing in the mesh community. For instance Garimella et al. stated in [101] (section 3.1): "It is burdensome to determine what the real shape of a curved polygonal face is (likely a minimal surface formed by the straight edges of the face) or where its centroid is". In practice, meshes with curved faces have been used for quite some time in simulation (see [42, 43]), but an interpretation of the precise geometry of cells or practical methods to compute their volumes and centroids have been little discussed. Even contributions (see [51, 73] for instance) which claim to be valid in arbitrary dimension on general (*i.e.* non-planar) meshes do not go through the trouble of detailing the computation – and incidentally the modeling – of volumes and derivatives for general meshes, but only for 2-dimensional meshes (whose faces are necessarily planar), and 3-dimensional tetrahedra and hexahedra.

Most authors who do examine these key points (see [145] for instance) end up with the same geometrical description as ours, but present it as an arbitrary choice rather than a consequence of the formula to compute volumes and centroids. Note that there might very

well exist other formulae that are exact for polygons with planar faces, that do not coincide with ours in the non-planar case, so that there is still an arbitrary element in the choice of modeling. Of course, the two choices (geometry and computation of metric quantities) need to be consistent.

In the most extreme cases, the geometry described by proposition 1.13 might not always be a polygonal mesh, or even a subdivided manifold. Indeed, cells can be flipped (*e.g.* so-called butterfly cells in 2-dimensions), negatively oriented, superposed *etc* ... Instead it is a well-defined signed simplicial decomposition of space. Additionally, cell volumes and centroids can be efficiently computed without any explicit reference to simplices or tuples.

The proposed algorithm has a proved linear convergence of rate $\frac{2}{p+1}$. We suspect the bounds given in the proof to be quite coarse as in practice, the observed rate is always very close to $\frac{1}{p+1}$.

The proof also gives better intuition about the meaning of the volume of a cell. In the planar case, we showed that $V(\not{c}^p)$ is decomposable (it is proportional to $\text{vol}_{\mathbb{H}(\not{c}^p)}$, which is decomposable). This is not true anymore in the general non-planar case as $V(\not{c}^p)$ might not be decomposable. Consequently, $V(\not{c}^p)$ cannot in general be associated to an averaged p-dimensional linear subspace of $\mathbb{R}^n$. This situation cannot happen for $n \leq 3$ though since 0-forms, 1-forms, $n-1$-forms and n-forms are always decomposable, so it might be difficult to apprehend. Similarly, the first order moment always decomposes into $\mathbf{m}_1(\not{c}^p) = V(\not{c}^p) \otimes \mathbf{x}(\not{c}^p)$, and this is no longer true in the non-planar case.

Corollary 1.14 (Non-iterative method for the computation of volumes and centroids)**.** The volumes and centroids of cells of a possibly non-planar mesh satisfy the following limit equality reminiscent of the planar case (expressions (1.10) and (1.11)):

$$V(\not{c}^p) = \frac{1}{p} \sum_{\not{c}^{p-1} \in \partial \not{c}^p} o(\not{c}^{p-1}, \not{c}^p)(\mathbf{x}(\not{c}^{p-1}) - \mathbf{x}(\not{c}^p)) \wedge V(\not{c}^{p-1}) \tag{1.32}$$

$$\|V(\not{c}^p)\|^2 \mathbf{x}(\not{c}^p) = \frac{1}{p} V(\not{c}^p) \cdot \sum_{\not{c}^{p-1} \in \partial \not{c}^p} o(\not{c}^{p-1}, \not{c}^p)(\mathbf{x}(\not{c}^{p-1}) - \mathbf{x}(\not{c}^p)) \wedge \mathbf{m}_1(\not{c}^{p-1}) \tag{1.33}$$

This inspires a non-iterative algorithm for the computation of volumes and centroids. To compute volumes, simply use expression (1.32) with any value for $\mathbf{x}(\not{c}^p)$ ($\mathbf{0}$ for instance). Indeed, expression (1.32) is non-iterative in nature as the computed value is independent from the actual value of $\mathbf{x}(\not{c}^p)$. For centroids, re-write expression (1.33) as a $d \times d$ linear system and solve with your favorite linear solver. Let us note that as in the linear case (proposition 1.8), a few iterations might avoid or alleviate precision loss due to massive cancellation.

Proof. The proof of proposition 1.13 actually shows that the following linear operator is invertible in $\mathrm{M}_d(\mathbb{R})$:

$$\mathbf{v} \to \|V(\not{c}^p)\|^2 \mathbf{v} + V(\not{c}^p) \cdot \mathbf{v} \wedge \frac{1}{p} \sum_{\not{c}^{p-1} \in \partial \not{c}^p} o(\not{c}^{p-1}, \not{c}^p) \mathbf{m}_1(\not{c}^{p-1})$$

The algorithm described in the proof actually amounts to computing the inverse of this operator with a standard convergent splitting (see [234] for more information about splitting methods in numerical linear algebra). □

1.3 Approximate integration on a mesh

This section is dedicated to the study of a few integration formulae using a mesh. These formulae are used in sections 2.7.3 and 3.7.2 for the construction of meshless operators. First,

we state the Bramble-Hilbert lemma in section 1.3.1, an important result of approximation theory in Sobolev spaces. Then, a few low-order formulae are proved in section 1.3.2 for the approximation of the integral of a function and of the gradient of a function, as well as the approximation of the derivative of the integral of a function with respect to the position of boundary nodes of the mesh.

1.3.1 Consistency of discrete operators and the Bramble-Hilbert lemma

In this section, we simply state one of the multiple forms of the Bramble-Hilbert lemma. More specifics about the assumptions of the theorem (Sobolev norms and regularity assumptions on the domain) as well as an explicit form of the bounding constant can be found in appendix A.

The relevance of te Bramble-Hilbert lemma is best understood when stated under the following operator form:

Lemma 1.15 (Operator form of the Bramble-Hilbert lemma)**.**
Let $\Omega \subset \mathbb{R}^n$ be a domain with a regularity condition and $\ell : W_p^k(\Omega) \to Y$ be a continuous linear operator. Suppose that ℓ vanishes on polynomials of maximal order $m-1$: $\ell(v) = 0 \; \forall \; v \in P_{m-1}$. Then there exists a constant C such that:

$$\|\ell(u)\|_Y \leq C\left(\frac{1}{D(\Omega)}\Omega, m\right) \; \|\ell\|_{\mathcal{L}(W_p^k(\Omega),Y)} \; D(\Omega)^{m-k}|u|_{W_p^m(\Omega)} \tag{1.34}$$

Proof. This is a straightforward combination of expression (A.4) and the continuity of ℓ. □

Under this form, the Bramble-Hilbert lemma is particularly useful to prove error bounds of numerical integration formulae. The sketch of such proofs is always the same: we want to use the Bramble-Hilbert lemma on the error functional ℓ. The hypothesis $\ell(v) = 0 \; \forall \; v \in P_{m-1}$ is a consistency requirement on the discrete operator. Convergence then follows from the regularity of Ω in a norm $|.|_{W_p^m(\Omega)}$for which ℓ is continuous. The exponent $m-k$ is called the order of accuracy of the method, since the transformation $\Omega \mapsto a\Omega$ changes the error bound by a factor a^{m-k}.

In the next section, we give several formulae for the approximate integration of a function using a mesh.

1.3.2 Miscellaneous low-order approximate integration formulae on a mesh

In this section, we use the previously computed metric quantities to establish approximate integration formulae using a mesh, which we exploit in sections 2.7.3 and 3.7.2 to effectively compute compatible meshless operators. These low order integration methods are well-known, but their proof is quite enlightening, especially as far as the topological structure of the mesh is concerned. This structure is precisely what enables us later on to transfer the compatibility properties of the exactly integrated meshless operators to the discrete level.

The convergence proofs detailed in this section are valid in the full dimension (*i.e.* $d = n$) with some classical assumptions on the mesh quality, but the first order consistency is retained even if the mesh is heavily distorted or if cells overlap.

1.3.2.1 Approximation of the integral of a function

The simplest case is that of a simplex because their number of vertices are ideally suited to write consistent affine approximations.

Proposition 1.16 (Nodal integration on a simplex)**.**
Let $\mathcal{S}$ denote a simplex with nodes $(\mathbf{x}_1, \dots, \mathbf{x}_n)$. The following expression is a first order consistent approximation to the integral of a function over $\mathcal{S}$ and its convergence is quadratic.

$$\fint_{\mathcal{S}}^{\text{Nodal}} f \overset{\text{def}}{=} \frac{V(\mathcal{S})}{n+1} \sum_{k \in \mathcal{S}} f(\mathbf{x}_k) \tag{1.35}$$

Proof. Proposition 1.12 readily entails that the formula is exact on linear polynomials. Continuity in the L^∞ norm is immediate, corollary 1.15 is thus readily applicable with $p = \infty$, $k = 0$ and $m = 2$. □

The following formula exploits the fact that the centroid of a cell is its first order moment: by definition, evaluation of an affine function at this point gives its mean value over the cell.

Proposition 1.17 (One point integration on a cell)**.**
Let ϕ be a cell of a mesh and $f : \phi \to \mathbb{R}$. The following one point integration formula aims at approximating the quantity $\int_\phi f \operatorname{vol}_\phi$:

$$\fint_{\phi}^{1\text{ pt}} f \overset{\text{def}}{=} V(\phi) f(\mathbf{x}(\phi)) \tag{1.36}$$

The above approximation is first order consistent and the convergence is quadratic.

Proof. By construction, the method is exact on first order polynomials: $\forall\, \mathbf{a} \in \mathbb{R}^n, \forall\, b \in \mathbb{R}$, $\int_\phi \mathbf{a} \cdot \mathbf{x} + b \operatorname{vol}_\phi = \fint_\phi^{1 \text{ point}} \mathbf{a} \cdot \mathbf{x} + b$. Hence, in order to use the Bramble-Hilbert lemma under the form given in corollary 1.15, we need to find a Sobolev space for which the error functional $\ell(f) \overset{\text{def}}{=} \int_\phi f \operatorname{vol}_\phi - \fint_\phi^{1 \text{ point}} f$ is continuous. The $L^\infty = W^0_\infty$ norm is the perfect candidate since we have: $\forall\, f \in L^\infty(\phi)$,

$$\begin{aligned}
|\ell(f)| &\leq \int_\phi |f| \operatorname{vol}_\phi + V(\phi) |f(\mathbf{x}(\phi))| \\
&\leq \sup_{\mathbf{x} \in \phi} |f(\mathbf{x})| \left(\int_\phi \operatorname{vol}_\phi + V(\phi) \right) \\
&= 2V(\phi) \|f\|_{W^0_\infty}
\end{aligned}$$

Hence corollary 1.15 is readily applicable with $p = \infty$, $k = 0$ and $m = 2$. □

1.3.2.2 Approximation of derivatives: Gauss-type gradients

The following formula for the integral of the gradient of a function exploits Stokes's theorem to avoid exact gradient evaluations. Instead, the function is evaluated on the boundary of the cell, and the vector areas of boundary faces provide a linear consistent approximation of the gradient. This technique is heavily used in FVM discretizations since a face separating two cells has opposite orientation with respect to each cell. The fluxes are thus discretized in such a way that global quantities are conserved. This feature translates into compatibility of the mesh-integrated meshless Smoothed Finite Element Method (SFEM) operators in section 2.7.3.

Definition 1.16 (Gauss gradient on a cell)**.**
Let ϕ be a cell of a mesh. Using Stokes's formula, we can write the integral of the divergence of a vector field $\mathbf{u} : \phi \to \mathbb{R}^d$ as its flux on the boundary $\partial\phi$:

$$\begin{aligned}\int_\phi \nabla \cdot \mathbf{u} \operatorname{vol}_\phi &= \int_{\partial\phi} \iota_{\mathbf{u}} \operatorname{vol}_{\partial\phi} \\ &= \sum_{\phi^{n-1}\in\partial\phi} o(\phi^{n-1},\phi) \int_{\phi^{n-1}} \iota_{\mathbf{u}} \operatorname{vol}_{\partial\phi} \\ &= \sum_{\phi^{n-1}\in\partial\phi} o(\phi^{n-1},\phi) \int_{\phi^{n-1}} \mathbf{u}\cdot\mathbf{n} \operatorname{vol}_{\partial\phi^{n-1}}\end{aligned} \tag{1.37}$$

Hence, any approximation of the flux of a vector field on the boundary faces of a cell translates to an approximation of the integral of the divergence of the vector field on the cell. This approximation technique is called "Gauss gradient" (see [223] for instance).

Definition 1.17 (One point integrated Gauss gradient)**.**
For instance, we can use the one point integration formula of proposition 1.17 on faces. Indeed the following formula is first order accurate for $\mathbf{u} \in W^1_\infty(\phi)$:

$$\oint_\phi \mathbb{\nabla} \cdot \mathbf{u} \overset{\text{Gauss 1pt}}{\overset{\text{def}}{=}} \sum_{\phi^{n-1}\in\partial\phi} o(\phi^{n-1},\phi)\mathbf{\Gamma}(\phi^{n-1})\cdot\mathbf{u}(\mathbf{x}(\phi^{n-1})) \tag{1.38}$$

Proof. Let us first check that the formula is exact on polynomials of maximum degree 1 on ϕ. For $\mathbf{u}(\mathbf{x}) = \mathrm{A}\cdot\mathbf{x}+\mathbf{b}$, we have on the one hand:

$$\begin{aligned}\int_\phi \nabla\cdot\mathbf{u}\operatorname{vol}_\phi &= \int_\phi \nabla\cdot(\mathrm{A}\cdot\mathbf{x}+\mathbf{b})\operatorname{vol}_\phi \\ &= \int_\phi \operatorname{Tr}(\mathrm{A})\operatorname{vol}_\phi \\ &= V(\phi)\operatorname{Tr}(\mathrm{A})\end{aligned}$$

And, on the other hand,

$$\begin{aligned}\oint_\phi \mathbb{\nabla}\cdot(\mathrm{A}\cdot\mathbf{x}+\mathbf{b}) &\overset{\text{Gauss 1pt}}{=} \sum_{\phi^{n-1}\in\partial\phi} o(\phi^{n-1},\phi)\mathbf{\Gamma}(\phi^{n-1})\cdot(\mathrm{A}\cdot\mathbf{x}(\phi^{n-1})+\mathbf{b}) \\ &= \operatorname{Tr}\left(\mathrm{A}\cdot\sum_{\phi^{n-1}\in\partial\phi} o(\phi^{n-1},\phi)\mathbf{\Gamma}(\phi^{n-1})\mathbf{x}(\phi^{n-1})^T\right) \\ &= \operatorname{Tr}(\mathrm{A}\cdot V(\phi)\mathrm{I}_d) \\ &= V(\phi)\operatorname{Tr}(\mathrm{A})\end{aligned}$$

Hence the two expressions are equal and the method is exact on linear functions. In order to use corollary 1.15, we still need to prove boundedness of the error functional:

$$\begin{aligned}\left\|\int_\phi \nabla\cdot\mathbf{u}\operatorname{vol}_\phi\right\| &\le \int_\phi \|\nabla\cdot\mathbf{u}\|\operatorname{vol}_\phi \\ &\le V(\phi)|\mathbf{u}|_{W^1_\infty}\end{aligned}$$

And for the discrete part,

$$\left\| \oint_{\partial\phi}^{\text{Gauss 1pt}} \nabla \cdot \mathbf{u} \right\| = \left\| \sum_{\phi^{n-1} \in \partial\phi} o(\phi^{n-1}, \phi) \mathbf{\Gamma}(\phi^{n-1}) \cdot (\mathbf{u}(\mathbf{x}(\phi^{n-1})) - \mathbf{u}(\mathbf{x}(\phi))) \right\|$$
$$\leq D(\phi) S(\phi) |\mathbf{u}|_{W^1_\infty}$$

Where $D(\phi)$ denotes the diameter of the cell and $S(\phi)$ the (non-algebraic) surface area of its boundary. We can apply corollary 1.15 with $p = \infty$, $k = 1$ and $m = 2$ if the ratio $\frac{D(\phi^n) S(\partial\phi^n)}{V(\phi^n)}$ is bounded from above. This corresponds to an additional regularity assumption on the mesh, complementary to the isoperimetric-isodiametric inequality (see [166]) saturated by balls:

$$V(\Omega) \leq \frac{1}{2n} D(\Omega) S(\partial\Omega) \tag{1.39}$$

Intuitively speaking, if cells of a sequence of meshes verify this additional requirement, their shape cannot increasingly deviate from that of a ball with refinement level. □

Remark. Gauss gradient approximations can be built with other approximations of boundary integrals. See for instance the discrete Gauss gradient of section 2.7.1

1.3.2.3 Derivative of integral with respect to node position

The example of the Gauss gradient developed in the last section is arguably the most well-known means of exploiting the topological structure (the cell-face structure in this case) of the mesh for integration purposes. In this section, we investigate a less popular alternative focused on the cell-node relation through the derivatives of the discrete integral with respect to node position. The idea that the derivative of the volume is a meaningful representation of boundary area is not new (see [79] for instance), and has already been used to characterize the discrete geometry defined by a mesh (see [63] for an excellent introduction, especially section 5.3).

And indeed, according to the structure theorem (see theorem 3.1), there exists a function ϕ_{ϕ^0} defined on $\partial\Omega$ such that:

$$\frac{\partial}{\partial \mathbf{x}(\phi^0)} \int_\Omega f \, \text{vol}_\Omega = \int_{\partial\Omega} f \phi_{\phi^0} \, \text{vol}_{\partial\Omega} \tag{1.40}$$

Hence, it is possible to write a discrete approximation of the derivative of the integral of a function using only boundary values of the function. The discretization method described in section 4.2.1 using mesh-integrated volume-based SFEM operators of section 3.7.2 requires such a formula. Of course, designing such a formula is only possible with an in-depth understanding of the shape of cells, especially in non-planar situations. Indeed, no assumption is made on the variations of the position of a node and these variations can lead to non-planar situations. Actually, these non-planar situations are the most important because they often end up being the only non-vanishing contributions.

The differentiation of the one point integration formula on a mesh (definition 1.17) gives the following approximation of the derivative of the integral of a function with respect to the movement of a boundary node $\phi^0 \in \partial\mathcal{M}$:

$$\frac{\partial}{\partial \mathbf{x}(\phi^0)} \int_\Omega f \, \text{vol}_\Omega \approx \sum_{\phi^n \in \mathcal{M}} \frac{\partial V(\phi^n)}{\partial \mathbf{x}(\phi^0)} f(\mathbf{x}(\phi^n)) + V(\phi^n) \frac{\partial \mathbf{x}(\phi^n)}{\partial \mathbf{x}(\phi^0)}^T \cdot \nabla f(\mathbf{x}(\phi^n)) \tag{1.41}$$

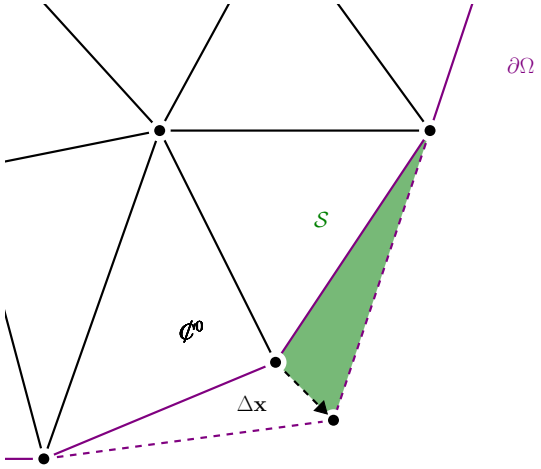


Figure 1.6 – *One boundary node $\mathcal{C}^0$ of a simplex mesh is slightly displaced by the amount $\Delta\mathbf{x}$. The contribution of every boundary simplex $\mathcal{S}$ to the variation of the integral of a field can be approximated to the first order with the computation of the volume and centroid of the added region.*

This formula is not well suited since it requires evaluations of the gradient of f, and of values outside the boundary.

The following proposition gives an alternative that only requires evaluations of f at the boundary:

Proposition 1.18 (Derivative of integral with respect to node movement)**.**
The following expression is a consistent approximation of $\frac{\partial}{\partial \mathbf{x}(\mathcal{C}^0)} \int_\Omega f \,\mathrm{vol}$ of order 1.

$$\frac{1}{n(n+1)} \sum_{q=0}^{d-1} \sum_{\substack{\mathcal{C}^q \in \partial\mathcal{M} \\ \mathcal{C}^0 \in \partial\mathcal{C}^d}} \left(\sum_{t \in \mathcal{C}^q} f(\mathbf{x}(\mathcal{C}^q)) + \sum_{p=0}^{d-1} f(\mathbf{x}(\mathcal{C}^p(t))) \right) \frac{\partial \mathbf{x}(\mathcal{C}^q)}{\partial \mathbf{x}(\mathcal{C}^0)}^T \cdot \mathbf{\Gamma}(t^{d-1}) \tag{1.42}$$

The above expression requires evaluation of the function at every centroid of every boundary cell of every dimension, but does not use interior values or gradient information. Unfortunately, we did not manage to simplify it to bypass the sum over tuples as with previous formulae.

Proof. Let us first consider the simpler case of a simplex mesh. Since the union of all boundary tuples enclose the interior of the mesh, the non-vanishing contributions to the total result can be added for each boundary tuple as represented in figure 1.6. The volume of the green region is $\frac{1}{n}\mathbf{\Gamma}(\mathcal{S}) \cdot \Delta\mathbf{x}$ and its centroid is $\frac{1}{n+1}\left(\mathbf{x}(\mathcal{C}^0)\Delta\mathbf{x} + \sum i \in \mathcal{S}\mathbf{x}_i\right)$, so that the total contribution to the integral reads:

$$\frac{1}{n}\mathbf{\Gamma}(\mathcal{S}) \cdot \Delta\mathbf{x}\ f\left(\frac{1}{n+1}\left(\mathbf{x}(\mathcal{C}^0)\Delta\mathbf{x} + \sum_{i \in \mathcal{S}} \mathbf{x}_i\right)\right) \tag{1.43}$$

Adding up all contributions from boundary simplices and using $f(\sum_i \alpha_i \mathbf{x}_i) = \sum_i \alpha_i f(\mathbf{x}_i)$ if f is affine and $\sum_i \alpha_i = 1$, we get at the first order in $\Delta\mathbf{x}$:

$$\sum_{\mathcal{S}\in\partial\mathcal{M}} \frac{1}{n(n+1)} \mathbf{\Gamma}(\mathcal{S}) \cdot \Delta\mathbf{x} \left(f(\mathbf{x}(\mathcal{C}^0)) + \sum_{i\in\mathcal{S}} f(\mathbf{x}_i) \right) \tag{1.44}$$

The formulae given in proposition 1.13 for the volumes and centroids of cells describes the sub-cell geometry as a union of simplices whose nodes are the centroids of cells of every dimension. This description fully determines the meaning of $\frac{\partial}{\partial \mathbf{x}(\mathcal{C}^0)} \int_\Omega f \,\mathrm{vol}$ and incidentally that of $\phi_{\mathcal{C}^0}$ in expression (1.40). The movement of a single boundary node impacts all centroids of cells linked to this node via the derivatives $\frac{\partial \mathbf{x}(\mathcal{C}^q)}{\partial \mathbf{x}(\mathcal{C}^0)}$, and adding up all contributions gives the desired expression. □

A quick summary

This first chapter is dedicated to detailing the topological and geometrical aspects of meshes. These concepts are fundamental to understand the challenge of meshless methods since we need to know what we leave behind when we decide not to use a mesh. Moreover, the topological structure of meshes is still exploited for integration purposes in the rest of this work.

In section 1.1, we detail two abstract cell complex structures: the simplicial complex and the cell-tuple structure. This study is motivated by a result by Brisson [46]: the topology of subdivided manifolds can be described using cell-tuple structures. We introduce the notions of (local) orientability and orientation, and show that cell-tuple structures are locally orientable. This enables the definition of a boundary operator on cell-tuple structure, laying the foundations for a complete homological study of subdivided manifolds.

In section 1.2, we study the geometrical aspects of meshes. We give a formula and a practical algorithm to compute volumes and centroids of cells of a mesh, under the assumption that every cell is planar, namely that it is contained in an affine hyperspace of the corresponding dimension. This algorithm only requires the *a priori* knowledge of the cell-tuple structure and node positions. Then, we generalize this algorithm to the non-planar case, and give a full geometrical description of the corresponding geometry of the non-planar cells. This original result is the most important of this first chapter. Once again, this description only requires the *a priori* knowledge of the cell-tuple structure and node positions, and is fully consistent with the computation of volumes and centroids. In particular, this description allows a proper definition of the derivative of cell volumes and centroids with respect to node positions, where the planarity assumption is too strong.

Finally, we prove in section 1.3 a few mesh-based formulae for the approximation of integrals. These formulae make use of the previously computed volumes and centroids of cells, and we need them for several of our meshless developments: in sections 2.7.3 and 3.7.2, we use these formulae and take advantage of the structure of the mesh to retain without any approximation some interesting properties of exactly integrated meshless methods.

Chapter 2

A Meshless Nodal Operator Framework

Contents

We have seen in chapter 1 that a mesh defines a powerful computational structure and provides and adequate theoretical foundation for the discretization of PDEs. In this chapter, we build a meshless computational structure for simulation purposes. It is the combination of a point cloud and several linear operators defined on this point cloud. All the concepts related to this structure (the point cloud, the different operators, ...) are called "discrete" as opposed to the "continuous" objects (the computational domain Ω, the integration or differentiation operators, ...). The term "continuous" is used here very loosely and does not refer to the calculus concept of continuity. For instance, fields defined on the computational domain are referred to as "continuous functions", even if they do not satisfy the $\varepsilon - \delta$ definition of continuity, and their discretized version defined on the point cloud are called

"discrete functions". This choice of vocabulary has become quite widespread in the last decade (see [178] for another usage of the term "discrete function" and [72] for the term "discrete differential form").

Most mesh-based methods consider that the geometry of the underlying computational domain Ω is exactly described by the mesh, but meshless methods cannot start with this assumption. In this work, we suppose that Ω is given and do not treat the difficult problem of finding a possible underlying geometry starting from a point cloud. This problem is the subject of a dedicated literature (see [32, 171] and references therein for a review).

The central objects of our meshless framework are the point cloud and discrete differentiation and integration operators acting on discrete fields defined on the point cloud. The link between continuous and discrete fields is provided by the reduction map, which we choose to be the pointwise evaluation at nodes of the point cloud. For this reason, our meshless framework is called "nodal". Our operator-based framework is greatly inspired from that of Chiu [60] and can be considered as its generalization.

The quality of the discretization is characterized with two main criteria. First, we look at consistency, namely we explore the nullspace of the error functional. Consistency is most often characterized by its order, namely the maximum degree of polynomial space on which an operator is exact. Most meshless discretization techniques – especially for the gradient operator – focus on this aspect. The relevance of consistency concerning the convergence rate comes from the Bramble-Hilbert lemma stated earlier (see lemma 1.15).

Secondly, we consider the mimetic properties of the discrete operators. In particular, we pay close attention to the fact that they jointly verify a discrete version of Stokes's formula. This highly desirable feature of the discretization – which we call compatibility – is at the heart of our work. In fact, the difficulty in achieving compatibility has been a recurring theme in the meshless community (see [12, 36] for instance) , and we conjecture that the computational effort necessary to satisfy it exactly is at least that of building a mesh, or solving a global linear system. We call this speculated limitation the *meshless curse*, and we verify it on multiple examples throughout this work. This conjecture is the most important result of this chapter.

In section 2.1, we give a definition of the point cloud: our geometrical equivalent of a mesh. In section 2.2, we define meshless discrete integration and differentiation operators on the point cloud, interpret the resulting structure and compare it to that defined by a mesh. The evaluation of the "performance" of discrete operators compared to their continuous analogs is performed in section 2.3 with the definition of consistency, and in section 2.4, where we propose a way to get more insight into the previously defined operators with the definition of a dual gradient operator. Its consistency properties characterize how well the initial operators work together. In particular, consistency of order zero of the dual gradient is equivalent to compatibility and is thus of the utmost importance.

In sections 2.5, we show how the classical SPH discretizations and renormalized improvements fit in our meshless framework. Along with standard MLS-type discrete operators (developed in appendix D), these methods can achieve first order consistency, albeit their poor compatibility properties motivates the development in section 2.6 of a least-square correction procedure allowing the recovery of compatibility without sacrificing primal first order consistency. This correction procedure requires the solution of a global linear system in accordance with the meshless curse. Additionally, we prove that the corrected DMLS gradient is globally optimal, and our correction method allows a more efficient computation than directly solving the global optimality conditions.

Finally, in section 2.7, we adapt the SFEM operators to our discrete meshless framework. Most importantly, we show how the compatibility properties inherently attached to a mesh can be transferred – via an adequate background integration procedure – to the meshless

structure, illustrating the meshless curse once again.

2.1 The geometrical support for meshless discretization: the point cloud

Mesh-based computational methods use a mesh as a discrete analogue of the domain Ω. Meshless methods on the other hand, avoid using a mesh as much as possible. Instead, the discrete geometrical object allowing the discretization of fields in a meshless framework is the point cloud:

Definition 2.1 (Point cloud)**.**
A point cloud $(\mathcal{C}, \mathbf{X})$ in $\mathbb{R}^d$ is made of:

- A set of n_n nodes (or particles) $\mathcal{C} = [\![1, n_n]\!]$.
- A position operator $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$.

In order to geometrically characterize how good of a discrete substitute for the computational domain a point cloud really is, we need to give a definition of the separation and fill distance (see definition 4.6 and 1.4 of [233]):

Definition 2.2 (Separation, fill distance and quasi-uniformity)**.**
The separation distance of a point cloud $(\mathcal{C}, \mathbf{X})$ is defined as the largest possible radius of two balls centered at nodes of the point cloud with an intersection of vanishing measure:

$$d_{s,\mathcal{C}} \overset{\text{def}}{=} \frac{1}{2} \min_{i \neq j \in \mathcal{C}} \|\mathbf{x}_j - \mathbf{x}_i\| \tag{2.1}$$

In a sense, the separation distance represents the size of the smallest phenomenon uniformly representable with nodal data on a point cloud.

At the other end of the spectrum, the fill distance of a point cloud $(\mathcal{C}, \mathbf{X})$ is the radius of the largest ball with center in Ω that does not contain any node of the point cloud:

$$d_{f,\mathcal{C}} \overset{\text{def}}{=} \sup_{\mathbf{x} \in \Omega} \min_{i \in \mathcal{C}} \|\mathbf{x} - \mathbf{x}_j\| \tag{2.2}$$

The fill distance is the largest distance that nodes of a point cloud needs to "oversee" to insure that there is no hole in the discrete representation of the computational domain.

A sequence of point clouds is said quasi-uniform if the ratio of fill distance to separation distance remains bounded, *i.e.* if there exists a constant C such that:

$$d_{s,\mathcal{C}} \leq d_{f,\mathcal{C}} \leq C d_{s,\mathcal{C}} \tag{2.3}$$

Quasi-uniformity gives a precise sense to the idea that as a point cloud is refined, nodes should be added in the domain without either having nodes too close to each other or creating node-free regions. This requirement is quite similar to its homonym for mesh-based methods (see [41] definition 4.4.13 for instance), which is one of the most usual assumptions for convergence proofs. In the rest of this work, we only use quasi-uniform point clouds.

Example. Any distribution of points qualifies as a point cloud. The most well known node distribution is probably the Cartesian distribution (see figure 2.1a)). In our work, we seldom use the Cartesian distributions because its symmetry properties are likely to endow numerical schemes with far better approximation properties than in the general case, which can be heavily misleading. Obviously, it does not mean that we do not recommend the use

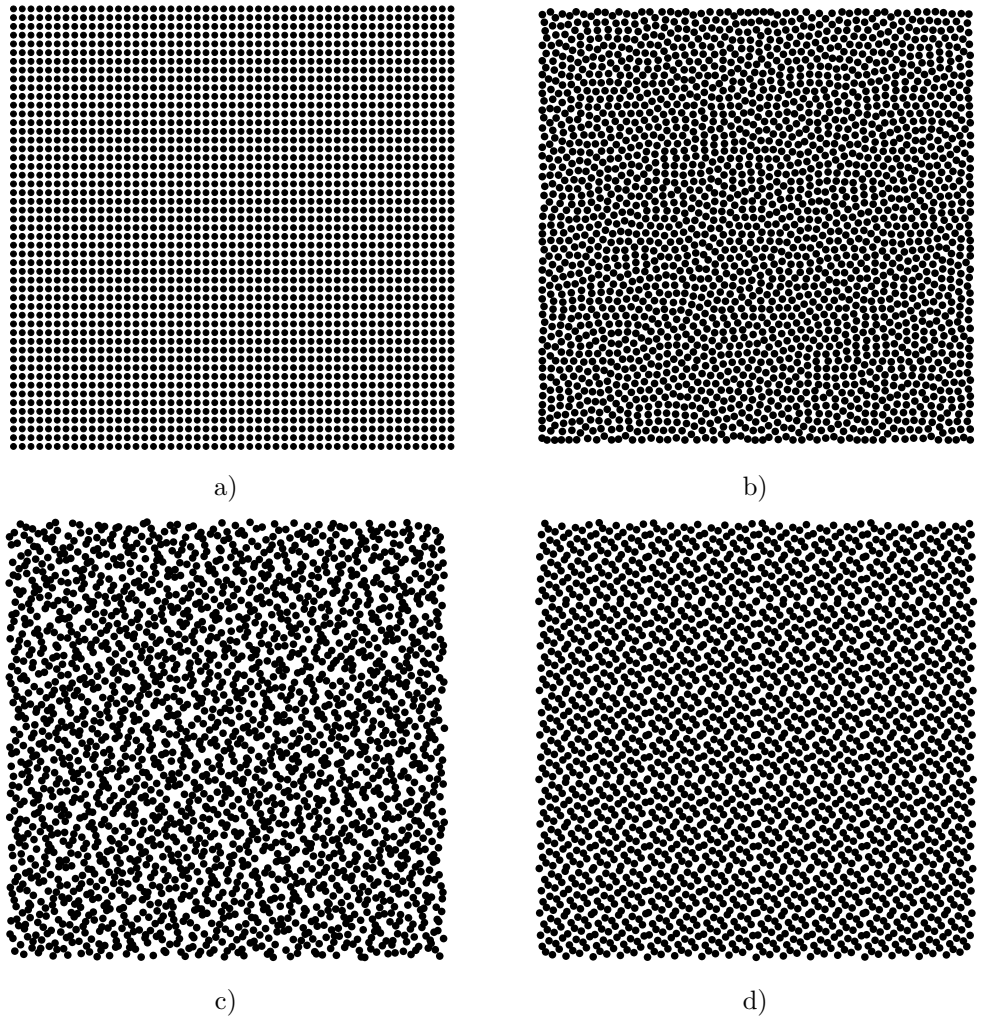


Figure 2.1 – *Simple distributions of points in the square.*
a) Cartesian distribution. *b) Blue noise distribution (see section 3.6 and [92]).*
c) Halton distribution (see [93]). *d) Hammersley distribution (see [93]).*

of Cartesian distribution for industrial simulations, only that convergence studies using such distributions can result in excessive confidence in the precision of the numerical schemes.

Non-regular distribution of nodes include blue noise node distributions (see figure 2.1b)), which are designed to avoid large scale patterns while retaining a constant local density of points. We also use low-discrepancy sequences (see [93] for a definition of discrepancy in a point cloud and example of low-discrepancy sequences) including the Halton distribution (see figure 2.1c)) and the Hammersley distribution (see figure 2.1d)), which are less regular . Note that there is a relationship between discrepancy and the fill distance of a point cloud (see theorem 3 of [64] for a more precise statement). In particular, all families of point cloud cited above are quasi-uniform.

In this work, we restrict ourselves to nodal discretizations. Namely, we consider discrete

fields mapping each node of a point cloud to a value. This stance translates into the following definition:

Definition 2.3 (Discrete fields and reduction map)**.**
A discrete scalar field f on a point cloud $(\mathcal{C}, \mathbf{X})$ is a set of values indexed by the nodes of the point cloud $f : \mathcal{C} \to \mathbb{R}$. The dimension of the space of discrete scalar fields is the number of nodes in the point cloud.

In opposition to discrete fields, we call "continuous field" any function $f : \Omega \to \mathbb{R}$ defined on the computational domain. We do not assume these continuous fields to be continuous in a topological sense. The process of giving a discrete equivalent $f : \mathcal{C} \to \mathbb{R}$ to a continuous function $f : \Omega \to \mathbb{R}$ is called reduction. A reduction map $\mathcal{R} : (\Omega \to \mathbb{R}) \to (\mathcal{C} \to \mathbb{R})$ is a linear map that transforms a continuous field $f : \Omega \to \mathbb{R}$ into a discrete field $\mathcal{R}(f) : \mathcal{C} \to \mathbb{R}$.

Example. A famous reduction map in mesh-based methods is the so-called "De Rham map" defined as the integration of the continuous field on a sub-domain (see [34] for a unified framework for the mimetic discretization of differential operators using the De Rham map as one of its fundamental building blocks). For instance, in finite volume methods, the reduction map is defined as the average value of a function on a cell (see for instance [140] section 4.1, or [190]):

$$\mathcal{R}^{FV}_{\phi'}(f) = \frac{1}{V(\phi')} \int_{\mathbf{x} \in \phi'} f(\mathbf{x}) \, \mathrm{d}V \tag{2.4}$$

In the meshless nodal operator framework, we use the following reduction map:

Definition 2.4 (Pointwise evaluation)**.**
Pointwise evaluation $\mathcal{PE}$ is the reduction map defined as: $\forall \, f : \Omega \to \mathbb{R}$,

$$\mathcal{PE}_i(f) = f(\mathbf{x}_i) \tag{2.5}$$

In this work, we always use point-wise evaluation as our reduction map. For improved readability, and when there is no risk of confusion, we drop the notation $\mathcal{PE}$ and simply denote $f : \mathcal{C} \to \mathbb{R}$ the discrete field associated to $f : \Omega \to \mathbb{R}$. Even though this choice of reduction map is the usual choice for Finite Differences (FD), the methods developed in this work have little in common with FD-type discretization.

In the next section, we enrich the point cloud with several discrete operators, effectively building the necessary tools for the discretization of PDE

2.2 Nodal discrete meshless operators for the discretization of partial differential equations

A recent tendency in the field of numerical methods for the discretization of PDE is to develop operator-based formulations (see [102, 172, 189]). Our work follows this trend and proposes a full operator-based meshless framework, building on the work of Bonet and Kulasegaram [36], Lanson and Vila [135] and others. This framework largely generalizes the one proposed by Chiu [60], allowing for non-symmetric gradient operators, thus separating the conditions for consistency and compatibility.

Even though this framework is quite general already, we do not claim that it is a final solution to any discretization need, and it might need to be adapted to suit specific requirements (an example of such adaptation is given in section 4.3.4 in order to better accommodate the essential enforcement of Dirichlet boundary conditions).

Let us first clarify what we mean by "meshless operator":

Definition 2.5 (Meshless operator)**.**
A discrete operator $\mathbb{O}p$ on a point cloud $(\mathcal{C}, \mathbf{X})$ with continuous analogue Op is a linear map aiming at providing a discrete counterpart of Op. The domain and co-domain of $\mathbb{O}p$ either are those of Op, or close analogs where Ω is substituted for $\mathcal{C}$.

The following sections detail each of the discrete operators used throughout this work.

2.2.1 The volume and boundary integration operators

We now give a definition of the integral of a discrete field. This integration operator is arguably the most important, as it can be used to define all other operators as developed in chapter 3.

Definition 2.6 (Nodal volume integration operator)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud. A *nodal volume integration operator* $\oint\!\!\!\!\!\!\square_{\mathcal{C}} : (\mathcal{C} \to \mathbb{R}) \to \mathbb{R}$ is a linear form on real discrete fields, which aims at providing a discrete approximation to the continuous integration operator over the computational domain $\int_\Omega . \, \mathrm{d}V : (\Omega \to \mathbb{R}) \to \mathbb{R}$.
Being a linear form, the discrete volume integration operator reads in general: $\forall\, f : \mathcal{C} \to \mathbb{R}$,

$$\oint\!\!\!\!\!\!\square_{\mathcal{C}} f \overset{\text{def}}{=} \sum_{i \in \mathcal{C}} V_i f_i \tag{2.6}$$

Hence, defining a discrete volume integration operator is tantamount to defining nodal volume weights $(V_i)_{i \in \mathcal{C}}$.

Remark. The physical dimension of the volume weights is that of a volume: $[V_i] = L^d$ and its International System of Units (SI) unit is m^d. Of course, the volume weights V_i are the meshless analogues of the volume of cells in a mesh. In fact, this choice is a sound alternative as developed in section 2.7.1. In this sense, cells of a mesh are themselves similar to nodes of a point cloud.

Example. The simplest nodal volume weights are arguably the uniform weights:

$$V_i^{\text{Uniform}} \overset{\text{def}}{=} \frac{1}{n_n} \int_\Omega 1 \, \mathrm{d}V \tag{2.7}$$

Even though these nodal weights might not be prime choice in a simulation context, their simplicity and reliability make them a reasonable alternative for the approximation of integrals in very high dimensions. Theoretical efforts aiming at taking advantage of their simplicity has led to the development of low discrepancy sequences (like the Halton and Hammersley sequences for instance, see figure 2.1), which are designed to provide quasi-optimal approximations of the integral of functions with bounded variations. See [93] for more information about low discrepancy sequences.

Remark. If the real volume weights $(V_i)_{i \in \mathcal{C}}$ are positive $V_i > 0$, then the following bilinear form is a scalar product on discrete fields: $\forall\, f, g \in \mathcal{C} \to \mathbb{R}$,

$$(f, g)_{\mathcal{C}} \overset{\text{def}}{=} \oint\!\!\!\!\!\!\square_{\mathcal{C}} f \cdot g = \sum_{i \in \mathcal{C}} V_i f_i \cdot g_i \tag{2.8}$$

The space of fields defined on $\mathcal{C}$ equipped with this scalar product is an analogue of the Lebesgue space of square integrable fields. Following this remark, we denote it $L^2(\mathcal{C}) \overset{\text{def}}{=} \mathcal{C} \to \mathbb{R}$ when it is supposed to come equipped with the above scalar product. Building discrete function spaces in addition to discrete operators is one possible way to formulate discrete equations as further developed in section 4.3.4. Existence of positive integration weights satisfying some consistency requirements is explored in appendix C.

We now define the discrete equivalent of the boundary integral of the product of fields.

Definition 2.7 (Discrete boundary integration operator)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud. A *discrete nodal boundary integration operator* $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}} . \, . : (\mathcal{C} \to \mathbb{R}) \times (\mathcal{C} \to \mathbb{R}^d) \to \mathbb{R}$ is a bilinear form whose aims is to provide a discrete approximation to the continuous integration operator over the boundary of the computational domain $\int_{\partial\Omega} . \, . \cdot \mathbf{n} \, \mathrm{d}S : (\Omega \to \mathbb{R}) \times (\Omega \to \mathbb{R}^d) \to \mathbb{R}$. Being a bilinear form, the discrete boundary integration operator reads in general: $\forall\, f : \mathcal{C} \to \mathbb{R}$, $\forall\, \mathbf{u} : \mathcal{C} \to \mathbb{R}^d$,

$$\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}} f\mathbf{u} \overset{\text{def}}{=} \sum_{i,j\in\mathcal{C}} f_i \boldsymbol{\Gamma}_{i,j} \cdot \mathbf{u}_j \tag{2.9}$$

Defining a discrete boundary integration operator reduces to defining the vector weights $(\boldsymbol{\Gamma}_{i,j})_{i,j\in\mathcal{C}}$.

Remark. We have defined the discrete boundary integration operator as a *bilinear* operator. To our knowledge, this is very untypical, but we will need (see section 3.2 for instance) to deal with cases where the discrete boundary integration of the product of two fields does not reduce to a linear form on the product field. This special case actually corresponds to a diagonal bilinear form, which reads in coordinates:

$$\boldsymbol{\Gamma}^{\text{dia}}_{i,j} \overset{\text{def}}{=} \boldsymbol{\Gamma}_i \delta_{i,j} \tag{2.10}$$

Moreover, we do not *a priori* assume that the boundary integration operator is symmetric: $\boldsymbol{\Gamma}_{i,j} \neq \boldsymbol{\Gamma}_{j,i}$ in general. Great care must thus be taken with the order of the arguments in $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}}$. The discrete operator with coefficients $\boldsymbol{\Gamma}_{j,i}$ is called the transposed of $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}}$.

Definition 2.8 (Boundary of a point cloud)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud equipped with a nodal boundary integration operator $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}}$. We define the *boundary of the point cloud* $\partial\mathcal{C}$ as the set of directed edges yielding a non-vanishing contribution in the computation of discrete boundary integrals:

$$\partial\mathcal{C} \overset{\text{def}}{=} \{(i,j) \in \mathcal{C}^2 \mid \boldsymbol{\Gamma}_{i,j} \neq \mathbf{0}\} \tag{2.11}$$

As such, the boundary of the point cloud $\partial\mathcal{C}$ is a directed graph. For a given node of the point cloud, this provides two possibly different definitions of "being on the boundary":

- A left-sided version: i is a left-sided boundary node if there exists a node $j \in \mathcal{C}$ such that $\boldsymbol{\Gamma}_{i,j} \neq \mathbf{0}$. This is the relevant notion for the weak imposition of Dirichlet boundary conditions as explained in section 4.2.1.
- A right-sided version: i is a right-sided boundary node if there exists a node $j \in \mathcal{C}$ such that $\boldsymbol{\Gamma}_{j,i} \neq \mathbf{0}$. This is the relevant notion for Neumann boundary conditions as explained in section 4.2.1.

Remark (Mixed discrete-continuous boundary integration operator)**.** For the weak imposition of boundary conditions, we will also need the following mixed discrete-continuous boundary integration operator. Given a *continuous* function $f : \Omega \to \mathbb{R}$, we also define boundary integration as a linear forms on discrete fields $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}} f \, . \, : (\mathcal{C} \to \mathbb{R}^d) \to \mathbb{R}$, or $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}} \, . \, f : (\mathcal{C} \to \mathbb{R}^d) \to \mathbb{R}$ whose aims is to provide a discrete approximation to the following continuous form: $\int_{\partial\Omega} f \, . \cdot \mathbf{n} \, \mathrm{d}S : (\Omega \to \mathbb{R}^d) \to \mathbb{R}$. In certain cases, we will need to make sure that the result of the linear discrete boundary operators agrees with that of the bi-linear operator of definition 2.7.

2.2.2 The primal gradient operator

In this section, we define the discrete differentiation operator and support the idea that such a definition provides the point cloud with a sufficient structure to play the part of a mesh for the discretization of PDEs.

Definition 2.9 (Discrete nodal gradient)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud. The discrete nodal gradient operator $\mathbb{\nabla} : (\mathcal{C} \to \mathbb{R}) \to (\mathcal{C} \to \mathbb{R}^d)$ maps discrete scalar fields to discrete vector fields. Its aim is to provide a discrete approximation to the continuous gradient operator over the computational domain $\nabla : (\Omega \to \mathbb{R}) \to (\Omega \to \mathbb{R}^d)$.

Given a volume integration operator, we can in general write the gradient operator as: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$V_i \mathbb{\nabla}_i f \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \mathbf{A}_{i,j} f_j \tag{2.12}$$

The vectors $(\mathbf{A}_{i,j})_{i,j \in \mathcal{C}}$ are called gradient coefficients. Note that a factor V_i is introduced in their definition to simplify subsequent developments. In the rest of this work, we implicitly assume that the volume weights are non-vanishing so that the gradient vector coefficients fully determine the gradient operator.

The only difference between definition 2.9 and definition (2.1) of Chiu in [60] is that we do not *a priori* enforce any symmetry conditions and instead define a dual gradient operator (see section 2.2.2).

Remark. We naturally extend definition 2.9 of the discrete gradient operator in a component-wise fashion to tensors of other shapes. For instance we define the discrete divergence of a discrete vector field $\mathbf{u} : \mathcal{C} \to \mathbb{R}^d$ as:

$$V_i \mathbb{\nabla}_i \cdot \mathbf{u} \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \mathbf{A}_{i,j} \cdot \mathbf{u}_j \tag{2.13}$$

In fact, $\mathbb{\nabla}$ is precisely the discrete exterior derivative operator announced in section 1.1.2: its definition imposes a structure on the point cloud that replaces the mesh. Indeed, in order to keep locality and sparsity properties of operators, coefficients $\mathbf{A}_{i,j}$ typically vanish if the distance $\|\mathbf{x}_j - \mathbf{x}_i\|$ is larger than a fixed threshold. This distance is denoted h and called the smoothing length. The non-vanishing coefficients implicitly define a directed graph on the point cloud, and the set of the neighbors of a node i is denoted $\mathcal{N}(i)$:

$$\mathcal{N}(i) \overset{\text{def}}{=} \{j \in \mathcal{C} \mid \mathbf{A}_{i,j} \neq \mathbf{0}\} \tag{2.14}$$

This graph defines connectivities between nodes and is the meshless substitute for the boundary relation defined on a mesh (see figure 2.2). By extension, this graph is sometimes called "graph of the point cloud", although it would be more correct to speak about the graph of a discrete gradient on the point cloud.

Remark. In a sense, faces of a mesh are similar to edges of the graph of a point cloud: they separate cells (resp. nodes) of a mesh (resp. point cloud) and the gradient coefficient $\mathbf{A}_{i,j}$ can be thought of as vector "face" surface areas between the meshless "cells" i and j. This idea is exploited in section 2.7.1 where dual faces of a mesh are used in the construction of a mesh-based gradient operator. Moreover, the physical dimensions of the gradient edge weights is that of an area: $[\mathbf{A}_{i,j}] = L^{d-1}$ and its SI unit is m^{d-1}. This interpretation justifies our choice of notation ($\mathbf{A}$ for "area").

Remark. The discrete operator $\mathbb{\nabla}$ is called *primal* gradient in opposition to the *dual* gradient defined in section 2.4

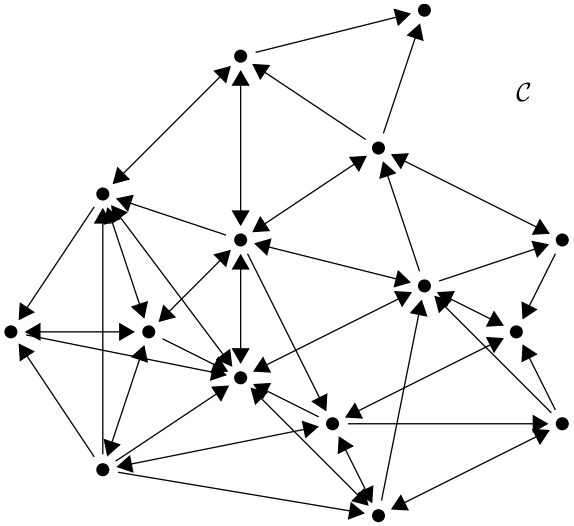


Figure 2.2 – *Non vanishing gradient coefficients* $\mathbf{A}_{i,j} \neq \mathbf{0}$ *define edges of a directed graph on* $\mathcal{C}$*. This gradient graph is the best meshless equivalent to the incidence relation nested bipartite graph defined by a cell-tuple structure (see figure 1.2).*

2.2.3 Meshless shape functions and the reconstruction operator

In order to build Galerkin-type meshless discretization (see section 4.2.2) or as a way to compute discrete integration operators (see definition 2.24 of SFEM operators), we need the definition of reconstruction operator:

Definition 2.10 (Reconstruction operator)**.**
A reconstruction operator (or reconstruction map) $< . >: (\mathcal{C} \to \mathbb{R}) \to (\Omega \to \mathbb{R})$ is a linear map that associates a continuous field to a discrete field. It aims at providing a discrete approximation of the identity map on $\Omega \to \mathbb{R}$.

In general, a reconstruction map can be written as: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ \mathbf{x} \in \Omega$,

$$< f > (\mathbf{x}) = \sum_{i \in \mathcal{C}} \phi_i(\mathbf{x}) f_i \tag{2.15}$$

In accordance with the FEM terminology, the functions $\phi_i : \Omega \to \mathbb{R}$ are called shape functions.

Definition 2.11 (Delta property)**.**
The reconstruction map $< . >$ satisfies the delta property if it is a right-inverse of the reduction map: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$\mathcal{R} \circ < f >= f \tag{2.16}$$

Or again, in terms of shape functions nodal values, $\forall\ i, j \in \mathcal{C}$,

$$\phi_i(\mathbf{x}_j) = \delta_{i,j} \tag{2.17}$$

The apparition of the Kronecker delta symbol in expression (2.17) is the reason for the name "delta-property".

Remark. The reconstruction operation $< . >$ is essentially the converse to the reduction operation $\mathcal{R}$ (definition 2.3). However, the delta-property should not be mistaken for

reproducibility: the delta-property requires that $< . >$ is a right-inverse of $\mathcal{R}$, whereas reproducibility (or exactness, see definition 2.12) means that $< . >$ is a left-inverse of $\mathcal{R}$ on a finite dimensional subspace.

In the next section, we introduce the notions needed to characterize discrete operators.

2.3 Consistency of discrete meshless operators

As we have seen in section 1.3.1 with the Bramble-Hilbert lemma, a first step in the characterization of discrete operators is to ensure that they behave similarly to their continuous counterparts on a finite set of fields:

Definition 2.12 (Exactness of a discrete operator)**.**
A discrete operator $\mathbb{O}p$ with continuous analogue Op is exact on a continuous field $f : \Omega \to \mathbb{R}$ if its action on the reduced field $\mathbb{O}p \circ \mathcal{R}(f)$ matches the discretized value of the target result $Op(f)$.

Remark. The discretization of $Op(f)$ may or may not involve the reduction operator. For instance, in the case of the volume integration, $\int_{\mathbf{x}\in\Omega} f(\mathbf{x})\ \mathrm{d}V$ is readily discretized (it is a scalar value), whereas in the case of the gradient operator, ∇f is a continuous field that needs reduction.

In any case, as illustrated in figure 2.3, the set of continuous fields on which an operator is exact can be written as the kernel of the error operator, namely the commutator $[\text{Discretization}, \text{Operator}]$. In particular, this shows that this set natively comes with a vector space structure.

Remark. Exactness of the reconstruction operator on a function space φ is equivalent to $\forall\ f \in \varphi,\ f = \sum_{i\in\mathcal{C}} f(\mathbf{x}_i)\phi_i$. For this reason, we also say that $< . >$ *reproduces* fields of φ, and exactness of the reconstruction operator is termed *reproducibility*. Moreover, if the delta property (definition 2.11) is satisfied, then the set of reproduced functions is exactly the image of the reconstruction operator $\mathrm{Im}(< . >)$.

Example. For instance, the gradient operator $\mathbb{\nabla} : (\mathcal{C} \to \mathbb{R}) \to (\mathcal{C} \to \mathbb{R})$ is exact on $f : \Omega \to \mathbb{R}$ if the following commutation property is satisfied:

$$\mathbb{\nabla} \circ \mathcal{R}(f) = \mathcal{R} \circ \nabla(f) \tag{2.18}$$

Since we have chosen point-wise evaluation as our reduction map, expression (2.18) can be re-written in terms of gradient coefficients as: $\forall\ i \in \mathcal{C}$,

$$\sum_{j\in\mathcal{C}} \mathbf{A}_{i,j} f(\mathbf{x}_j) = V_i(\nabla f)(\mathbf{x}_i) \tag{2.19}$$

Definition 2.13 (Order of consistency of a discrete operator)**.**
A discrete operator is said "consistent of order $p \in \mathcal{N}$" if it is exact on all polynomials of degree $q \leq p$. For $p = 1$, the operator is also termed "linearly consistent".

Remark. Because of the vector space structure of the kernel of the error operator, it is sufficient to check exactness on a basis of the polynomial space in order to prove consistency. The monomial basis is of course the most used.

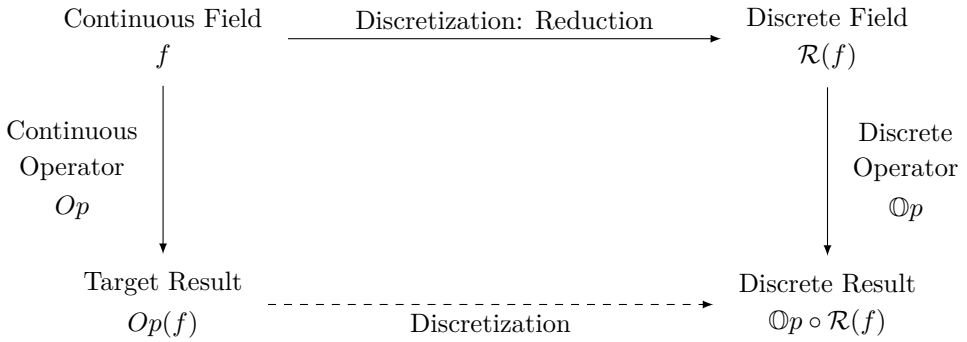


Figure 2.3 – *A discrete operator $\mathbb{O}p$ with continuous analogue Op is exact on a continuous field $f : \Omega \to \mathbb{R}$ if the discretization process sketched on this diagram is commutative.*

Example (Gradient consistency of order zero and one)**.** A gradient operator is consistent of order zero if and only if $\mathbb{\nabla} 1 = \mathbf{0}$. In coordinates, zeroth order consistency reads: $\forall\, i \in \mathcal{C}$,

$$\sum_{j \in \mathcal{C}} \mathbf{A}_{i,j} = \mathbf{0} \tag{2.20}$$

Similarly, it is first order consistent if it additionally satisfies $\mathbb{\nabla}\mathbf{x} = \mathrm{I}_d$. In coordinates, first order consistency reads: $\forall\, i \in \mathcal{C}$,

$$\sum_{j \in \mathcal{C}} \mathbf{A}_{i,j} \mathbf{x}_j^T = V_i \mathrm{I}_d \tag{2.21}$$

Building consistent meshless operators (and especially meshless gradients) with good consistency properties is a well-covered topic in the literature. Several examples of such constructions are given in section D and 2.7. Moreover, chapter 3 builds on the idea that consistency of the volume integration formula can be exploited to build consistent (dual) gradient and reconstruction operators. In the next section, we quickly expose the renormalization, a popular meshless procedure to ensure zeroth or first order consistency of a given gradient operator.

2.3.1 Renormalization of meshless gradient operators

Re-normalization is a meshless discretization technique aiming at increasing the order of consistency of a given gradient operator. It was first proposed by Randles and Libersky [193] for its variant of order zero. Other similar but less famous propositions have then been made like the Johnson-Beissel correction [123] and variants by Krongauz and Belytschko [132]. Its first order variant originated in the work of Moussa et al. (see [135, 169, 227] for instance) as a means to lift the requirement that the ratio of a characteristic length of the discretization to the smoothing length asymptotically vanishes for convergence.

The following sections present the zeroth and first order variants of renormalization in our meshless framework.

Renormalization of order zero

Renormalization of order zero is often presented in the SPH litterature as inspired by the product rule for differentiation (see e.g. [161] expression (2.9)). With our notations, this can be expressed as:

Definition 2.14 (Renormalization of order zero)**.**
The renormalized gradient of order zero $\mathbb{\nabla}^{R0}$ is defined as: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ i \in \mathcal{C}$,

$$\mathbb{\nabla}_i^{R0} f \overset{\text{def}}{=} \mathbb{\nabla}_i f - f_i \mathbb{\nabla}_i 1 \tag{2.22}$$

Or equivalently,

$$V_i \mathbb{\nabla}_i^{R0} f \overset{\text{def}}{=} \sum_{i \in \mathcal{C}} \mathbf{A}_{i,j} (f_j - f_i) \tag{2.23}$$

This is again equivalent to the following definition of the vector gradient coefficients of $\mathbb{\nabla}^{R0}$: $\forall\ i, j \in \mathcal{C}$,

$$\begin{cases} \mathbf{A}_{i,j}^{R0} \overset{\text{def}}{=} \mathbf{A}_{i,j} & \text{If } i \neq j \\ \mathbf{A}_{i,i}^{R0} \overset{\text{def}}{=} -\sum\limits_{\substack{j \in \mathcal{C} \\ j \neq i}} \mathbf{A}_{i,j} \end{cases} \tag{2.24}$$

The interest in such a definition lies in the following property:

Proposition 2.1 (Renormalized gradients of order zero are zeroth order consistent)**.**
$\mathbb{\nabla}^{R0}$ is a zeroth order consistent analogue to the continuous gradient operator ∇.

Proof. It is sufficient to check exactness on a unit constant field: $\forall\ i \in \mathcal{C}$,

$$\begin{aligned} V_i \mathbb{\nabla}_i^{R0} 1 &= \sum_{j \in \mathcal{C}} \mathbf{A}_{i,j}^{R0} \\ &= \sum_{\substack{j \in \mathcal{C} \\ j \neq i}} \mathbf{A}_{i,j} - \sum_{\substack{j \in \mathcal{C} \\ j \neq i}} \mathbf{A}_{i,j} = \mathbf{0} = V_i (\nabla 1)(\mathbf{x}_i) \end{aligned}$$

□

Renormalization of order one

The following definition gives the first order variant of the renormalization technique:

Definition 2.15 (Renormalization of order one)**.**
Suppose that $\mathbb{\nabla}_i^{R0}\mathbf{x}$ is an invertible matrix for all nodes i of the point cloud. Then, we can define the renormalized gradient of order one $\mathbb{\nabla}^{R1}$ as: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ i \in \mathcal{C}$,

$$\mathbb{\nabla}_i^{R1} f \overset{\text{def}}{=} \left(\mathbb{\nabla}_i^{R0}\mathbf{x}\right)^{-1} \cdot \mathbb{\nabla}_i^{R0} f \tag{2.25}$$

This is equivalent to the following definition of the vector gradient coefficients of $\mathbb{\nabla}^{R1}$: $\forall\ i, j \in \mathcal{C}$,

$$\begin{cases} \mathbf{A}_{i,j}^{R1} \overset{\text{def}}{=} V_i \left(\sum\limits_{j \in \mathcal{C}} \mathbf{A}_{i,j} \otimes (\mathbf{x}_j - \mathbf{x}_i) \right)^{-1} \cdot \mathbf{A}_{i,j} & \text{If } i \neq j \\ \mathbf{A}_{i,i}^{R1} \overset{\text{def}}{=} -\sum\limits_{\substack{j \in \mathcal{C} \\ j \neq i}} \mathbf{A}_{i,j}^{R1} \end{cases} \tag{2.26}$$

Proposition 2.2 (Renormalized gradients of order one are first order consistent)**.**
$\mathbb{\nabla}^{R1}$ is a first order consistent analogue to the continuous gradient operator ∇.

Proof. It is sufficient to check exactness on the unit constant field and the linear field $\mathbf{x}$: $\forall\ i \in \mathcal{C}$,

$$\begin{aligned} V_i \mathbb{\nabla}_i^{R1} 1 &= V_i \left(\mathbb{\nabla}_i^{R0}\mathbf{x}\right)^{-1} \underbrace{\mathbb{\nabla}_i^{R0} 1}_{=\mathbf{0}} \\ &= \mathbf{0} = V_i(\nabla 1)(\mathbf{x}_i) \\ V_i \mathbb{\nabla}_i^{R1} \mathbf{x} &= V_i \left(\mathbb{\nabla}_i^{R0}\mathbf{x}\right)^{-1} \mathbb{\nabla}_i^{R0}\mathbf{x} \\ &= V_i \mathrm{I}_d = V_i(\nabla \mathbf{x})(\mathbf{x}_i) \end{aligned}$$

□

The cost of renormalization is quite small as it only requires the solution of local $(d \times d)$ linear problems, which explains its frequent use to correct non consistent operators particularly in the SPH community.

In the next section, we approach the notion of compatibility, which can be considered as a meshless equivalent of the fact that interior faces cancel each other in the computation of the boundary of a formal sum of cells. It is the most important characteristic when it comes to the discretization of PDEs, as demonstrated in chapter 4.

2.4 Compatibility of meshless operators and the dual gradient

In section 2.3, we introduced the notion of consistency, which characterizes the approximation power of a single operator. In this section, we combine the integration and differentiation operators in a discrete version of the integration by parts formula to define another differentiation operator. Consistency of order one of this operator is found equivalent to a discrete version of Stokes's theorem. Verifying this property exactly is a notoriously difficult task in meshless method.

The definition of the dual gradient stems from the integration by parts formula recalled below:

Definition 2.16 (Integration by parts formula)**.**
In a more general differential geometry framework, the codifferential operators d^* (sometimes also denoted δ) are defined as the only operators mapping p-forms to $p-1$-forms ($p \in [\![1, d]\!]$) satisfying: $\forall\ \alpha^{p-1}, \in \Lambda^{p-1}\Omega$, $\forall\ \beta^p, \in \Lambda^p\Omega$,

$$\int_\Omega \mathrm{d}\alpha^{p-1} \wedge *\beta^p = \int_\Omega \alpha^{p-1} \wedge *\mathrm{d}^*\beta^p + \int_{\partial\Omega} \alpha^{p-1} \wedge *\beta^p \tag{2.27}$$

On a manifold without boundary, the above expression reads $\left(\mathrm{d}\alpha^{p-1}, \beta^p\right)_{L^2} = \left(\alpha^{p-1}, \mathrm{d}^*\beta^p\right)_{L^2}$, so that d^* is the pre-Hilbert adjoint of d. See [100] for more information about the role of the codifferential operator in Hodge theory.

In our restricted affine setting, the difference between the exterior derivative d and the codifferential d^* is not made: their expressions in orthogonal bases are the same. In the discrete world, the duality relationship (2.27) might not define a discrete codifferential operator identical to the initial differential. Nevertheless, we translate this idea into the following definition:

Definition 2.17 (Discrete dual gradient operator)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud equipped with discrete nodal volume $\fint_{\mathcal{C}}$ and boundary $\fint_{\partial\mathcal{C}}$ integration operators and a discrete nodal gradient operator $\mathbb{\nabla}$. In general, integration by parts formulae are not satisfied at the discrete level. Still, by analogy with the continuous case (p=3 in expression (2.27)), we can define the following *dual gradient operator* $\mathbb{\nabla}^*$: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ \mathbf{u} : \mathcal{C} \to \mathbb{R}^d$

$$\fint_{\mathcal{C}} f\mathbb{\nabla} \cdot \mathbf{u} + \mathbf{u} \cdot \mathbb{\nabla}^* f \overset{\text{def}}{=} \fint_{\partial\mathcal{C}} f\mathbf{u} \tag{2.28}$$

The vector coefficients of $\mathbb{\nabla}^*$ are denoted: $\forall\ i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}^*_i f \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \mathbf{A}^*_{i,j} f_j \tag{2.29}$$

Expression (2.28) can be re-written in terms of the original operator coefficients as: $\forall\ i, j \in \mathcal{C}$,

$$\mathbf{A}^*_{i,j} \overset{\text{def}}{=} -\mathbf{A}_{j,i} + \mathbf{\Gamma}_{j,i} \tag{2.30}$$

Just as d^* is the adjoint of d, the (opposite of the) dual gradient $\mathbb{\nabla}^*$ is the adjoint of $\mathbb{\nabla}$ under the duality pairing $(f, g)_{L^2} = \fint_{\mathcal{C}} fg$. More exactly, $\mathbb{\nabla}$ and $\mathbb{\nabla}^*$ are adjoints of each other with respect to $\fint_{\mathcal{C}}$ and $\fint_{\partial\mathcal{C}}$. A gradient operator $\mathbb{\nabla}$ is called symmetric if it is equal to its own dual: $\mathbb{\nabla} = \mathbb{\nabla}^*$.

This new operator takes its name from the duality principle from which it emerges. In opposition, the initial differentiation operator is called the "primal" operator. Consistency of the dual gradient operator characterizes how well the integration and differentiation operators work together. The dual gradient has previously appeared in the literature under different names. For instance, Moussa et al. call it the "adjoint operator" (see expression (6) of [169] and expression (23) of [135]), Samarskii et al. call it the "determined operator" (see paragraph 1.4 of [199]) and Lipnikov et al. call it the "derived operator" (see section 1.1 of [144]). Other authors like Chiu constrain the definition of the gradient operator to ensure symmetry, and circumvent the notion of a dual gradient altogether.

Remark. The simplicity of expression (2.30) is one of the reasons why we include the factor V_i in the definition of gradient operators. Without it, the relationship between the coordinates of the primal and dual gradient would read: $\forall\ i, j \in \mathcal{C}$,

$$V_i \mathbf{A}_{i,j} + V_j \mathbf{A}^*_{j,i} = \mathbf{\Gamma}_{i,j} \tag{2.31}$$

With those definitions, the coupled approach of Chiu (see section 4.2 of [60]) would not define a linear problem without the appropriate change of variable. This choice is the most widespread in the literature (also see table I of [78]).

Let us now give a definition of the most important property of meshless operator: compatibility.

Definition 2.18 (Compatibility)**.**
A set of discrete operators $\left(\fint_{\mathcal{C}}, \fint_{\partial\mathcal{C}}, \mathbb{\nabla}\right)$ on a point cloud $(\mathcal{C}, \mathbf{X})$ is termed compatible if the following discrete version of Stokes's theorem is verified: $\forall\ \mathbf{u} : \mathcal{C} \to \mathbb{R}^d$,

$$\fint_{\mathcal{C}} \mathbb{\nabla} \cdot \mathbf{u} = \fint_{\partial\mathcal{C}} 1\ \mathbf{u} \tag{2.32}$$

In term of coordinates, expression (2.32) reads: $\forall\ j \in \mathcal{C}$,

$$\sum_{i\in\mathcal{C}} \mathbf{A}_{i,j} = \sum_{i\in\mathcal{C}} \mathbf{\Gamma}_{i,j} \tag{2.33}$$

If we substitute f with 1 in expression (2.28), we see that compatibility is equivalent to zeroth order consistency of the dual gradient: $\forall\ i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}_i^* 1 = \mathbf{0} \tag{2.34}$$

In a sense, compatibility (and *a fortiori* higher order consistency of the dual gradient) characterizes how well the three initial operators $\oint_{\mathcal{C}}$, $\oint_{\partial\mathcal{C}}$ and $\mathbb{\nabla}$ work together. This notion is complementary to that of consistency, which characterizes how well each operator performs.

Remark. Compatibility can also be considered as a meshless equivalent of the fact that face vector areas bound the region defined by a cell, without gap or superimposition. Indeed, let us consider the discrete Gauss gradient of definition 1.17, which maps a discrete field $\mathbf{u}$ defined at faces ϕ^{d-1} to a field defined at cells ϕ^d of a mesh $\mathcal{M}$:

$$V(\phi^d)\mathbb{\nabla}^{\text{Gauss}}_{\phi^d} \cdot \mathbf{u} = \sum_{\phi^{d-1}\in\partial\phi^d} o(\phi^{d-1},\phi^d)\mathbf{\Gamma}(\phi^{d-1}) \cdot \mathbf{u}_{\phi^{d-1}} \tag{2.35}$$

Let us prove that this Gauss gradient satisfies a discrete version of Stokes's theorem: $\forall\ \mathbf{u} : \mathcal{M}^{d-1} \to \mathbb{R}^d$,

$$\begin{aligned}
\oint_{\mathcal{M}}^{\text{Gauss 1pt}} \mathbb{\nabla}\cdot\mathbf{u} &= \sum_{\phi^d\in\mathcal{M}} \sum_{\phi^{d-1}\in\partial\phi^d} o(\phi^{d-1},\phi^d)\mathbf{\Gamma}(\phi^{d-1}) \cdot \mathbf{u}_{\phi^{d-1}} \\
&= \sum_{\phi^{d-1}\in\mathcal{M}^{d-1}} \mathbf{\Gamma}(\phi^{d-1}) \cdot \mathbf{u}_{\phi^{d-1}} \underbrace{\sum_{\phi^d \mid \phi^{d-1}\in\partial\phi^{d-1}} o(\phi^{d-1},\phi^d)}_{=1 \text{ if } \phi^{d-1}\in\partial\mathcal{M},0 \text{ else}} \\
&= \sum_{\phi^{d-1}\in\partial\mathcal{M}} \mathbf{\Gamma}(\phi^{d-1}) \cdot \mathbf{u}_{\phi^{d-1}} \\
&= \oint_{\partial\mathcal{M}} \mathbf{u}
\end{aligned}$$

The crux of the proof lies in the topological properties of the mesh: the added relative orientations of interior faces cancel out, and only boundary faces remain. Compatibility can thus be seen as this weakened topological requirement: the gradient operator can be used to write fluxes of conserved quantities traveling from one meshless node to another as if they were going through a common boundary between these nodes. During this process, there is no accumulation of the conserved quantity, rather an exchange between interior cells. Only boundary nodes can exchange with the exterior of the domain. For this reason, compatibility is one of the sufficient conditions to achieve first order consistency of the discrete symmetric weak diffusion equations. This is further investigated in chapter 4 where we numerically demonstrate in several discretization settings that compatibility allows the recovery of second-order convergence for the simulation of diffusion systems. Similarly, compatibility plays a role in the conservation properties of meshless discretizations of hyperbolic equations as explained in appendix G.

Contrary to what expression (2.34) might suggest, compatibility does not only concern the gradient operator. In fact, only assuming that $\mathbb{\nabla}$ is first order consistent, the following

proposition gives necessary conditions on volume and boundary integration to be part of a compatible set of operators:

Proposition 2.3 (Necessary integral condition for compatibility)**.**
Linear consistency of $\mathbb{\nabla}$ entails the following necessary relation for compatibility: $\forall\ \mathrm{A} \in \mathrm{M}_d(\mathbb{R})$, $\forall\ \mathbf{b} \in \mathbb{R}^d$,

$$\fint_{\mathcal{C}} \mathrm{Tr}(\mathrm{A}) = \fint_{\partial\mathcal{C}} 1(\mathrm{A}\mathbf{x} + \mathbf{b}) \tag{2.36}$$

This set of necessary conditions constitute weak consistency requirements. In a nutshell, it expresses the fact that the integration operators that are part of a compatible set of operators should have the same consistency order as the differentiation operator.

Proof. Simply substitute f for a constant field and $\mathbf{u}$ for a linear field in expression (2.28) of definition 2.17. Of course, assuming a higher order of consistency of the primal gradient would give additional necessary conditions for compatibility. □

Building discrete sets of operator that are both first order consistent and compatible is widely recognized as one of the most important challenge of meshless methods. Contrary to the case of meshes, which natively comes equipped with a compatible discrete structure (as we have seen with the Gauss operators of section 2.7.1 for instance), building such a structure in a meshless framework is always cumbersome. In fact, the difficulty of meeting compatibility requirements exactly can be considered as a specificity of meshless methods.

Several propositions have been made to recover compatibility. Chiu tackles the problem with a definition of discrete operators as the solution of a global least-squares problem, imposing both compatibility and consistency at the same time (see [57, 58, 59, 60] and section 2.6.1). In a SPH framework, Bonet and Kulasegaram developed a similar global nodal correction system to recover compatibility given a first order consistent primal gradient operator (see [36, 38] for instance). In Babuška et al. [12, 14], compatibility conditions are called the "row sum conditions" and a modification of the final second order operator is proposed to recover optimal convergence. The compatibility conditions were also rephrased in a non-symmetric Petrov-Galerkin framework in [55, 132], and several remedies to restore convergence are proposed: modification of the integration formulae, modification of the trial functions, ...

To the author's knowledge, the difficulty of building compatible meshless operators is such that there is no numerical method that violates the following conjecture:

Conjecture 2.4 (Meshless curse)**.**
The computational effort necessary to build a first order consistent and compatible set of discrete operators $\left(\fint_{\mathcal{C}}, \fint_{\partial\mathcal{C}}, \mathbb{\nabla}\right)$ on a point cloud $(\mathcal{C}, \mathbf{X})$ is at least that of either:

- Actually building a mesh.
- Solving a *global* system of linear equations (*i.e.* scaling with the number of nodes n_n in the cloud).

We verify the validity of the meshless curse throughout this work: in section 2.6, we study compatible meshless gradients, which are defined as the solution of a global quadratic optimization problem. In section 2.7.2 and 3.7.1, we develop two compatible sets of meshless operators expressed using integrals that cannot be exactly computed in practice. We develop in section 2.7.3 and 3.7.2 discrete integration procedures using a mesh that retain the first order consistency and compatibility of the exactly integrated operators.

Nevertheless, in chapter 4, we numerically demonstrate that exact compatibility is not really needed. Instead, it is sufficient to make sure that the compatibility error remains

bounded as the discretization is refined to ensure optimal convergence. In the author's opinion, this is the most promising approach to circumvent the meshless curse and design efficient meshless simulation methods.

2.5 Smooth Particle Hydrodynamics

Smooth Particle Hydrodynamics (SPH) is probably the first meshless method for the simulation of PDEs. It may be traced back to 1977 and attributed to the independent work of Lucy [154] and Gingold and Monaghan ([105]) and was initially developed to tackle problems in astrophysics. The founding step of SPH is the "SPH approximation", a mass-weighted version of the Kernel Density Estimation (KDE). KDE is one of the earliest (see [177]) meshless approximation method. It is a non-parametric statistical tool that allows the estimation of the unknown underlying probability density function of a random variable given nodal data.

Several versions of the SPH operators exist. In this section, we recall the operators from [162], and analyze them with regards to consistency and compatibility. Another version of the SPH operators is discussed in section 3.5.

Definition 2.19 (SPH operators)**.**
Let $(W_h)_{h\in\mathbb{R}^+}$ be a "kernel" function *i.e.* a family of positive, radially decreasing functions with support included in $B^d(\mathbf{0},h)$ satisfying $\int_{\mathbb{R}^d} W_h(\mathbf{x})\ \mathrm{d}V = 1$. Then, for all integrable function $f:\mathbb{R}^d\to\mathbb{R}$, we have:

$$\lim_{h\longrightarrow 0}\int_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{y})W_h(\mathbf{x}-\mathbf{y})\ \mathrm{d}V = f(\mathbf{x}) \tag{2.37}$$

Let $(\mathcal{C},\mathbf{X})$ be a point cloud. If we consider $\mathbf{X} = (\mathbf{x}_1,\ldots,\mathbf{x}_{n_n})$ as an independent and identically distributed sample drawn from a distribution with unknown density f, then, expression (2.37) suggests the following estimation of the underlying density:

$$f(\mathbf{x}) \approx \frac{1}{n_n}\sum_{i\in\mathcal{C}} W_h(\mathbf{x}-\mathbf{x}_i) \tag{2.38}$$

This method is called Kernel Density Estimation (KDE) and was first proposed by Rosenblatt in [198]. Let us now suppose that each node $i\in\mathcal{C}$ of the point cloud physically represents a "chunk" of matter of given mass m_i, and that $\mathbf{x}_i$ is a point picked at random inside this chunk. Then, applying the KDE idea, we can approximate the underlying mass density ρ as:

$$\rho(\mathbf{x}) \approx \sum_{i\in\mathcal{C}} m_i W_h(\mathbf{x}-\mathbf{x}_i) \tag{2.39}$$

Evaluation of expression (2.39) at node positions defines nodal mass densities ρ_i for all nodes i of the point cloud given the mass weights m_i:

$$\rho_i \overset{\text{def}}{=} \sum_{j\in\mathcal{C}} m_j W_h(\mathbf{x}_i-\mathbf{x}_j) \tag{2.40}$$

Using expression (2.37) again, we define the so-called *SPH approximation* of a discrete function $f:\mathcal{C}\to\mathbb{R}$ as:

$$< f >^{SPH}(\mathbf{x}) \overset{\text{def}}{=} \sum_{i\in\mathcal{C}} \frac{m_i}{\rho_i} W_h(\mathbf{x}-\mathbf{x}_i) f_i \tag{2.41}$$

The corresponding SPH shape functions read: $\forall\, i \in \mathcal{C}$, $\forall\, \mathbf{x} \in \mathbb{R}$,

$$\phi_i^{SPH}(\mathbf{x}) = \frac{m_i}{\rho_i} W_h(\mathbf{x} - \mathbf{x}_i) \tag{2.42}$$

Since we assumed $\int_{\mathbb{R}^d} W_h(\mathbf{x})\, \mathrm{d}V = 1$, integration of the SPH reconstruction defines a discrete integration operator with the following volume weights: $\forall\, i \in \mathcal{C}$,

$$V_i^{SPH} \overset{\text{def}}{=} \frac{m_i}{\rho_i} = \frac{m_i}{\sum_{j\in\mathcal{C}} m_j W_h(\mathbf{x}_j - \mathbf{x}_i)} \tag{2.43}$$

Similarly, differentiation of the SPH reconstruction yields the classical SPH gradient: $\forall\, i \in \mathcal{C}$, $\forall\, f : \mathcal{C} \to \mathbb{R}$,

$$\begin{aligned} \mathbb{\nabla}_i^{SPH} f &\overset{\text{def}}{=} \nabla < f >^{SPH} (\mathbf{x}_i) \\ &= -\sum_{j\in\mathcal{C}} V_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) f_j \end{aligned} \tag{2.44}$$

The corresponding gradient coefficients read: $\forall\, i, j \in \mathcal{C}$,

$$\mathbf{A}_{i,j}^{SPH} = -V_i V_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \tag{2.45}$$

Because of its genesis in astrophysical simulation – a field of research mainly concerned with unbounded domains – there is no definition of boundary integration operator in SPH. Hence, since the gradient coefficients are skew-symmetric (*i.e.* $\mathbf{A}_{i,j}^{SPH} = -\mathbf{A}_{j,i}^{SPH}$), the classical SPH gradient operator is symmetric: $\mathbb{\nabla}^{SPH} = \mathbb{\nabla}^{SPH\ *}$.

Remark. As a consequence of the normalization condition $\int_\Omega W_h(\mathbf{x})\, \mathrm{d}V = 1$, the kernel function has the physical dimension of the inverse of a volume: $[W_h] = L^{-d}$ and its SI unit is m^{-d}. Given a decreasing function $W : \mathbb{R}^+ \to \mathbb{R}^+$ with support in $[0, 1]$, we can define a radial kernel function as: $\forall\, \mathbf{x} \in \mathbb{R}^d$,

$$W_h(\mathbf{x}) \overset{\text{def}}{=} \frac{1}{c_d(W)} W\left(\frac{\|\mathbf{x}\|}{h}\right) \tag{2.46}$$

A change to spherical coordinates in the integral shows that the value of the dimension dependent renormalization constant is $c_d(W) = h^d S_{d-1} m_{d-1}(W)$, where $m_d(W)$ denotes the d-th moment of W: $m_d(W) \overset{\text{def}}{=} \int_0^1 W(x) x^d\, \mathrm{d}x$ and $S_d = \frac{2\pi^{\frac{n+1}{2}}}{\Gamma\left(\frac{n+1}{2}\right)}$ is the surface area of the unit d-sphere.

The most commonly used kernel functions are the family of B-spline functions introduced by Schoenberg in [203]. These functions have piece-wise polynomial expressions and can be generated using the following Fourier transform (see section 2.3 of [187]):

$$W_n^{\text{B-spline}}(x) \overset{\text{def}}{=} \int_{t=-\infty}^{+\infty} \left(\frac{\sin(\frac{nt}{2})}{t}\right)^n \cos(tx)\, \mathrm{d}t \tag{2.47}$$

These weight functions can also be used for least square reconstruction, although the constraint $\int_\Omega W_h(\mathbf{x})\, \mathrm{d}V = 1$ is not needed there.

Remark (Consistency of SPH operators and renormalized SPH gradient)**.** Even though there exists a few convergence results concerning the SPH operators of definition 2.19 (see [31, 134] for instance), none of them are as little as zeroth-order consistent. Indeed the expression of the SPH gradient of a unit field is:

$$\mathbb{\nabla}_i^{SPH} 1 = -\sum_{j\in\mathcal{C}} V_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \tag{2.48}$$

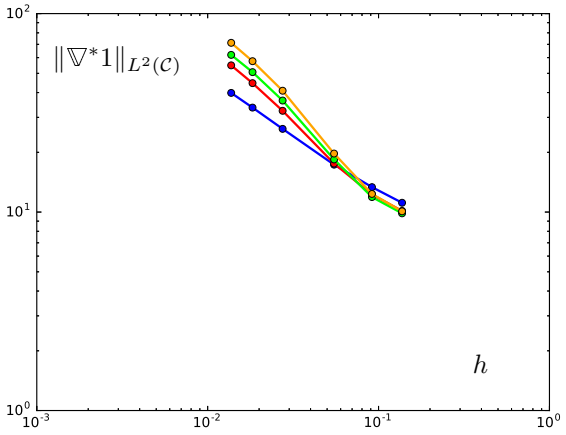


Figure 2.4 – L^2 *norm of the error on the dual of a unit field on Halton distributed point clouds with the standard SPH gradient, the renormalized SPH gradient of order zero, the renormalized SPH gradient of order one, and the linear DMLS gradient.*

This expression does not vanish in general. For this reason, renormalized versions (see section 2.3.1) of the SPH gradients are most often used. For instance, the renormalized SPH gradient of order zero reads: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}_i^{SPH\ R0} f = -\sum_{j \in \mathcal{C}} V_i V_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(f_j - f_i) \tag{2.49}$$

Renormalization breaks the symmetry of the SPH gradient: we do not have $\mathbb{\nabla}^{SPH\ R0} = \mathbb{\nabla}^{SPH\ R0\ *}$. Instead, we can derive:

$$V_i \mathbb{\nabla}_i^{SPH\ R0\ *} f = -\sum_{j \in \mathcal{C}} V_i V_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(f_j + f_i) \tag{2.50}$$

Even though renormalization allows the recovery of consistency of the primal SPH gradient, it does not have any positive impact on that of the dual gradient. On figure 2.4, we give the compatibility error of SPH and MLS gradients. Far from vanishing, this error actually increases with cloud refinement. The rate at which it increases itself increases with the consistency order: a least-square fit shows that the compatibility error grows as $h^{-0.56}$ for the standard SPH gradient, which is not even zeroth-order consistent. The compatibility error of first order consistent operator is almost inversely proportional to the smoothing length: we measure a $h^{-0.84}$ rate of increase for the SPH renormalized operator and a $h^{-0.89}$ rate for the DMLS gradient (defined in appendix D). We confirm in section 4.3.2 that this growth is responsible for the poor convergence behavior of weak discretizations of elliptic PDEs.

With their poor dual consistency properties, the SPH operators are not satisfying as they are. In section 3.6, we interpret several stabilization techniques aiming at getting rid of the infamous "tensile instability" as actually decreasing the compatibility error, confirming that the lack of compatibility is the real source of the inadequacy of the SPH operators.

This is why in the next section we define a correction procedure to retrieve compatibility of a set of meshless operators achieving first order consistency. It is formulated as a global least squares problem, and thus requires the solution of a linear system whose size scales with the total number of nodes, in complete agreement with the meshless curse.

2.6 Compatible meshless operators from quadratic optimization

In this section, we investigate meshless gradient operators defined as the solution of globally defined least square problems. Contrary to the Moving Least Squares (MLS) gradients detailed in appendix D, we not only seek first order consistency, but also compatibility. As a consequence of the combination of the two sets of requirements, all local problems are coupled and the size of the resulting system inevitably scales with the number of nodes in the point cloud. We first recall in section 2.6.1 one of Chiu's approach for the generation of first order consistent compatible meshless gradient. In section 2.6.2, we introduce a related least-norm correction for compatibility: given a first order consistent gradient, a method is proposed to only solve the compatibility equations, while retaining first order consistency. The main result of this section is given in proposition 2.5, where the correction procedure applied to the Diffuse Moving Least Squares (DMLS) gradient with singular weights is proved to be equivalent to the segregated approach of Chiu, where the size of the system to solve is drastically reduced.

2.6.1 The segregated approach of Chiu

In [60], Chiu develops two approaches to generate first order consistent compatible discrete operators. First a "coupled approach" where given $\mathbb{f}_{\partial\mathcal{C}}$ and an *a priori* graph, he seeks both $\mathbb{f}_{\mathcal{C}}$ and $\mathbb{V}$ as the solution of a quadratic optimization problem. Then, a "segregated" variant where the integration operators are supposed known and the only unknown is the gradient operator. We slightly generalize this segregated approach by allowing non symmetric gradients and introducing weights in the optimization problem instead of only a graph (which corresponds to weights in $\{0,1\}$):

Definition 2.20 (Chiu's segregated gradient)**.**
The weighted segregated approach of Chiu reads: Find $\mathbf{A}^{\text{Chiu}} : \mathcal{C}\times\mathcal{C}\to\mathbb{R}^d$ such that the following cost function is minimal:

$$\mathbf{A} \mapsto \sum_{i,j\in\mathcal{C}} \frac{1}{W_h(\mathbf{x}_j-\mathbf{x}_i)}\mathbf{A}_{i,j}^2 \tag{2.51}$$

Under the constraint that the corresponding gradient is linear consistent and that $\left(\mathbb{f}_{\mathcal{C}}, \mathbb{f}_{\partial\mathcal{C}}, \widetilde{\mathbb{V}}^{\text{Chiu}}\right)$ form a compatible set of operators: $\forall\, i\in\mathcal{C}$,

$$\begin{cases} V_i\mathbb{V}_i^{\text{Chiu}}1 = \sum_{j\in\mathcal{C}} \mathbf{A}_{i,j}^{\text{Chiu}} & = \mathbf{0} \\ V_i\mathbb{V}_i^{\text{Chiu}}\mathbf{x} = \sum_{j\in\mathcal{C}} \mathbf{A}_{i,j}^{\text{Chiu}}\mathbf{x}_j^T & = V_i\mathrm{I}_d \\ V_i\mathbb{V}_i^{\text{Chiu}\ *}1 = \sum_{j\in\mathcal{C}} \mathbf{\Gamma}_{j,i} - \mathbf{A}_{j,i}^{\text{Chiu}} & = \mathbf{0} \end{cases} \tag{2.52}$$

The compatibility conditions in the above system define a set of globally coupled linear equations, contrary to the moving least square gradients of appendix D.

In the next section, we adapt this approach to correct a given set of meshless operators for compatibility. In the special case of the DMLS gradient operator, this correction procedure is re-interpreted as a splitting of system (2.52) into many small local systems and a

smaller global system, effectively lowering the computational burden associated with Chiu's segregated approach.

2.6.2 Least-norm correction for compatibility

In this section, we suppose given a point cloud $(\mathcal{C}, \mathbf{X})$ equipped with discrete volume $\fint_{\mathcal{C}}$ and boundary $\fint_{\partial\mathcal{C}}$ integration operators verifying the necessary conditions for compatibility of proposition 2.3 and a *linear consistent* primal gradient $\mathbb{\nabla}$. Least-norm correction for compatibility is a technique which allows us to build a *corrected* gradient $\widetilde{\mathbb{\nabla}}$ such that $\left(\fint_{\mathcal{C}}, \fint_{\partial\mathcal{C}}, \widetilde{\mathbb{\nabla}}\right)$ form a compatible set of operators.

Definition 2.21 (Linear consistent corrections)**.**
Suppose that $\mathbb{\nabla} : (\mathcal{C} \to \mathbb{R}) \to (\mathcal{C} \to \mathbb{R}^d)$ is a linear consistent gradient operator. Then, any linear consistent gradient operator $\widetilde{\mathbb{\nabla}}$ reads: $\forall\ i \in \mathcal{C}$, $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$V_i \widetilde{\mathbb{\nabla}}_i f = V_i \mathbb{\nabla}_i f + \sum_{j \in \mathcal{C}} \lambda_{i,j} (f_j - f_i - (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbb{\nabla}_i f) \tag{2.53}$$

The above expression provides a way to define linear consistent discrete gradients from a particular linear consistent discrete gradient and vector edge coefficients. Given $\mathbb{\nabla}$, the operator $\widetilde{\mathbb{\nabla}}$ is called the *corrected gradient* and the vector coefficients $(\boldsymbol{\lambda}_{i,j})_{i,j\in\mathcal{C}}$ are called *correction coefficients*.

Proof. Let us first prove that expression (2.53) defines a linear consistent gradient for any set of correction coefficients $(\boldsymbol{\lambda}_{i,j})_{i,j\in\mathcal{C}}$: $\forall\ i \in \mathcal{C}$,

$$\begin{aligned}
V_i \widetilde{\mathbb{\nabla}}_i 1 &= V_i \underbrace{\mathbb{\nabla}_i 1}_{=0} + \sum_{j \in \mathcal{C}} \boldsymbol{\lambda}_{i,j} (\underbrace{1 - 1}_{=0} - (\mathbf{x}_j - \mathbf{x}_i) \cdot \underbrace{\mathbb{\nabla}_i 1}_{=0}) \\
&= 0 \\
V_i \widetilde{\mathbb{\nabla}}_i \mathbf{x} &= V_i \underbrace{\mathbb{\nabla}_i \mathbf{x}}_{=\mathrm{I}_d} + \sum_{j \in \mathcal{C}} \boldsymbol{\lambda}_{i,j} (\mathbf{x}_j - \mathbf{x}_i - (\mathbf{x}_j - \mathbf{x}_i) \cdot \underbrace{\mathbb{\nabla}_i \mathbf{x}}_{=\mathrm{I}_d}) \\
&= V_i \mathrm{I}_d + \sum_{j \in \mathcal{C}} \boldsymbol{\lambda}_{i,j} \underbrace{((\mathbf{x}_j - \mathbf{x}_i) - (\mathbf{x}_j - \mathbf{x}_i))}_{=\mathbf{0}} \\
&= V_i \mathrm{I}_d
\end{aligned}$$

Hence, $\widetilde{\mathbb{\nabla}}$ is linear consistent. Reciprocally, if $\widetilde{\mathbb{\nabla}}$ is first order consistent, then $\boldsymbol{\lambda}_{i,j} = \widetilde{\mathbf{A}}_{i,j} - \mathbf{A}_{i,j}$ satisfies expression (2.53). Indeed,: $\forall\ i \in \mathcal{C}$,

$$\begin{aligned}
&\mathbb{\nabla}_i f + \sum_{j \in \mathcal{C}} (\widetilde{\mathbf{A}}_{i,j} - \mathbf{A}_{i,j})(f_j - f_i - (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbb{\nabla}_i f) \\
&\quad = \mathbb{\nabla}_i f + \widetilde{\mathbb{\nabla}}_i f - \mathbb{\nabla}_i f + f_i \underbrace{(\widetilde{\mathbb{\nabla}}_i 1 - \mathbb{\nabla}_i 1)}_{=\mathbf{0}} + \underbrace{(\widetilde{\mathbb{\nabla}}_i \mathbf{x} - \mathbb{\nabla}_i \mathbf{x})}_{=0} \cdot \mathbb{\nabla}_i f \\
&\quad = \widetilde{\mathbb{\nabla}}_i f
\end{aligned}$$

□

With this parametrization of first order consistent gradients, we can now only pay attention to the compatibility properties:

Definition 2.22 (Least-norm corrected gradient operator)**.**
The *least-norm correction for compatibility* is the discrete gradient operator $\widetilde{\mathbb{\nabla}}$ whose correction coefficients minimize the quadratic expression:

$$\sum_{i,j\in\mathcal{C}} \frac{1}{W_h(\mathbf{x}_j - \mathbf{x}_i)} \boldsymbol{\lambda}_{i,j}^2 \tag{2.54}$$

Under the constraint that $\left(\mathbb{I}_{\mathcal{C}}, \mathbb{I}_{\partial\mathcal{C}}, \widetilde{\mathbb{\nabla}}\right)$ form a compatible set of operators: $\forall\, i \in \mathcal{C}$,

$$\widetilde{\mathbb{\nabla}}_i^* 1 = \mathbf{0} \tag{2.55}$$

This last condition is a set of linear constraints restricting the set of admissible correction coefficients.

Building a gradient operator that is the least norm correction of a set of discrete operators means finding the solution of a linearly constrained quadratic optimization problem. Assuming that this problem has a unique solution, the computational effort is essentially equivalent to solving a linear system whose size is the minimum of the number of constraints and the number of unknowns (see appendix B). Here, the number of constraints is the number of nodes in the point cloud, which is necessarily smaller than the number of unknowns, given by the number of edges. Hence, building the least norm correction for compatibility comes down to solving a global sparse linear system of size n_n, which does not contradict the meshless curse (see conjecture 2.4).

The next result links the least-norm correction procedure applied to DMLS discrete gradients to the segregated approach of definition 2.20. Even though this result was proved by Pierrot more than a decade ago [180, 182], it stayed confidential as a part of ESI-Group's intellectual property and was only published recently [183]. The proof given here is the work of the author.

Proposition 2.5 (Global optimality of least-norm corrected DMLS gradient operator)**.**
Let W_h be a positive singular weight function with compact support (*i.e.* we suppose $W_h(\mathbf{0}) = +\infty$ such that $\frac{1}{W_h(\mathbf{0})} = 0$). Then assuming linear consistent compatible gradients exist, the least-norm correction (definition 2.22) of the DMLS gradient $\widetilde{\mathbb{\nabla}}^{DMLS}$ (definition D.5) coincides with Chiu's segregated gradient $\mathbb{\nabla}^{\text{Chiu}}$.

Before proving this proposition, let us explain why this result is so important. Accounting for the trivial dimension independence, finding Chiu's segregated gradient means finding the solution of d global quadratic optimization problems with $(d+2)n_n$ coupled constraints. In opposition, the least-norm correction only requires solving d global quadratic optimization problems with n_n coupled constraints in addition to n_n local problems with $d+1$ constraints (even reducible to d constraints using a similar trick). This represents a dramatic drop in the size of the global linear system, translating in lower computational time and memory costs.

Proof. Only edges (i,j) such that $W_h(\mathbf{x}_j - \mathbf{x}_i) \neq 0$ are allowed in the graph. This effectively already defines the neighborhood relation $\mathcal{N}$ and the graph of the sought gradient operator is constrained to (at most) those edges.

Let us first prove that the problem is well defined with singular weights. Indeed, in this case, the quadratic function $\mathbf{A} \mapsto \sum_{i,j\in\mathcal{C}} \frac{1}{W_h(\mathbf{x}_j - \mathbf{x}_i)} \mathbf{A}_{i,j}^2$ is not coercive because of the vanishing terms for $i = j$. However, zeroth order consistency allows us to re-write the corresponding diagonal coefficients as $\mathbf{A}_{i,i} = -\sum_{j\neq i\in\mathcal{C}} \mathbf{A}_{i,j}$, and thus we have:

$$\begin{aligned}\sum_{i\in\mathcal{C}}\sum_{j\in\mathcal{N}(i)}\mathbf{A}_{i,j}^2 &= \sum_{i\in\mathcal{C}}\mathbf{A}_{i,i}^2 + \sum_{i\in\mathcal{C}}\sum_{j\neq i\in\mathcal{N}(i)}\mathbf{A}_{i,j}^2\\ &\leq 2\sum_{i\in\mathcal{C}}\sum_{j\neq i\in\mathcal{N}(i)}\mathbf{A}_{i,j}^2\\ &\leq 2\sup_{i\neq j}W(\mathbf{x}_j-\mathbf{x}_i)\sum_{j\neq i\in\mathcal{N}(i)}\frac{1}{W_h(\mathbf{x}_j-\mathbf{x}_i)}\mathbf{A}_{i,j}^2\end{aligned}$$

Hence, the quadratic cost function is coercive on the subspace defined by the constraint $\mathbb{V}1=\mathbf{0}$.

Let us now show that the least-norm corrected compatible DMLS gradient is Chiu's segregated gradient. First, we introduce the following operator on the space of gradient coefficients:

$$P(\boldsymbol{\mu})_{i,j} \overset{\text{def}}{=} \sum_{k\in\mathcal{C}}\boldsymbol{\mu}_{i,k}\Big(\delta_{k,j}-\delta_{i,j}-\frac{1}{V_i}(\mathbf{x}_k-\mathbf{x}_i)\cdot\mathbf{A}_{i,j}^{DMLS}\Big) \tag{2.56}$$

P is such that the coefficients of $\widetilde{\mathbb{V}}^{DMLS}$ read $\overset{\sim}{\mathbf{A}}{}^{DMLS}=\mathbf{A}^{DMLS}+P(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ are the correction coefficients. We now compute $P^2(\boldsymbol{\mu})$: $\forall\ i,j\in\mathcal{C}$,

$$\begin{aligned}P^2(\boldsymbol{\mu})_{i,j} &= \sum_{k\in\mathcal{C}}P(\boldsymbol{\mu})_{i,k}\Big(\delta_{k,j}-\delta_{i,j}-\frac{1}{V_i}(\mathbf{x}_k-\mathbf{x}_i)\cdot\mathbf{A}_{i,j}^{DMLS}\Big)\\ &= \sum_{k\in\mathcal{C}}\sum_{p\in\mathcal{C}}\boldsymbol{\mu}_{i,p}\Big(\delta_{p,k}-\delta_{i,k}-\frac{1}{V_i}(\mathbf{x}_p-\mathbf{x}_i)\cdot\mathbf{A}_{i,k}^{DMLS}\Big)\Big(\delta_{k,j}-\delta_{i,j}-\frac{1}{V_i}(\mathbf{x}_k-\mathbf{x}_i)\cdot\mathbf{A}_{i,j}^{DMLS}\Big)\\ &= \sum_{p\in\mathcal{C}}\boldsymbol{\mu}_{i,p}\Big(\delta_{p,j}-\delta_{i,j}-\frac{1}{V_i}(\mathbf{x}_p-\mathbf{x}_i)\cdot\mathbf{A}_{i,j}^{DMLS}\Big)-(1-1-(\mathbf{x}_i-\mathbf{x}_i)\cdot\mathbf{A}_{i,j}^{DMLS})\\ &\quad -(\mathbf{x}_p-\mathbf{x}_i)\cdot(\mathbf{A}_{i,j}^{DMLS}-\delta_{i,j}\underbrace{V_i\mathbb{V}_i^{DMLS}1}_{=\mathbf{0}}-\underbrace{\mathbb{V}_i^{DMLS}(\mathbf{x}-\mathbf{x}_i)}_{=\mathrm{I}_d}\cdot\mathbf{A}_{i,j}^{DMLS})\\ &= \sum_{p\in\mathcal{C}}\boldsymbol{\mu}_{i,p}\Big(\delta_{p,j}-\delta_{i,j}-\frac{1}{V_i}(\mathbf{x}_p-\mathbf{x}_i)\cdot\mathbf{A}_{i,j}^{DMLS}\Big)\\ &= P(\boldsymbol{\mu})_{i,j}\end{aligned}$$

Hence, P is a linear projection. Its image is the set of correction coefficients $\boldsymbol{\mu}$ such that $P(\boldsymbol{\mu})=\boldsymbol{\mu}$, namely such that: $\forall\ i,j\in\mathcal{C}$,

$$\Big(\sum_{k\in\mathcal{C}}\boldsymbol{\mu}_{i,k}\Big)\delta_{i,j}+\frac{1}{V_i}\Big(\sum_{k\in\mathcal{C}}\boldsymbol{\mu}_{i,k}(\mathbf{x}_k-\mathbf{x}_i)\Big)\mathbf{A}_{i,j} \tag{2.57}$$

Since for all i we have $\mathrm{I}_d=\mathbb{V}_i^{DMLS}(\mathbf{x}-\mathbf{x}_i)=\sum_{j\in\mathcal{C}}\mathbf{A}_{i,j}(\mathbf{x}_j-\mathbf{x}_i)$, the vectors $(\mathbf{A}_{i,j})_{j\neq i\in\mathcal{C}}$ necessarily span $\mathbb{R}^d$. Thus, expression (2.57) implies:

$$\begin{cases}\displaystyle\sum_{k\in\mathcal{C}}\boldsymbol{\mu}_{i,k}=\mathbf{0}\\ \displaystyle\sum_{k\in\mathcal{C}}\boldsymbol{\mu}_{i,k}(\mathbf{x}_k-\mathbf{x}_i)=0\end{cases} \tag{2.58}$$

These equations characterize the image of P. Interestingly enough, this is an alternative proof of the definition 2.21.

With these newly introduced notations, the optimality conditions (cf proposition B.1) for the DMLS local optimization problems (definition D.5) read: $\forall\ \boldsymbol{\mu} : \mathcal{C} \times \mathcal{C} \to \mathbb{R}^d \mid P(\boldsymbol{\mu}) = \boldsymbol{\mu}$, we have $\mathbf{A}^{DMLS} W^{-1} \boldsymbol{\mu} = \mathbf{0}$, or using the fact that P is a projection, $\forall\ \boldsymbol{\mu} : \mathcal{C} \times \mathcal{C} \to \mathbb{R}^d$, $\mathbf{A}^{DMLS} W^{-1} P(\boldsymbol{\mu}) = \mathbf{0}$.

Similarly to the projection P onto the space of linear consistent corrections, let us introduce the projection onto the space of compatible operators: $\forall\ i, j \in \mathcal{C}$,

$$L(\boldsymbol{\mu})_{i,j} \overset{\text{def}}{=} \sum_{k \in \mathcal{C}} \boldsymbol{\mu}_{k,j} (\delta_{i,k} - \delta_{i,j}) \tag{2.59}$$

Using L, the optimality conditions for the least-norm correction for compatibility (definition 2.22) read: $\forall\ \boldsymbol{\mu} \mid L \circ P(\boldsymbol{\mu}) = P(\boldsymbol{\mu})$, we have: $\boldsymbol{\lambda} W^{-1} \boldsymbol{\mu} = 0$.
Let us now prove that $P(\boldsymbol{\lambda}) = \boldsymbol{\lambda}$: Firstly, since $P(P(\boldsymbol{\lambda}) - \boldsymbol{\lambda}) = 0 = L \circ P(P(\boldsymbol{\lambda}) - \boldsymbol{\lambda})$, we have in particular: $\boldsymbol{\lambda} W^{-1} (P(\boldsymbol{\lambda}) - \boldsymbol{\lambda}) = 0$. Let us now compute $P(\boldsymbol{\lambda}) W^{-1} (P(\boldsymbol{\lambda}) - \boldsymbol{\lambda})$:

$$\begin{aligned} P(\boldsymbol{\lambda}) W^{-1} (P(\boldsymbol{\lambda}) - \boldsymbol{\lambda}) &= \sum_{i,j \in \mathcal{C}} \frac{1}{W(\mathbf{x}_i - \mathbf{x}_j)} P(\boldsymbol{\lambda})_{i,j} (P(\boldsymbol{\lambda}) - \boldsymbol{\lambda})_{i,j} \\ &= - \sum_{i,j \in \mathcal{C}} \frac{1}{V_i W(\mathbf{x}_i - \mathbf{x}_j)} P(\boldsymbol{\lambda})_{i,j} \sum_{p \in \mathcal{C}} \boldsymbol{\lambda}_{i,p} (\mathbf{x}_p - \mathbf{x}_i) \cdot \mathbf{A}^{DMLS}_{i,j} \\ &= P(\boldsymbol{\lambda}') W^{-1} \mathbf{A}^{DMLS} \end{aligned}$$

Where $\boldsymbol{\lambda}'_{i,j} \overset{\text{def}}{=} -\frac{1}{V_i} \left(\sum_{p \in \mathcal{C}} \boldsymbol{\lambda}_{i,p} (\mathbf{x}_p - \mathbf{x}_i) \right)^T \boldsymbol{\lambda}_{i,j}$. And thus, using the optimality conditions for the DMLS local optimization problems, we have $P(\boldsymbol{\lambda}) W^{-1} (P(\boldsymbol{\lambda}) - \boldsymbol{\lambda}) = 0$. Thus, $(P(\boldsymbol{\lambda}) - \boldsymbol{\lambda}) W^{-1} (P(\boldsymbol{\lambda}) - \boldsymbol{\lambda}) = 0$, which in turn implies $P(\boldsymbol{\lambda}) = \boldsymbol{\lambda}$, and $\underset{\sim}{\mathbf{A}}^{DMLS} = \mathbf{A}^{DMLS} + P(\boldsymbol{\lambda}) = \mathbf{A}^{DMLS} + \boldsymbol{\lambda}$.

Hence, we finally have: $\forall\ \boldsymbol{\mu} : \mathcal{C} \times \mathcal{C} \to \mathbb{R}^d \mid P(\boldsymbol{\mu}) = \boldsymbol{\mu}$ and $L(\boldsymbol{\mu}) = \boldsymbol{\mu}$, we have $\underset{\sim}{\mathbf{A}}^{DMLS} W^{-1} \boldsymbol{\mu} = \mathbf{A}^{DMLS} W^{-1} \boldsymbol{\mu} + \boldsymbol{\lambda} W^{-1} \boldsymbol{\mu} = 0$. Thus, $\underset{\sim}{\mathbf{A}}^{DMLS}$ satisfies the optimality conditions of Chiu's segregated problem. □

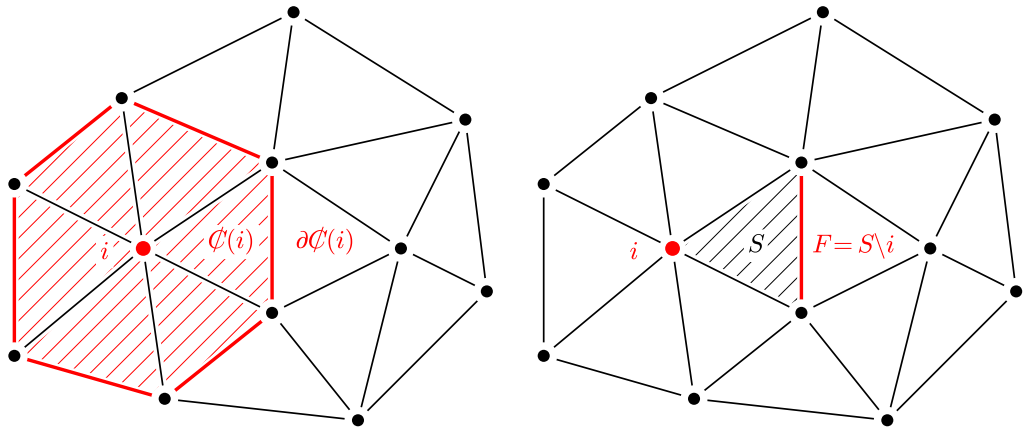
2.7 Inheriting compatibility from meshes: the construction of Smoothed Finite Element Method operators

In the last section, we defined a procedure to restore compatibility of a set of first order consistent meshless operators via the resolution of a global linear system. In this section, we illustrate the other case of the meshless curse: we show how a mesh can be used to define sets of consistent compatible discrete operators.

We first expose a nodal mesh-based example of compatible operators in section 2.7.1. Its reinterpretation in a meshless context leads to a generalization of the SFEM operators in section 2.7.2, for which we give a compatibility-preserving mesh-based integration procedure in section 2.7.3.

2.7.1 Gauss gradient: a mesh-based example of discrete operators

The aim of this section is to give a mesh-based example of nodal operators achieving both first order consistency and compatibility. The following construction is very close to those of Qian [189] and Schaap [200]. It can be generalized to the construction of meshless compatible operators in several ways as explored in sections 2.7.2 and 3.7.1.



a) b)

Figure 2.5 – *Agglomerate cells are formed around each node of the simplex mesh, and provide a well-defined cell for the definition of a nodal Gauss gradient.*
a) The agglomerate cell $\mathcal{C}(i)$ is the union of all simplices containing node i.
b) Its boundary $\partial\mathcal{C}(i)$ is a set of faces of the original simplex mesh.

Definition 2.23 (Discrete mesh-based nodal Gauss gradient)**.**
Let us suppose that Ω is a domain covered by a simplex mesh $\mathcal{M}$. We define the point cloud as the set of 0-cells (or nodes) in the mesh: $(\mathcal{C}, \mathbf{X}) = (C^0, (\mathbf{x}_i)_{i \in C^0})$. Let us define nodal volumes as the volume of the corresponding cell of the dual mesh $\mathcal{M}^*$ (see definition 1.8 and figure 1.4). In terms of the primal mesh, this is equivalent to defining the volume of a node $i \in \mathcal{C}$ as the volume of the "aggregate cell" made of all simplices containing i, scaled by a factor $\frac{1}{d+1}$ to compensate for overlaps (see figure 2.5 a)).

$$V_i^{\text{Gauss}} \overset{\text{def}}{=} \frac{1}{d+1} V(\mathcal{C}(i)) = \frac{1}{d+1} \sum_{S \ni i} V(S) \tag{2.60}$$

Then, we define the discrete gradient operator $\mathbb{\nabla}^{\text{Gauss}}$ as the Gauss gradient (see definition 1.16) on this aggregate cell $\mathcal{C}(i) \overset{\text{def}}{=} \bigcup_{S \ni i} S$ using nodal integration on simplex faces (see proposition 1.16): $\forall\, f : \mathcal{C} \to \mathbb{R}$,

$$\begin{aligned} V(\mathcal{C}(i)) \mathbb{\nabla}_i^{\text{Gauss}} f &\overset{\text{def}}{=} \sum_{F \in \partial\mathcal{C}(i)} o(F, \mathcal{C}(i)) \fint_F^{\text{Nodal}} f \\ &= \sum_{F \in \partial\mathcal{C}(i)} o(F, \mathcal{C}(i)) \frac{\mathbf{\Gamma}(F)}{d} \sum_{j \in F} f_j \end{aligned} \tag{2.61}$$

In coordinates, the above definition reads: $\forall\, i, j \in \mathcal{C}$,

$$\mathbf{A}_{i,j}^{\text{Gauss}} \overset{\text{def}}{=} \frac{1}{d(d+1)} \sum_{\substack{F \in \partial\mathcal{C}(i) \\ F \ni j}} o(F, \mathcal{C}(i)) \mathbf{\Gamma}(F) \tag{2.62}$$

This way of introducing the mesh-based nodal Gauss gradient emphasizes its link with vertex-centered finite volume methods.

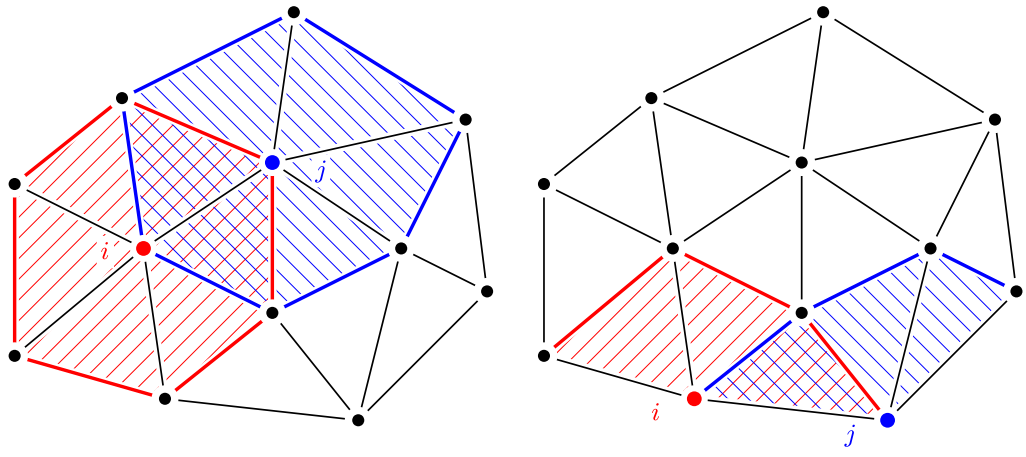


Figure 2.6 – *Edges of the graph of the Gauss gradient correspond to pairs of aggregate cells with d-simplices in their intersection. The corresponding edge vector coefficients can be expressed using the vector areas of the faces of the boundary of this intersection.*
a) If at least one node is an interior node, the boundary of the intersection of aggregate cells is made of the boundary faces of both aggregate cells.
b) If both nodes are boundary nodes, the boundary of the intersection of aggregate cells also contains the boundary faces that are incident to both nodes.

Proposition 2.6 (Linear consistency of the Gauss gradient)**.**
The discrete operator $\mathbb{\nabla}^{\text{Gauss}}$ is linear consistent.

Proof. We need to check consistency on constant and linear fields. For $f = 1$, we have:

$$\begin{aligned} V(\phi\!\!\!-(i))\mathbb{\nabla}_i^{\text{Gauss}} 1 &= \sum_{F \in \partial\phi\!\!\!-(i)} o(F, \phi\!\!\!-(i)) \fint_F^{\text{Nodal}} 1 \\ &= \sum_{F \in \partial\phi\!\!\!-(i)} o(F, \phi\!\!\!-(i)) \mathbf{\Gamma}(F) \\ &= \mathbf{0} \end{aligned}$$

For $\mathbf{a} \in \mathbb{R}^d$ and $f = \mathbf{a} \cdot \mathbf{x}$, we have:

$$\begin{aligned} V(\phi\!\!\!-(i))\mathbb{\nabla}_i^{\text{Gauss}} \mathbf{a} \cdot \mathbf{x} &= \sum_{F \in \partial\phi\!\!\!-(i)} o(F, \phi\!\!\!-(i)) \fint_F^{\text{Nodal}} \mathbf{a} \cdot \mathbf{x} \\ &= \sum_{F \in \partial\phi\!\!\!-(i)} o(F, \phi\!\!\!-(i)) \mathbf{\Gamma}(F) \mathbf{x}(F) \cdot \mathbf{a} \\ &= V(\phi\!\!\!-(i)) \mathrm{I}_d \cdot \mathbf{a} \end{aligned}$$

□

Proposition 2.7 (Alternative expression of the discrete mesh-based Gauss gradient)**.**
Let $< . >$ denote the piecewise affine reconstruction on the simplices of a mesh and N_i the

corresponding shape functions. Their piecewise expressions match the barycentric coordinates of definition 1.15. Then, the Gauss nodal volumes read: $\forall\ i \in \mathcal{C}$,

$$V_i^{\text{Gauss}} = \int_{\mathbf{x}\in\Omega} N_i(\mathbf{x})\ \mathrm{d}V \tag{2.63}$$

Moreover, the discrete mesh-based Gauss gradient has the following expression: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$V_i \mathbb{\nabla}_i^{\text{Gauss}} = \int_{\mathbf{x}\in\Omega} N_i(\mathbf{x}) \nabla < f > (\mathbf{x})\ \mathrm{d}V \tag{2.64}$$

And the gradient vector coefficients read: $\forall\ i, j \in \mathcal{C}$,

$$\mathbf{A}_{i,j}^{\text{Gauss}} = \int_{\mathbf{x}\in\Omega} N_i(\mathbf{x}) \nabla N_j(\mathbf{x})\ \mathrm{d}V \tag{2.65}$$

Proof. The expression for the nodal volumes is a consequence of proposition 1.11. Using the expression for the barycentric coordinates given in proposition 1.10, we can write the constant value gradient of the shape function N_j on the convex hull of the simplex $S = (F, j)$ as:

$$\nabla N_j = -o(F, S)\frac{1}{d}\frac{\mathbf{\Gamma}(F)}{V(S)} \tag{2.66}$$

Hence, we have: $\forall\ i \neq j \in \mathcal{C}$,

$$\begin{aligned}
\int_{\mathbf{x}\in\Omega} N_i(\mathbf{x}) \nabla N_j(\mathbf{x})\ \mathrm{d}V &= \int_{\mathbf{x}\in\not{C}(i)\cap\not{C}(j)} N_i(\mathbf{x}) \nabla N_j\ \mathrm{d}V \\
&= \sum_{S\ni i,j} \frac{1}{d+1} V(S) \times (-o(S\backslash j, S))\frac{\mathbf{\Gamma}(S\backslash j)}{d} \\
&= -\frac{1}{d(d+1)} \sum_{S\ni i,j} o(S\backslash j, S))\mathbf{\Gamma}(S\backslash j)
\end{aligned}$$

The boundary of $\not{C}(i)\cap\not{C}(j)$ decomposes into three groups of faces (see figure 2.6): faces of the form $S\backslash j$, faces of the form $S\backslash i$ and boundary faces containing both i and j. Forsaking symmetry, we can rewrite these last two groups as faces of $\partial\mathcal{C}(i)$ containing j. Thus, using the closure property (see proposition 1.13), we can rewrite the above as:

$$\int_{\mathbf{x}\in\Omega} N_i(\mathbf{x}) \nabla N_j(\mathbf{x})\ \mathrm{d}V = \frac{1}{d(d+1)} \sum_{\substack{F\in\partial\not{C}(i) \\ F\ni j}} o(F, \not{C}(i))\mathbf{\Gamma}(F) \tag{2.67}$$

Which is exactly $\mathbf{A}_{i,j}^{\text{Gauss}}$. □

Under the form given in proposition 2.7, we recognize in $\mathbb{\nabla}^{\text{Gauss}}$ the expression of nodal strains in the Node-Smoothed Finite Element Method (compare for instance with expressions (7-11) of [149] and section 6.3 of [150]). In the next section, we generalize the Gauss and the SFEM operators replacing the linear interpolation on a mesh with any kind of (possibly meshless) reconstruction.

Remark. Using the expression for the barycentric coordinates given in proposition 1.10, we can easily deduce the following formula: $\forall\ i, j \in \mathcal{C}$, $\forall\ \mathbf{x} \in \Omega$,

$$N_i(\mathbf{x}) \nabla N_j(\mathbf{x}) = -\frac{\partial N_j}{\partial \mathbf{x}_i}(\mathbf{x}) \tag{2.68}$$

This proves that the Gauss discrete operator can also be thought of as a volume-based SFEM operator. This family of meshless operators is studied in section 3.7.1.

2.7.2 The general construction of Smoothed Finite Element Method discrete operators

The Node-Smoothed Finite Element Method is a recent mesh-based simulation method developed by Liu et al. in [147, 148, 149]. The main idea underlying the construction of Smoothed Finite Element Method (SFEM) is to define a so-called *smoothed strain* (analogous to our discrete gradient operator) by integration on dual cells of the mesh, hence providing a node-based definition of the strain.

An interesting feature of the Node SFEM is to provide an upper bound of the unknown exact solution on elliptic problems. This is especially convenient when used in conjunction with the classical P-1 Finite Element Method, which provides a lower bound. Recent reviews of the Smoothed Finite Element Methods principles can be found in [150, 235].

In this section, we generalize the construction of section 2.7.1 that was found identical to the SFEM on simplicial meshes to arbitrary shape functions. For this reason, we call the corresponding meshless discrete operator "SFEM operators". This is not the first time that these operators appear in a meshless context, as they are almost identical to the MLSPH operators devised by Dilts [77, 78]. The difference with Dilts' work however is that we do not consider these definitions as "a convenient theoretical device to guide us to a reasonable evolution equation", but as a completely legitimate set of discrete operators.

Definition 2.24 (Smoothed Finite Element Method discrete integration operator)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud. Let us assume given a reconstruction operator $< . >$ on $(\mathcal{C}, \mathbf{X})$ with shape functions $(\phi_i)_{i\in\mathcal{C}}$. Inspired by the expression of the Gauss volumes given in expression (2.63), we define the SFEM discrete volume integration operator as: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$\oint\!\!\!\!\!\!\square_{\mathcal{C}}^{SFEM} f \stackrel{\text{def}}{=} \int_{\mathbf{x}\in\Omega} < f > (\mathbf{x})\ \mathrm{d}V \tag{2.69}$$

The corresponding volume weights read:

$$V_i^{SFEM} \stackrel{\text{def}}{=} \int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x})\ \mathrm{d}V \tag{2.70}$$

In a sense, the nodal basis functions ϕ_i play the part of the dual cells of a mesh. Even if they might overlap, the fact that constant functions are exactly reproduced means that basis functions sum to unity on the whole domain. This property, called partition of unity is a relaxed version of the requirement that the cells tile the domain.

Proposition 2.8 (Reproducibility entails exactness of discrete integration)**.**
With the above notations, if $< . >$ reproduces a continuous function f on Ω, then the SFEM discrete integration operator is exact on f.

Proof. The proof is a direct consequence of the definition. Reproducibility reads: $\forall\ \mathbf{x} \in \Omega$,

$$f(\mathbf{x}) = \sum_{i\in\mathcal{C}} \phi_i(\mathbf{x}) f(\mathbf{x}_i)$$

Integrating both sides of the above expression and using linearity of the integral, we get:

$$\int_{\mathbf{x}\in\Omega} f(\mathbf{x})\ \mathrm{d}V = \sum_{i\in\mathcal{C}} V_i^{SFEM} f(\mathbf{x}_i)$$

Hence, $\oint\!\!\!\!\!\!\square_{\mathcal{C}}^{SFEM}$ is exact on f. □

In particular, proposition 2.8 shows that it is possible to build discrete integration operators that have arbitrarily high order of consistency. The next definition is inspired by proposition 2.7: substituting the mesh-based barycentric coordinates with any set of meshless shape functions generates the following discrete gradient operator.

Definition 2.25 (Smoothed Finite Element Method discrete gradient)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud, and $< . >: (\mathcal{C} \to \mathbb{R}) \to (\mathbb{R}^d \to \mathbb{R})$ a reconstruction operator on the point cloud with shape functions $(\phi_i)_{i\in\mathcal{C}}$. We define the following SFEM discrete gradient operator: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ i \in \mathcal{C}$,

$$V_i \nabla_i^{SFEM} f \overset{\text{def}}{=} \int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x}) \nabla < f > (\mathbf{x})\ \mathrm{d}V \tag{2.71}$$

In coordinates, the above operator reads: $\forall\ i, j \in \mathcal{C}$,

$$\mathbf{A}_{i,j}^{SFEM} \overset{\text{def}}{=} \int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x}) \nabla \phi_j(\mathbf{x})\ \mathrm{d}V \tag{2.72}$$

Remark. We readily see using expression (2.72) that edges (i, j) of the graph of the SFEM discrete gradient are such that the intersection of the supports of ϕ_i and ϕ_j has nonzero measure. These overlaps are the SFEM analogues of oriented faces of a mesh, with vector surface areas given by the gradient coefficients $\mathbf{A}_{i,j}$.

If for instance (as in the MLS case, see section D.2), $\mathrm{supp}(\phi_i) \subset B^d(\mathbf{x}_i, h)$, then $\mathbf{A}_{i,j}^{SFEM} \neq \mathbf{0}$ implies $\|\mathbf{x}_j - \mathbf{x}_i\| < 2h$. In other words, ∇^{SFEM} inherits its locality property from the reconstruction operator $< . >$. This idea is further developed on figure 2.7.

Proposition 2.9 (Linear consistency of the SFEM discrete gradient)**.**
With the above notations, if $< . >$ reproduces affine functions on Ω, then ∇^{SFEM} is linear consistent.

Proof. Let $\mathbf{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. We have: $\forall\ i \in \mathcal{C}$,

$$\begin{aligned} V_i \nabla_i^{SFEM}(\mathbf{a}\cdot\mathbf{x} + b) &= \int_{\mathbf{y}\in\Omega} \phi_i(\mathbf{y}) \nabla < \mathbf{a}\cdot\mathbf{x} + b > (\mathbf{y})\ \mathrm{d}V \\ &= \int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x}) \nabla(\mathbf{a}\cdot\mathbf{x} + b)\ \mathrm{d}V \\ &= \left(\int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x})\ \mathrm{d}V\right)\mathbf{a} \\ &= V_i^{SFEM}\mathbf{a} \end{aligned} \tag{2.73}$$

Hence, ∇^{SFEM} is linear consistent. □

Note that this computation also incidentally shows that consistency of higher order is not achieved even if the reconstruction operator reproduces higher order polynomials.

Definition 2.26 (SFEM discrete boundary integration and dual gradient)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud, and $< . >: (\mathcal{C} \to \mathbb{R}) \to (\mathbb{R}^d \to \mathbb{R})$ a reconstruction operator on the point cloud with shape functions $(\phi_i)_{i\in\mathcal{C}}$. We define the following symmetric SFEM discrete boundary integration operator: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ \mathbf{u} : \mathcal{C} \to \mathbb{R}^d$,

$$\oint_{\partial\mathcal{C}}^{SFEM} f\mathbf{u} \overset{\text{def}}{=} \int_{\mathbf{x}\in\partial\Omega} < f > (\mathbf{x}) < \mathbf{u} > (\mathbf{x}) \cdot \mathbf{n}\ \mathrm{d}S \tag{2.74}$$

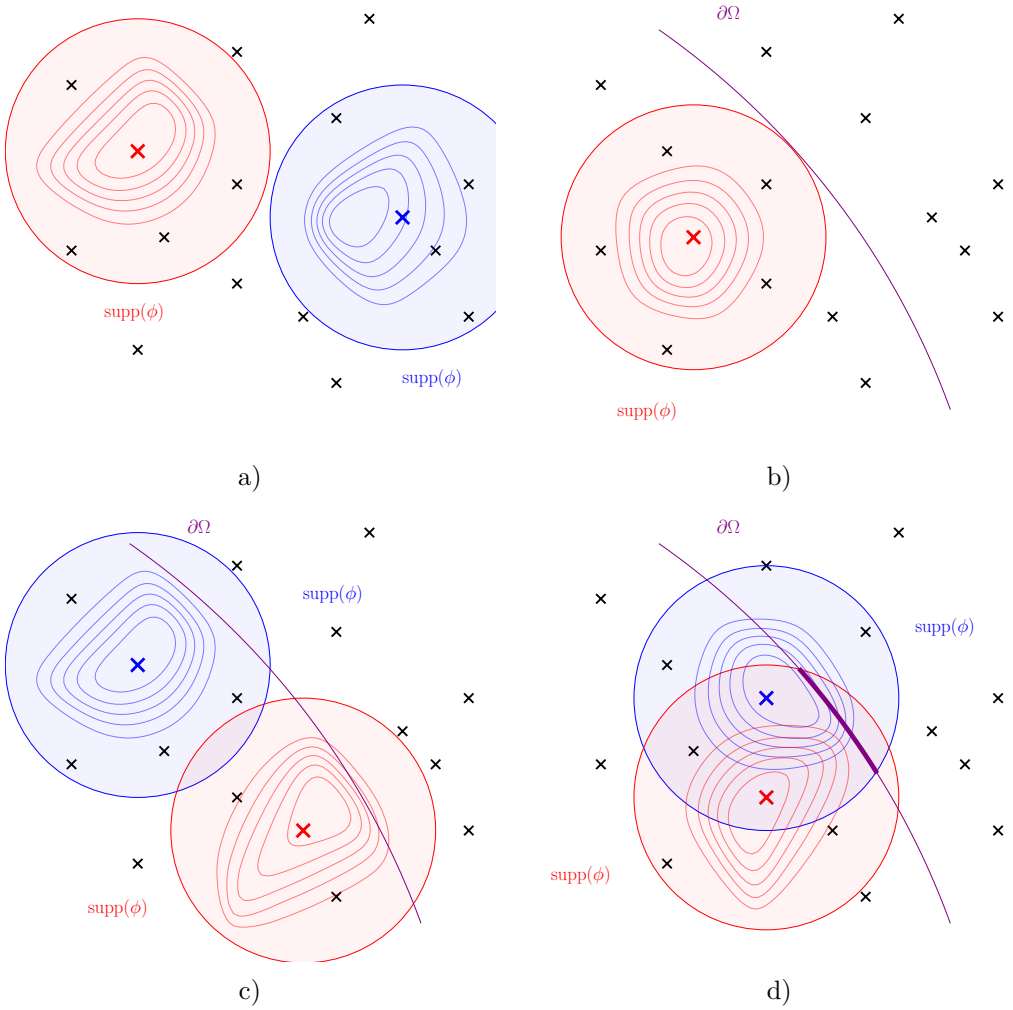


Figure 2.7 – *Intersections of supports of the shape functions give the graph of the SFEM operators. Several iso-values of the Shepard shape functions are represented above.*
a) The supports of the red node and the blue node do not intersect, thus $\mathbf{A}_{blue,red} = \mathbf{A}_{red,blue} = \mathbf{0}$*: neither* $(blue, red)$ *nor* $(red, blue)$ *are edges of the graph of* ∇^{SFEM}*.*
b) The red node is not a boundary node since the intersection of $\mathrm{supp}(\phi)$ *with the boundary* $\partial\Omega$ *has non vanishing measure.*
c) If a shape function intersects the boundary $\partial\Omega$*, then the corresponding node is a boundary node. Hence both the red node and the blue are boundary nodes. Moreover, these two supports intersect, so* $\mathbf{A}_{red,blue} \neq \mathbf{0}$*. However,* $\mathrm{supp}(\phi) \cap \mathrm{supp}(\phi)$ *does not intersect the boundary, hence* $\mathbf{\Gamma}_{red,blue} = \mathbf{\Gamma}_{blue,red} = \mathbf{0}$*.*
d) The intersection of the supports of the two shape functions and the boundary is non-empty, thus $\mathbf{\Gamma}_{red,blue} = \mathbf{\Gamma}_{blue,red} \neq \mathbf{0}$*.*

In coordinates, the above operator reads: $\forall\ i,j \in \mathcal{C}$,

$$\mathbf{\Gamma}_{i,j}^{SFEM} \overset{\text{def}}{=} \int_{\mathbf{x}\in\partial\Omega} \phi_i(\mathbf{x})\phi_j(\mathbf{x})\mathbf{n}\ \mathrm{d}S \tag{2.75}$$

With this definition, the gradient operator is symmetric: $\mathbb{\nabla}^{SFEM} = \mathbb{\nabla}^{SFEM\ *}$. In particular, this means that $\mathbb{\nabla}^{SFEM\ *}$ is linear consistent, and consequently, $\left(\mathbb{f}_{\mathcal{C}}^{SFEM}, \mathbb{f}_{\partial\mathcal{C}}^{SFEM}, \mathbb{\nabla}^{SFEM}\right)$ form a set of compatible discrete operators.

Proof. Using integration by parts, we have: $\forall\ i,j \in \mathcal{C}$,

$$\begin{aligned}
\mathbf{A}_{i,j}^{SFEM} &= \int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x})\nabla\phi_j(\mathbf{x})\ \mathrm{d}V \\
&= -\int_{\mathbf{x}\in\Omega} \phi_j(\mathbf{x})\nabla\phi_i(\mathbf{x})\ \mathrm{d}V + \int_{\mathbf{x}\in\partial\Omega} \phi_i(\mathbf{x})\phi_j(\mathbf{x})\mathbf{n}\ \mathrm{d}S \\
&= -\mathbf{A}_{j,i}^{SFEM} + \mathbf{\Gamma}_{i,j}^{SFEM} \\
&= \mathbf{A}_{i,j}^{SFEM\ *}
\end{aligned}$$

□

Remark. We have devised a compatible set of discrete meshless operators. Does it necessarily mean that the SFEM operators break the meshless curse (see conjecture 2.4)? The main difficulty with the SFEM operators is to compute the exact integrals of the shape functions (*e.g.* $\int_\Omega \phi_i(\mathbf{x})\ \mathrm{d}V$): unfortunately, there is no currently known set of local and linear consistent meshless shape functions with closed-form integrals. Thus, the SFEM operators cannot be exactly computed in practice, only approximated, and their existence thus does not contradict conjecture 2.4.

The difficulty of the discrete integration process consists in preserving the good properties of the SFEM operators. In the next section, we exploit the structure of a background integration mesh to define a compatible set of mesh-integrated SFEM operators.

2.7.3 A mesh integration procedure for the Smoothed Finite Element Method operators

Since we have in general no closed form formula to compute the SFEM operators defined in section 2.7.2, we need to use a numerical quadrature method. Contrary to most discrete integration procedures for meshless methods (see [155] for a quick review), we do not focus the analysis on the precision of the formula, but rather on the exact conservation of interesting properties of the exactly integrated case. The numerical quadrature of the SFEM operators has already been attempted by Dilts [77, 78] in a fully meshless context, but his methods either fail to transfer compatibility to the discrete level, or leads to non-local boundary operators.

In this section, we propose to use a background integration mesh to give approximation of these operators. Our approach is directly inspired by the Gauss approximation of section 1.3.2.2 and bears strong similarities with the older stabilized conforming nodal integration method of Chen et al. (see [55, 56]), where discrete integration is performed on the Voronoi tessellation (see the definition given in expression F.1) of the point cloud. We prove that this approach is able to preserve the locality, the first order consistency and the compatibility of the exact SFEM operators.

Let us define the following mesh-integrated SFEM discrete operators on the point cloud:

Definition 2.27 (Mesh-integrated SFEM discrete operators)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud, and $< . >: (\mathcal{C} \to \mathbb{R}) \to (\mathbb{R}^d \to \mathbb{R})$ a reconstruction operator on the point cloud with shape functions $(\phi_i)_{i\in\mathcal{C}}$. Let $\mathcal{M}$ be a d-dimensional mesh with oriented cells (see section F). The discrete volume integration method is defined as the approximation of expression (2.69) using the one point integration method (definition 1.17) on cells of the mesh: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$\fint_{\mathcal{C}}^{SFEM\ \mathcal{M}} f \overset{\text{def}}{=} \fint_{\mathcal{M}} < f > \tag{2.76}$$

The corresponding nodal volume weights read: $\forall\ i \in \mathcal{C}$,

$$V_i^{SFEM\ \mathcal{M}} \overset{\text{def}}{=} \sum_{\phi^d \in \mathcal{M}} V(\phi^d)\phi_i(\mathbf{x}(\phi^d)) \tag{2.77}$$

Similarly, we define the discrete boundary integration method as the approximation of expression (2.74) using the one point integration method (definition 1.17) on the boundary mesh $\partial\mathcal{M}$: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ \mathbf{u} : \mathcal{C} \to \mathbb{R}^d$,

$$\fint_{\partial\mathcal{C}}^{SFEM\ \mathcal{M}} f\mathbf{u} \overset{\text{def}}{=} \fint_{\partial\mathcal{M}} < f >< \mathbf{u} > \tag{2.78}$$

The corresponding vector boundary integration weights read: $\forall\ i \in \mathcal{C}$

$$\mathbf{\Gamma}_{i,j}^{SFEM\ \mathcal{M}} \overset{\text{def}}{=} \sum_{\phi^{d-1} \in \partial\mathcal{M}} \mathbf{\Gamma}(\phi^{d-1})\phi_i(\mathbf{x}(\phi^{d-1}))\phi_j(\mathbf{x}(\phi^{d-1})) \tag{2.79}$$

In the above expression, the orientation of mesh boundary faces is chosen such that vector face areas point outwards.

Finally, the discrete gradient is defined as the approximation of expression (2.71) combining the one point integration method (proposition 1.17) and the one-point integrated Gauss gradient (definition 1.17) on cells of the mesh: $\forall\ i \in \mathcal{C}$, $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$V_i \mathbb{\nabla}_i^{SFEM\ \mathcal{M}} f \overset{\text{def}}{=} \sum_{\phi^d \in \mathcal{M}} \phi_i(\mathbf{x}(\phi^d)) \sum_{\phi^{d-1} \in \partial\phi^d} o(\phi^{d-1}, \phi^d)\mathbf{\Gamma}(\phi^{d-1}) < f > (\mathbf{x}(\phi^{d-1})) \tag{2.80}$$

The corresponding gradient coefficients read:

$$\mathbf{A}_{i,j}^{SFEM\ \mathcal{M}} f \overset{\text{def}}{=} \sum_{\phi^d \in \mathcal{M}} \phi_i(\mathbf{x}(\phi^d)) \sum_{\phi^{d-1} \in \partial\phi^d} o(\phi^{d-1}, \phi^d)\mathbf{\Gamma}(\phi^{d-1})\phi_j(\mathbf{x}(\phi^{d-1})) \tag{2.81}$$

Note that these expressions (2.80) and (2.81) do not require the evaluation of the gradient of the shape functions.

Remark. The construction of the mesh-integrated SFEM operators only requires a mesh for the discrete integration procedure. The reconstruction operator can be chosen to be purely meshless (see for instance the MLS reconstruction operator of definition D.4).

The discrete operators defined above are similar to the exact SFEM discrete operators, but the discrete nature of the integration on the mesh subtly changes the properties of the mesh-integrated SFEM operators compared to the exact ones. Some other features are lost:

the gradient is no longer symmetric: $\mathbb{\nabla}^{SFEM\ \mathcal{M}} \neq \mathbb{\nabla}^{SFEM\ \mathcal{M}\ *}$, and the dual gradient is not linear consistent.

However, the discrete integration method cleverly takes advantage of the structure of the mesh and retains the most important features of the exact integration: the mesh-integrated SFEM discrete operators are local and form a compatible set of operators!

Proposition 2.10 (Compatibility of the mesh-integrated SFEM operators)**.**
If the reconstruction operator $< . >$ is first order consistent, then the mesh-integrated SFEM operators form a compatible set of discrete operators.

Proof. Let us first check that $\mathbb{\nabla}^{SFEM\ \mathcal{M}}$ is linear consistent: $\forall\ \mathbf{a} \in \mathbb{R}^d$, $\forall\ b \in \mathbb{R}$, $\forall\ i \in \mathcal{C}$,

$$\begin{aligned} V_i \mathbb{\nabla}_i^{SFEM\ \mathcal{M}} \mathbf{a} \cdot \mathbf{x} + b &= \sum_{\phi' \in \mathcal{M}} \phi_i(\mathbf{x}(\phi')) \sum_{\phi'^{d-1} \in \partial \phi'} o(\phi'^{d-1}, \phi') \mathbf{\Gamma}(\phi'^{d-1}) < \mathbf{a} \cdot \mathbf{x} + b > (\mathbf{x}(\phi'^{d-1})) \\ &= \sum_{\phi' \in \mathcal{M}} \phi_i(\mathbf{x}(\phi')) \sum_{\phi'^{d-1} \in \partial \phi'} o(\phi'^{d-1}, \phi') \mathbf{\Gamma}(\phi'^{d-1}) (\mathbf{a} \cdot \mathbf{x}(\phi'^{d-1}) + b) \\ &= \sum_{\phi' \in \mathcal{M}} \phi_i(\mathbf{x}(\phi')) V(\phi') \mathbf{a} \\ &= V_i^{SFEM\ \mathcal{M}} \mathbf{a} \end{aligned}$$

Now we need to check that Stokes's theorem holds at a discrete level: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$\begin{aligned} \oint\!\!\!\!\!\!\square_{\mathcal{C}}^{SFEM\ \mathcal{M}} \mathbb{\nabla} f &= \sum_{i \in \mathcal{C}} V_i \mathbb{\nabla}_i^{SFEM\ \mathcal{M}} f \\ &= \sum_{i \in \mathcal{C}} \sum_{\phi' \in \mathcal{M}} \phi_i(\mathbf{x}(\phi')) \sum_{\phi'^{d-1} \in \partial \phi'} o(\phi'^{d-1}, \phi') \mathbf{\Gamma}(\phi'^{d-1}) < f > (\mathbf{x}(\phi'^{d-1})) \\ &= \sum_{\phi' \in \mathcal{M}} \sum_{\phi'^{d-1} \in \partial \phi'} o(\phi'^{d-1}, \phi') \mathbf{\Gamma}(\phi'^{d-1}) < f > (\mathbf{x}(\phi'^{d-1})) \\ &= \sum_{\phi'^{d-1} \in \mathcal{M}} \mathbf{\Gamma}(\phi'^{d-1}) < f > (\mathbf{x}(\phi'^{d-1})) \sum_{\phi' \ |\ \phi'^{d-1} \in \partial \phi'} o(\phi'^{d-1}, \phi') \end{aligned}$$

The sum $\sum_{\phi' \ |\ \phi'^{d-1} \in \partial \phi'} o(\phi'^{d-1}, \phi')$ vanishes except for all faces of the mesh that are not in $\partial \mathcal{M}$. Assuming that these boundary faces are positively oriented, we have:

$$\begin{aligned} \oint\!\!\!\!\!\!\square_{\mathcal{C}}^{SFEM\ \mathcal{M}} \mathbb{\nabla} f &= \sum_{\phi'^{d-1} \in \partial \mathcal{M}} \mathbf{\Gamma}(\phi'^{d-1}) < f > (\mathbf{x}(\phi'^{d-1})) \\ &= \oint\!\!\!\!\!\!\square_{\partial \mathcal{C}}^{SFEM\ \mathcal{M}} 1 f \end{aligned}$$

Which is the desired result. □

Note that the compatibility of the mesh-integrated SFEM operators does not contradict the meshless curse (conjecture 2.4) since we used a background integration mesh.

Remark. The graph of the mesh-integrated SFEM gradient operator is quite different from its exactly integrated counterpart, especially if $\mathcal{M}$ is coarse compared to $\mathcal{C}$. An edge (i, j) is in the graph of $\mathbb{\nabla}^{SFEM\ \mathcal{M}}$ if there is a cell ϕ' in $\mathcal{M}$ whose centroid $\mathbf{x}(\phi')$ is in $\text{supp}(\phi_i)$ and if one of its faces $\phi'^{d-1} \in \partial \phi'$ has its centroid in $\text{supp}(\phi_j)$. Hence, it is possible that $\mathbf{A}_{i,j}^{SFEM\ \mathcal{M}} = \mathbf{0}$ even if $\text{supp}(\phi_i) \cap \text{supp}(\phi_j)$ has non empty interior. In their mesh-integrated form however, it is possible that $\mathbf{A}_{i,j}^{SFEM\ \mathcal{M}} \neq \mathbf{0}$ even if $\text{supp}(\phi_i) \cap \text{supp}(\phi_j) = \varnothing$! Examples of such cases are shown in figure 2.8.

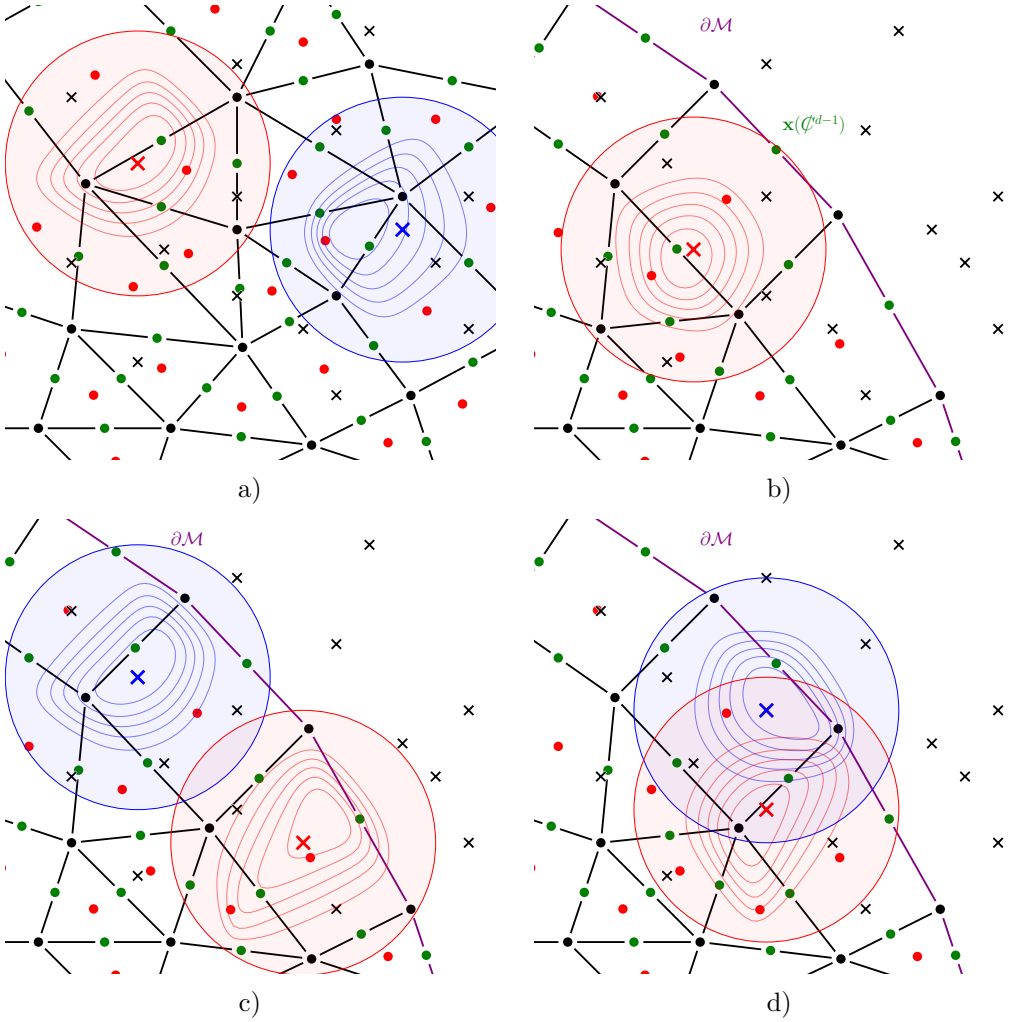


Figure 2.8 – *Discrete mesh-integrated SFEM operators can have a slightly different graph from their exact analogues (see figure 2.7), especially if the mesh $\mathcal{M}$ is coarse compared to the point cloud.*
a) The red node and the blue node have non-intersecting supports. Still, there is a cell whose centroid is in $\text{supp}(\phi)$ *and which has a face whose centroid is in* $\text{supp}(\phi)$*. Hence, (blue, red) is an edge of the graph of* $\nabla^{SFEM}\,\mathcal{M}$*.*
b) Even if the intersection of $\text{supp}(\phi)$ *with the exact boundary* $\partial\Omega$ *is empty, the centroid of a boundary face lies inside* $\text{supp}(\phi)$*. Thus,* $\mathbf{\Gamma}_{red,red} \neq \mathbf{0}$ *and the red node is a boundary node.*
c) Both the red and the blue nodes are boundary nodes since each corresponding support contains the centroid of a face of the boundary of the mesh. Moreover, $\mathbf{\Gamma}_{red,blue} = \mathbf{\Gamma}_{blue,red} = \mathbf{0}$ *as in the continuous case. There is a cell of the mesh whose centroid is in* $\text{supp}(\phi)$ *with a face whose centroid is in* $\text{supp}(\phi)$*, so* $\mathbf{A}_{blue,red} \neq \mathbf{0}$*. However, there is no cell of the mesh whose centroid is in* $\text{supp}(\phi)$ *and with a face whose centroid is in* $\text{supp}(\phi)$*, so* $\mathbf{A}_{red,blue} = \mathbf{0}$*. In particular, it is clear that* $\nabla^{SFEM}\,\mathcal{M}$ *is not symmetric contrary to the continuous case.*
d) Both the red and the blue nodes are boundary nodes. However, $\text{supp}(\phi) \cap \text{supp}(\Psi)$ *does not contain the centroid of a face of the boundary of the mesh, which means that* $\mathbf{\Gamma}_{red,blue} = \mathbf{\Gamma}_{blue,red} = \mathbf{0}$ *contrary to the continuous case.*

Similarly, an edge (i,j) is in the graph of $\oiint_{\partial\mathcal{C}}^{SFEM\ \mathcal{M}}$ if there is a boundary face $\mathcal{C}^{d-1}$ whose centroid is in the supports of both nodes: $\mathbf{x}(\mathcal{C}^{d-1}) \in \text{supp}(\phi_i) \cap \text{supp}(\phi_j)$. Hence, the graph of the boundary integration operator is affected by two factors. Firstly, if $\mathcal{M}$ does not accurately capture the geometry of Ω (for instance in the presence of curved boundaries), there can be some disagreements between the two graphs, but this is an expectable side-effect of discretization. More subtly, if $\text{supp}(\phi_i) \cap \text{supp}(\phi_j)$ does intersect the discrete mesh boundary, but not at a face centroid, then $\mathbf{\Gamma}_{i,j} = \mathbf{0}$.

In order to complete the discretization of Neumann and Dirichlet boundary conditions using Nitsche's weak form (as presented in section 4.2.1), we need to give discrete versions of $\int_{\partial\Omega} f\mathbf{v}\ \mathrm{d}S$ where one of the fields is discrete, and the other one is continuous. A straightforward adaptation of expression (2.78) simply yields for $f : \partial\Omega \to \mathbb{R}$ and $\mathbf{u} : \mathcal{C} \to \mathbb{R}^d$,

$$\oiint_{\partial\mathcal{C}}^{SFEM\ \mathcal{M}} f\mathbf{u} \overset{\text{def}}{=} \oiint_{\partial\mathcal{M}} f < \mathbf{u} > \tag{2.82}$$

This discretization coincides with $\oiint_{\partial\mathcal{C}}^{SFEM\ \mathcal{M}} f\mathbf{u}$ for a reproduced field f and it only requires values of f at the boundary.

A quick summary

This second chapter is dedicated to the exposition of a nodal meshless framework. In this framework, the geometric basis for the discretization of fields is the point cloud (hence the name "meshless"), and the reduction operator, which maps continuous fields onto discrete fields, is chosen to be evaluation at nodes of the point cloud (hence the name "nodal").

In section 2.2, we define several meshless discrete operators: volume and boundary integration, differentiation and reconstruction. These operators are recognized as a meshless substitute for the structure defined by a mesh. Volumes are associated to nodes giving them a status similar to that of cells of a mesh. The boundary integration and differentiation operators define graphs on the point cloud, and associate vector areas to edges of this graph. Although this graph does not come natively with a chain complex structure (we cannot define a boundary operator ∂ verifying $\partial \circ \partial = 0$), it is the meshless analogue of the incidence relation in an abstract cell complex as it defines a neighborhood relation between nodes.

The quality of the meshless operators is assessed in section 2.3 with the notions of exactness and consistency order. We also describe renormalization techniques for the recovery of zeroth and first order consistency of meshless discrete gradient operators.

In section 2.4, we use the duality pairing of discrete fields to define a dual gradient operator inspired by the integration by parts formula. Consistency of the dual gradient operator characterizes how well the initial volume integration, boundary integration and differentiation operators work together. Its consistency of order zero – which we call compatibility – is equivalent to a discrete Stokes formula and is a paramount feature of the discretization. It can be viewed as a meshless equivalent of the idea that faces of a mesh enclose the volume defined by cells, and that these cells are arranged without superimposition or gaps.

The computational effort of devising first order consistent compatible sets of meshless operators is speculated to be at least that of solving a global system of linear equations, or that of effectively building a mesh. We call this conjecture the "meshless curse", and it is the most important notion of this chapter.

In section 2.5, we fully detail the construction of the classical SPH operators. Their poor compatibility properties pushes us to devise a correction procedure to make first order consistent gradients compatible. This construction obeys the meshless curse as a global

linear system needs to be solved. We also prove that corrected DMLS gradients are globally optimal, lowering the cost of building these operators.

Finally, we give a generalization of the first order consistent compatible SFEM operators, and propose a mesh-based discrete integration method that conserves compatibility. Again, this method does not break the meshless curse since its compatibility properties are inherited from the mesh via the discrete integration procedure.

In the next chapter, we fully develop a new theory for the generation of meshless discretization operators. Its key ingredient is to exploit the dependency of the meshless nodal volume integration weights with respect to geometrical parameters (node positions in particular) to define all other meshless operators.

Chapter 3

The Volume Method

Contents

In chapter 2, we have introduced the notions of point clouds and meshless operators. On a point cloud $(\mathcal{C}, \mathbf{X})$, our discrete equivalent of the computational domain $\Omega \subset \mathbb{R}^d$, we have defined four main discrete operators: a volume integration operator $\oint_{\mathcal{C}}$ with volume weights $(V_i)_{i\in\mathcal{C}}$ (definition 2.6) , a boundary integration operator $\oint_{\partial\mathcal{C}}$ with surface vector weights $(\mathbf{\Gamma}_{i,j})_{i,j\in\mathcal{C}}$ (definition 2.7), a discrete gradient operator $\mathbb{\nabla}$ with vector coefficients $(\mathbf{A}_{i,j})_{i,j\in\mathcal{C}}$

(definition 2.9), and a reconstruction operator $< . >$ with shape functions $(\phi_i)_{i\in\mathcal{C}}$ (definition 2.10). These operators can be combined to define a fifth one: the dual gradient $\mathbb{V}^*$ with vector coefficients $(\mathbf{A}^*_{i,j})_{i,j\in\mathcal{C}}$ (definition 2.17).

In this chapter, we motivate the idea that the choice of an appropriate procedure to define the volume integration operator $\oint\!\!\!\!\!\!\int_{\mathcal{C}}$ is sufficient to define all other discrete operators. The key ingredient is to exploit the dependency of the volume weights with respect to the geometrical parameters $\mathbf{X}$, Ω and $\mathcal{C}$. This general idea is not new and has already been used successfully to define both mesh-based and meshless discretization methods.

Early versions of the concept (see [121, 156, 199]) already featured derivatives of the volume of cells of a logically rectangular grid with respect to node position. In [153], Loubère et al. argue that this method of defining discrete operators is the only sensible method to reconcile energy and entropy conservation within a staggered Lagrangian scheme. In [51, 62], Carré and coworkers use derivatives of the volume of cells to devise a cell-centered Lagrangian hydrodynamics scheme. Their scheme achieves both weak consistency [73] and conservation of linear and angular momenta [74], and their method of defining discrete operators can be considered as a mesh-based analogue of the method described in this chapter.

In the meshless community, the earliest usage of volume-based operators can be found in the work of Gingold and Monaghan (see [104] for instance) with the Lagrangian interpretation of SPH, closely followed by the work of Mikhailova et al. (see [158, 210] for instance) where similar definitions are given on a Voronoi tessellation. While the SPH community focused their efforts towards conservation properties of the scheme (see [161, 163]), resorting to sometimes *ad-hoc* stabilization procedures to improve consistency or avoid the so-called "tensile instability", a term coined by Swegle et al. in [220] (also see [221] by the same author on the subject) and characterized by a catastrophic failure of the method when a tensile stress threshold is reached.

The two main ideas proposed in the literature to get rid of the tension instability have been either to improve the discrete integration procedure (see [82, 83] for instance) with the introduction of so-called "stress points" *i.e.* additional points for the discrete nodal integration staggered with respect to the point cloud, or by introducing additional terms in the advection equation (see [160, 164]), a method mainly referred to as modified SPH or XSPH. In this chapter (specifically in sections 3.3 and 3.6), we give novel interpretations of the role of these modifications and reinterpret them both as achieving the very same goal: improving the consistency of the dual gradient.

To the author's knowledge, volume-based operators have not gained much traction besides SPH and Voronoi-based volumes. Even though some researchers have slightly generalized the approach (see the work of Serrano et al. in [87, 204, 205, 206] and that of de Goes et al. in [67] for example), no radically new propositions have been made in the past decades, and a general theory of volume-based operators is still missing.

The aim of this chapter is to fulfill this need. Even if the historical motivation for the definition of volume-based operators is to develop discretizations of the equations of hydrodynamics (see appendix G), we show in this chapter that they can have all the desired properties for the discretization of elliptic equations, and give examples of successful use in this context in chapter 4.

We first recall this volume-based construction of the primal gradient operator and widely extend existing characterizations of its consistency properties, even proving a previously unknown impossibility result.

Then, we separate the dependency of the volume weights with respect to node positions into two parts. The first contribution is purely nodal. We interpret it as the (opposite of the) volume-based dual gradient operator, and give local and global conditions for its consistency. Then, we generalize the construction of Sibson shape functions to general volume functions

(not necessarily based on a Voronoi diagram), and study its tight relationship with the dual gradient operator.

The second contribution to the volume weights is a composite dependency involving variations with respect to the computational domain Ω. This last term – which we recognize as a boundary integration operator – is presented here for the first time. It provides a means to impose boundary conditions within the volume-based operator context and without resorting to *ad-hoc* procedures as is commonplace in the literature.

Finally, we adapt the SPH and SFEM ideas to construct volume-based discrete operators. In particular, we give a theoretical construction of operators with maximal primal consistency and of arbitrary dual consistency.

3.1 Volume integration bears a gradient operator

The foundation of the so-called *volume method* is the following definition of a volume function:

Definition 3.1 (Volume function)**.**
A *volume function* is a procedure to compute a volume integration operator from geometrical parameters:

$$V : \mathcal{P}(\mathbb{R}^d) \times \mathbb{N} \times (\mathbb{N} \to \mathbb{R}^d) \to (\mathbb{N} \to \mathbb{R}) \tag{3.1}$$
$$(\Omega, \mathcal{C}, \mathbf{X}) \mapsto V(\Omega, \mathcal{C}, \mathbf{X})$$

Namely, a volume function is a procedure that defines the value of volume weights as a function of:

- The shape of the continuous computational domain Ω
- The (number of) nodes in the point cloud: $\mathcal{C} = [\![1, n_n]\!]$
- The position of these nodes: $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{n_n})$

Assuming the existence of a volume function is actually a very weak requirement. Volume weights do need to be generated somehow, and the definition of the volume function is only the realization that volume weights might vary as geometrical parameters of the problem vary. We have already given examples of volume functions in chapter 2, which are revisited in sections 3.5 and 3.7.

Definition 3.2 (Node indiscernibility)**.**
A volume function V is said *node indiscernible* if it is invariant with respect to node relabeling, *i.e.* under shuffling of node positions, or in mathematical terms if: $\forall\ \pi \in Sym(\mathcal{C})$, $\forall\ (\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$, $\forall\ i \in \mathcal{C}$,

$$V_{\pi(i)}(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) = V_i(\Omega, \mathcal{C}, \mathbf{x}_{\pi(1)}, \cdots, \mathbf{x}_{\pi(n_n)}) \tag{3.2}$$

In a sense, we can argue that methods that are the "most meshless" respect node indiscernibility. Indeed, any transgression of invariance with respect to node re-labeling requires the definition of an additional structure on the point cloud. This additional structure can for instance be a set of nodal weights (like mass weights in the SPH method of section 2.5), an imposed weighted graph (as in MLS-type reconstruction with imposed weights, see appendix D), or even a mesh (as for the Gauss operators of section 2.7.1).

Proposition 3.1 (Representation of node indiscernible volume functions)**.**
A volume function V satisfies node indiscernibility if and only if there exists a function $W : \mathcal{P}(\mathbb{R}^d) \times \mathbb{N} \times (\mathcal{C} \to \mathbb{R}^d) \to \mathbb{R}$ symmetric in its last $n-1$ variables such that: $\forall\ i \in \mathcal{C}$,

$$V_i(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) = W(\Omega, \mathcal{C}, \mathbf{x}_i, \underbrace{\mathbf{x}_1, \cdots, \mathbf{x}_{n_n}}_{\mathbf{x}_i \text{ is missing}}) \tag{3.3}$$

In this case, the function W is unique.

Proof. If such a function exists, choosing $i = 1$ and $\pi = id$ in expression (3.3) yields: $\forall\ (\mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$,

$$W(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) = V_1(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \tag{3.4}$$

This necessary form proves uniqueness. Additionally, $\forall\ \pi \in Sym(\mathcal{C})$, $\forall\ i \in \mathcal{C}$,

$$\begin{aligned} V_i(\Omega, \mathcal{C}, \mathbf{x}_{\pi(1)}, \cdots, \mathbf{x}_{\pi(n)}) &= W(\Omega, \mathcal{C}, \mathbf{x}_{\pi(i)}, \mathbf{x}_{\pi(1)}, \cdots, \mathbf{x}_{\pi(n)}) \\ &= W(\Omega, \mathcal{C}, \mathbf{x}_{\pi(i)}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \\ &= V_{\pi(i)}(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \end{aligned}$$

This proves that V is invariant with respect to node re-labeling. Conversely, if V is invariant, let us define W as in equation (3.4). Then, $\forall\ \pi \in Sym(\mathcal{C}) \mid \pi(1) = 1$:

$$\begin{aligned} W(\Omega, \mathcal{C}, \mathbf{x}_1, \mathbf{x}_{\pi(2)}, \cdots, \mathbf{x}_{\pi(n)}) &= V_1(\Omega, \mathcal{C}, \mathbf{x}_1, \mathbf{x}_{\pi(2)}, \cdots, \mathbf{x}_{\pi(n)}) \\ &= V_1(\Omega, \mathcal{C}, \mathbf{x}_{\pi(1)}, \cdots, \mathbf{x}_{\pi(n)}) \\ &= V_{\pi(1)}(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \\ &= V_1(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \\ &= W(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \end{aligned}$$

Hence, W has the desired symmetry property. Moreover,

$$\begin{aligned} V_i(\Omega, \mathcal{C}, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) &= V_1(\Omega, \mathcal{C}, \mathbf{x}_i, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \\ &= W(\Omega, \mathcal{C}, \mathbf{x}_i, \mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \end{aligned}$$

Which concludes the proof. □

3.1.1 Primal gradient: from node movement arises an operator

In the rest of this section, we consider the restricted case where the volume weights are functions of nodes position only. We explain in section 3.2.1 how this special case fits into the general picture. Hence, the number of nodes n_n is fixed and the volume integration operator reads: $\forall\ f : \mathcal{C} \to \mathbb{R}$:

$$\oint\!\!\!\!\!\!\int_{\mathcal{C}} f = \sum_{i \in \mathcal{C}} V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) f_i \tag{3.5}$$

The definition of the volume-based gradient operator is inspired by the following continuous transport theorem:

Definition 3.3 (A transport theorem)**.**
Let us consider that the integration domain Ω is advected (transported) by a regular vector field $\mathbf{v} : \Omega \to \mathbb{R}^d$. Then, for any regular enough time-dependent scalar field $f : \mathbb{R} \times \Omega \to \mathbb{R}$, we have:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega f \ \mathrm{d}V = \int_\Omega \frac{\mathrm{d}f}{\mathrm{d}t} + f \nabla \cdot \mathbf{v} \ \mathrm{d}V \tag{3.6}$$

This formula shows that two factors influence the variation of the integral of f over Ω:

- $\dfrac{\mathrm{d}f}{\mathrm{d}t}$: The point-wise evaluation variation of the scalar field on advected points of Ω.

- $f\nabla \cdot \mathbf{v}$: A volumetric term accounting for stretching or compression of neighborhoods of points in Ω, expressed as the divergence of the advection vector field.

The following definition transposes the above transport theorem to the discrete setting, and identifies the discrete divergence of a vector field as the variation of discrete volume during the advection process.

Definition 3.4 (Volume-based gradient operator)**.**
In order to transpose expression (3.6) to the discrete realm, we consider time-dependent discrete fields $f : \mathbb{R} \times \mathcal{C} \to \mathbb{R}$ and $\mathbf{v} : \mathbb{R} \times \mathcal{C} \to \mathbb{R}$. Nodes of the point cloud are advected by $\mathbf{v}$: $\forall\, t \in \mathbb{R}$, $\forall\, i \in \mathcal{C}$, $\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_i(t)$. We now define the volume-based gradient as:

$$\frac{\mathrm{d}}{\mathrm{d}t} \fint_{\mathcal{C}} f \overset{\text{def}}{=} \fint_{\mathcal{C}} \frac{\mathrm{d}f}{\mathrm{d}t} + f\mathbb{\nabla} \cdot \mathbf{v} \tag{3.7}$$

Which amounts to writing: $\forall\, \mathbf{v} : \mathcal{C} \to \mathbb{R}^d$,

$$V_i \mathbb{\nabla}_i \cdot \mathbf{v} \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \frac{\partial V_i}{\partial \mathbf{x}_j} \cdot \mathbf{v}_j \tag{3.8}$$

In terms of the gradient coordinates $\mathbf{A}_{i,j}$ (see definition 2.9), the above definition reads:

$$\mathbf{A}_{i,j} \overset{\text{def}}{=} \frac{\partial V_i}{\partial \mathbf{x}_j} \tag{3.9}$$

Proof. With the notations introduced above, we have:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \fint_{\mathcal{C}} f &= \frac{\mathrm{d}}{\mathrm{d}t} \sum_{i \in \mathcal{C}} V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) f_i \\
&= \sum_{i \in \mathcal{C}} V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) \frac{\mathrm{d}f_i}{\mathrm{d}t} + f_i \frac{\mathrm{d}}{\mathrm{d}t} V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) \\
&= \sum_{i \in \mathcal{C}} V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) \frac{\mathrm{d}f_i}{\mathrm{d}t} + f_i \sum_{j \in \mathcal{C}} \underbrace{\frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n})}_{\overset{\text{def}}{=} \mathbf{A}_{i,j}} \cdot \frac{\mathrm{d}\mathbf{x}_j}{\mathrm{d}t} \\
&= \fint_{\mathcal{C}} \frac{\mathrm{d}f}{\mathrm{d}t} + f\mathbb{\nabla} \cdot \mathbf{v}
\end{aligned}$$

□

Remark. Expression (3.9) allows us to interpret the graph of $\mathbb{\nabla}$: the edge (i, j) is present in this graph if $\frac{\partial V_i}{\partial \mathbf{x}_j} \neq \mathbf{0}$, *i.e.* if the volume of node i locally depends on the position of node j.

Proposition 3.2 (Representation of node indiscernible volume-based gradients)**.**
Node indiscernibility (definition 3.2) translates to the volume-based gradient as: $\forall\, \pi \in Sym(\mathcal{C})$, $\forall\, (\mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) \in (\mathcal{C} \to \mathbb{R}^d)$, $\forall\, i, j \in \mathcal{C}$:

$$\frac{\partial V_{\pi(i)}}{\partial \mathbf{x}_{\pi(j)}}(\mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) = \frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_{\pi(1)}, \cdots, \mathbf{x}_{\pi(n_n)}) \tag{3.10}$$

Moreover, it entails that there exists a unique function denoted $\mathbf{Z} : (\llbracket 1, n_n \rrbracket \to \mathbb{R}^d) \to \mathbb{R}^d$ (which is not the gradient of the function W introduced in (3.3)) symmetric in its last $n-2$ variables such that $\forall\ i \neq j \in \mathcal{C}$,

$$\frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_1, \cdots, \mathbf{x}_{n_n}) = \mathbf{Z}(\mathbf{x}_i, \mathbf{x}_j, \underbrace{\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}}_{\mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are missing}}) \tag{3.11}$$

Proof. Using proposition 3.1, we can write: $\forall\ \mathbf{a} \in \mathbb{R}^d$, $\forall\ i, j \in \mathcal{C}$,

$$\begin{aligned}
\frac{\partial V_{\pi(i)}}{\partial \mathbf{x}_{\pi(j)}} \cdot \mathbf{a} &= \frac{\mathrm{d}}{\mathrm{d}t} V_{\pi(i)}(\mathbf{x}_1, \ldots, \underbrace{\mathbf{x}_{\pi(j)} + t\mathbf{a}}_{\text{position } \pi(j)}, \ldots, \mathbf{x}_{n_n})\Big|_{t=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t} V_i(\mathbf{x}_{\pi(1)}, \ldots, \underbrace{\mathbf{x}_{\pi(j)} + t\mathbf{a}}_{\text{position } j}, \ldots, \mathbf{x}_{\pi(n_n)})\Big|_{t=0} \\
&= \frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(n_n)}) \cdot \mathbf{a}
\end{aligned}$$

The rest of the proof is identical to that of proposition 3.1. □

Remark. Getting rid of the scalar field f, and only retaining the advection vector field $\mathbf{v} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}$, expression (3.7) can equivalently be written as: $\forall\ i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}_i \cdot \mathbf{v} = \frac{\mathrm{d}V_i}{\mathrm{d}t} \tag{3.12}$$

Under this form, the volume-based divergence operator is completely identical to those given in the literature (expression (3.5) of [158] and expression (14) of [87]). Moreover, expression (3.12) is obviously intimately linked with the kinematics of the point cloud, namely the advection of discretization nodes: the value of the primal divergence of a vector field $\mathbf{v}$ evaluated at node i depends on the local variation of a single scalar volume weight V_i during the collective movement of neighboring nodes following the vector field. Of course, this displacement is virtual: no discretization node needs to actually move to compute $V_i \mathbb{\nabla} \cdot \mathbf{v}$.

In order to make the above remark more precise for continuous fields, we need the definition of the flow of a vector field (also see chapter 4 of [2]):

Definition 3.5 (Flow of a vector field)**.**
Let $\mathbf{v} : \mathbb{R}^d \to \mathbb{R}$ be a smooth enough vector field (C^1 regularity is sufficient). Then there exist an application $\mathrm{Fl}^{\mathbf{v}} : \mathbb{I} \times \mathbb{R}^d \to \mathbb{R}^d$ with $0 \in \mathring{\mathbb{I}} \subset \mathbb{R}$ called the flow of $\mathbf{v}$ such that $\forall\ \mathbf{x} \in \mathbb{R}^d$, $\forall\ \varepsilon \in \mathring{\mathbb{I}}$,

$$\begin{cases} \mathrm{Fl}^{\mathbf{v}}_0(\mathbf{x}) = \mathbf{x} \\ \dfrac{\mathrm{d}}{\mathrm{d}\varepsilon} \mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}) = \mathbf{v}(\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x})) \end{cases} \tag{3.13}$$

Corollary 3.3 (Volume-based gradient in terms of flow)**.**
The discrete divergence of the continuous field $\mathbf{v}$ can be written in terms of its flow as: $\forall\ i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}_i \cdot \mathbf{v} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} V_i(\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_1), \ldots, \mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_{n_n}))\Big|_{\varepsilon=0} \tag{3.14}$$

Of course, the above equation is the continuous counterpart to expression (3.12).

Proof. The Right Hand Side (RHS) of expression (3.14) can be re-written as:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\varepsilon}V_i(\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_1),\dots,\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_{n_n}))\Big|_{\varepsilon=0} &= \sum_{j\in\mathcal{C}}\frac{\partial V_i}{\partial \mathbf{x}_j}\cdot\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_j)\Big|_{\varepsilon=0}\\ &= \sum_{j\in\mathcal{C}}\frac{\partial V_i}{\partial \mathbf{x}_j}\cdot\mathbf{v}(\mathrm{Fl}^{\mathbf{v}}_0(\mathbf{x}_j))\\ &= \sum_{j\in\mathcal{C}}\frac{\partial V_i}{\partial \mathbf{x}_j}\cdot\mathbf{v}(\mathbf{x}_j)\\ &= V_i\mathbb{\nabla}\cdot\mathbf{v}\end{aligned}$$

□

Hence, $\mathbb{\nabla}$ is related to the evolution of the volume integration operator $\oiint_{\mathcal{C}}$ under the flow transformation.

3.1.2 Local error of volume-based primal gradients

We have seen in the previous section that the volume-based gradient operator reflects the evolution of the volume integration operator as the domain of integration is (infinitesimally) advected. In the continuous world, the change of variable formula (see chapter 8, and especially theorem 8.1.2 of [2]) can be used to express the updated integral in terms of the original non-advected space:

Proposition 3.4 (Change of variable formula)**.**
Let Ω be an open subset of $\mathbb{R}^d$ and $\varphi:\Omega\to\varphi(\Omega)$ be an orientation-preserving diffeomorphism with Jacobian $\mathrm{J}_\varphi=\frac{\partial\varphi}{\partial\mathbf{x}}$. Then, for all integrable scalar field $f:\Omega\to\mathbb{R}$, we have:

$$\int_{\mathbf{y}\in\varphi(\Omega)} f(\mathbf{y})\ \mathrm{d}V=\int_{\mathbf{x}\in\Omega} f\circ\varphi(\mathbf{x})\det(J_\varphi)(\mathbf{x})\ \mathrm{d}V \tag{3.15}$$

The flow $\mathrm{Fl}^{\mathbf{v}}$ of a continuous vector field $\mathbf{v}:\Omega\to\mathbb{R}^d$ provides infinitely many diffeomorphisms mapping the "reference" computational domain Ω to its advected counterparts $\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\Omega)\ \forall\ \varepsilon\in\mathbb{I}$ and the change of variable formula is applicable for each of these transformations (left side of figure 3.1). In the discrete world however, the change of variable formula does not hold in general (right side of figure 3.1).

Proposition 3.5 shows that the amount by which the change of variable formula fails at the discrete level for small ε is precisely the error on the primal divergence of $\mathbf{v}$. In other words, the consistency error of the primal differential operator makes the diagram in figure 3.1 non-commutative.

Proposition 3.5 (Integral expression of the error for volume-based gradients)**.**
Let $f:\Omega\to\mathbb{R}$ $\mathbf{v}:\Omega\to\mathbb{R}^d$ be regular enough continuous fields. Then, the following formula holds:

$$\oiint_{\mathcal{C}} f\,(\mathbb{\nabla}-\nabla)\cdot\mathbf{v}=\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\oiint_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathcal{C})} f\ -\ \oiint_{\mathcal{C}} f\circ\mathrm{Fl}^{\mathbf{v}}_\varepsilon\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon})\right)\Bigg|_{\varepsilon=0} \tag{3.16}$$

This integrated form of the error of the primal differentiation operator is *local* in the sense that it only involves one particular set of node positions $(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})$.

Proof. The proof is nothing but a direct computation. The RHS of (3.16) reads:

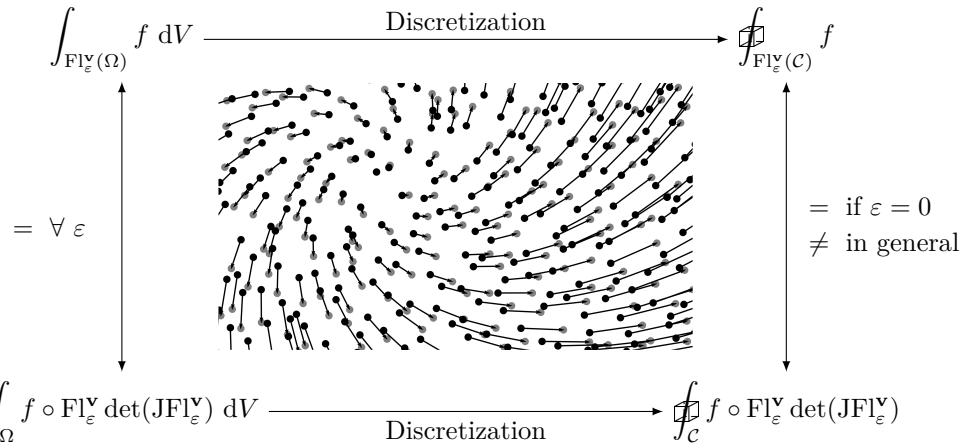


Figure 3.1 – *The change of variable formula is in general not valid at the discrete level. The picture at the center represents a cloud of points advected by a smooth vector field. On the left, the continuous change of variable formula ensures that the top and bottom expressions are equal, but such a result does not hold for their discrete equivalent on the right. The associated error is directly linked to the consistency error of the primal gradient as stated in proposition 3.5.*

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\oint_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathcal{C})} f \;-\; \oint_{\mathcal{C}} f \circ \mathrm{Fl}^{\mathbf{v}}_{\varepsilon} \det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}})\right)\Bigg|_{\varepsilon=0} \\
&= \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\sum_{i\in\mathcal{C}} V_i(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_1),\dots,\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_{n_n}))f(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_i)) - V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})f(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_i))\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}})(\mathbf{x}_i)\right)\Bigg|_{\varepsilon=0} \\
&= \sum_{i\in\mathcal{C}} \frac{\mathrm{d}}{\mathrm{d}\varepsilon}(V_i(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_1),\dots,\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_{n_n})) - V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}})(\mathbf{x}_i))\Bigg|_{\varepsilon=0} f(\mathbf{x}_i) \\
&= \sum_{i\in\mathcal{C}}\sum_{j\in\mathcal{C}} \frac{\partial V_i}{\partial \mathbf{x}_j}(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_1),\dots,\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_{n_n}))\cdot\frac{\mathrm{dFl}^{\mathbf{v}}_{\varepsilon}}{\mathrm{d}\varepsilon}(\mathbf{x}_j) \\
&\qquad - V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}})\,\mathrm{Tr}\left(\mathrm{J}^{-1}_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}}\frac{\mathrm{dJ}_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}}}{\mathrm{d}\varepsilon}\right)(\mathbf{x}_i)\Bigg|_{\varepsilon=0} f(\mathbf{x}_i) \\
&= \sum_{i\in\mathcal{C}}\left(\left(\sum_{j\in\mathcal{C}}\frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})\cdot\mathbf{u}(\mathbf{x}_j)\right) - V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})\nabla\cdot\mathbf{u}(\mathbf{x}_i)\right) f(\mathbf{x}_i) \\
&= \sum_{i\in\mathcal{C}} V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})f(\mathbf{x}_i)(\mathbb{\nabla}_i\mathbf{u} - \nabla\cdot\mathbf{u}(\mathbf{x}_i)) \\
&= \oint_{\mathcal{C}} f\,(\mathbb{\nabla} - \nabla)\cdot\mathbf{u}
\end{aligned}
$$

□

Much like the integral definition 3.4 can be reformulated in point-wise form as corollary 3.3, the integral error of proposition 3.5 is expressed in point-wise form below:

Corollary 3.6 (Pointwise expression of the error of the primal gradient)**.**
Getting rid of the scalar field f, and only retaining the advection vector field $\mathbf{v}$, we can derive the following point-wise expression of the error: $\forall\, i \in \mathcal{C}$,

$$V_i(\mathbb{\nabla}_i \cdot \mathbf{v} - \nabla \cdot \mathbf{v}(\mathbf{x}_i)) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \frac{V_i(\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_1), \dots, \mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_{n_n}))}{\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon}(\mathbf{x}_i))}\bigg|_{\varepsilon=0} \tag{3.17}$$

Proof. The proof is again nothing but a direct computation. The RHS of expression (3.17) reads:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\varepsilon} \frac{V_i(\mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_1), \dots, \mathrm{Fl}^{\mathbf{v}}_\varepsilon(\mathbf{x}_{n_n}))}{\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon}(\mathbf{x}_i))}\bigg|_{\varepsilon=0} \\
&\quad = \frac{1}{\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon}(\mathbf{x}_i))} \sum_{j \in \mathcal{C}} \frac{\partial V_i}{\partial \mathbf{x}_j} \cdot \frac{\mathrm{d}\mathrm{Fl}^{\mathbf{v}}_\varepsilon}{\mathrm{d}\varepsilon}\bigg|_{\varepsilon=0} - V_i \frac{1}{\det(\mathrm{J}^2_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon}(\mathbf{x}_i))} \mathrm{Tr}\left(\mathrm{J}^{-1}_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon} \frac{\mathrm{d}\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_\varepsilon}}{\mathrm{d}\varepsilon}\right)(\mathbf{x}_i)\bigg|_{\varepsilon=0} \\
&\quad = V_i(\mathbb{\nabla}_i \cdot \mathbf{v} - \nabla \cdot \mathbf{v}(\mathbf{x}_i))
\end{aligned}$$

□

Remark (Structure of the kernel of the error on the primal gradient)**.**
Figure 3.1 is similar to figure 2.3, where the operator Op is the flow $\mathrm{Fl}^{\mathbf{v}}_\varepsilon$. The integral (proposition 3.5) and pointwise (corollary 3.6) expressions of the error show that covariance of the discrete volume integral with respect to small transformations of space entails exactness on the infinitesimal generator.

More generally, this remains true if the flow is replaced with any Lie subgroup of diffeomorphisms of $\mathbb{R}^d$: failure of the change of variable formula on a Lie group of space transformations G directly translates to an error of the primal divergence of the vector fields of the corresponding Lie algebra $\mathfrak{g}$ *i.e.* the tangent space to the Lie group at the identity transformation. As a consequence, if $\mathbb{\nabla}$ is exact on the vector fields $\mathbf{u}$ and $\mathbf{v}$, then it is exact on their Lie bracket $[\mathbf{u}, \mathbf{v}]$.

Example (Expression of the error of volume-based gradients on constant and linear fields)**.**

Applying corollary 3.6 to the Lie group of uniform translations (resp. affine transformations) yield the following local expression of the error of the volume-based discrete gradient on constant (resp. linear) fields:

$$V_i \mathbb{\nabla}_i 1 = \frac{\partial}{\partial \mathbf{u}} V_i(\mathbf{x}_1 + \mathbf{u}, \dots, \mathbf{x}_{n_n} + \mathbf{u})\bigg|_{\mathbf{u}=0} \tag{3.18}$$

$$V_i(\mathbb{\nabla}_i \mathbf{x} - \mathrm{I}_d) = \frac{\partial}{\partial \mathrm{B}} \frac{1}{\det(\mathrm{B})} V_i(\mathrm{B}\mathbf{x}_1, \dots, \mathrm{B}\mathbf{x}_{n_n})\bigg|_{\mathrm{B}=\mathrm{I}_d} \tag{3.19}$$

3.1.3 Global error of volume-based primal gradients

Corollary 3.6 can be refined into a *global* set of requirement, which characterizes the exactness of the primal gradient *for all* node positions:

Proposition 3.7 (Global consistency conditions of volume-based gradients)**.**
Let $\mathrm{Fl}^{\mathbf{v}} : \mathbb{I} \times \Omega \to \mathbb{R}^d$ denote the flow of a regular vector field $\mathbf{v} : \mathbb{R}^d \to \mathbb{R}^d$ and i a node of the point cloud $(\mathcal{C}, (\mathbf{x}_1, \dots, \mathbf{x}_{n_n}))$. The following statements are equivalent:

- The discrete primal operator is exact on $\mathbf{v}$ for any nodal positions: $\forall\, (\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$,

$$V_i \mathbb{\nabla}_i \cdot \mathbf{v} = V_i \nabla \cdot \mathbf{v}(\mathbf{x}_i) \tag{3.20}$$

- The nodal volume weights are proportional to the nodal evaluation of the determinant of the Jacobian of the corresponding transformation.
  $\forall\ (\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d, \forall\ \varepsilon \in \mathbb{R},$

$$V_i(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_1), \dots, \mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_{n_n})) = \det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}}(\mathbf{x}_i)) V_i(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) \tag{3.21}$$

Proof. Let us prove separately that each statement implies the other:
• Expression (3.21) implies expression (3.20): This is a direct consequence of corollary 3.6: differentiate expression (3.21) with respect to ε and evaluate at $\varepsilon = 0$, which gives expression (3.20).
• Expression (3.20) implies expression (3.21): Let $\varepsilon \in \mathbb{I}$. Since expression (3.20) is supposed to hold for any node position, let us apply it at $(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_1), \dots, \mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_{n_n}))$. From corollary 3.6, the following quantity vanishes:

$$V_i(\mathbb{\nabla}_i \cdot \mathbf{v} - \nabla \cdot \mathbf{v}(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_i))) = \frac{\mathrm{d}}{\mathrm{d}\mu} \frac{V_i(\mathrm{Fl}^{\mathbf{v}}_{\mu}(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_1)), \dots, \mathrm{Fl}^{\mathbf{v}}_{\mu}(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_{n_n})))}{\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_{\mu}}(\mathrm{Fl}^{\mathbf{v}}_{\varepsilon}(\mathbf{x}_i)))}\Bigg|_{\mu=0} \tag{3.22}$$

From Lie group theory and elementary properties of the exponential map, the flow operator defines a group morphism between $(\mathbb{R}, +)$ and transformations of $\mathbb{R}^d$. In particular, $\mathrm{Fl}^{\mathbf{v}}_{\mu} \circ \mathrm{Fl}^{\mathbf{v}}_{\varepsilon} = \mathrm{Fl}^{\mathbf{v}}_{\mu+\varepsilon}$ and consequently, $\frac{\mathrm{d}}{\mathrm{d}\mu} \frac{V_i(\mathrm{Fl}^{\mathbf{v}}_{\mu}(\mathbf{x}_1), \dots, \mathrm{Fl}^{\mathbf{v}}_{\mu}(\mathbf{x}_{n_n}))}{\det(\mathrm{J}_{\mathrm{Fl}^{\mathbf{v}}_{\mu}}(\mathbf{x}_i))}\Big|_{\mu=\varepsilon} = 0$. This being true for all $\varepsilon \in \mathbb{I}$, the quantity is constant with respect to ε and evaluation at $\varepsilon = 0$, yields expression (3.21). □

Remark. Similarly to its local version, this global theorem remains true when the flow one-parameter subgroup is substituted for any Lie subgroup of the diffeomorphisms on $\mathbb{R}^d$: evolution of volume weights proportionally to the pointwise values of the determinant of the Jacobian matrix of the transformation entails exactness of the primal gradient on the vector fields of the corresponding Lie algebra (adaptation of the above proof is straightforward). Conversely, exactness on a given Lie algebra of vector fields implies proportional evolution of volume weights and pointwise values of the determinant of the Jacobian matrix of the transformations of the (unique) simply connected associated Lie group (this is an immediate consequence of corollary 3.6). This result effectively endows the space of discrete functions on which the primal gradient is exact with a Lie algebra structure, extending the previous formulae given in the literature (equations (8-9) of [206] for instance).

Example (Characterization of global consistency of order zero and one)**.**
Necessary and sufficient conditions for global zeroth and first order consistency at node $i \in \mathcal{C}$ of the primal differentiation operator can be written: $\forall\ (\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d, \forall\ \mathbf{a} \in \mathbb{R}^d, \forall\ \mathrm{B} \in \mathrm{GL}^+_d(\mathbb{R})$,

$$V_i(\mathbf{x}_1 + \mathbf{a}, \dots, \mathbf{x}_{n_n} + \mathbf{a}) = V_i(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) \tag{3.23}$$

$$V_i(\mathrm{B}\mathbf{x}_1, \dots, \mathrm{B}\mathbf{x}_{n_n}) = \det(\mathrm{B}) V_i(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) \tag{3.24}$$

Expressions (3.23) and (3.24) are the global counterparts of expressions (3.18) and (3.19). As a consequence, choosing $\mathbf{a} = -\mathbf{x}_i$ in expression (3.23) proves that the volume weight V_i yields a zeroth order consistent primal gradient $V_i \mathbb{\nabla}_i$ if and only if there exists a function U_i such that V_i reads: $\forall\ \mathbf{X} : \mathcal{C} \to \mathbb{R}^d$,

$$V_i(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) = U_i(\mathbf{x}_1 - \mathbf{x}_i, \dots, \mathbf{x}_{n_n} - \mathbf{x}_i) \tag{3.25}$$

Example (Covariance with respect to rigid motion)**.**
Applying proposition 3.7 to the Lie group of isometric transformations (also known as rigid

motions) of space is a bit more subtle. This group is isomorphic to $\mathbb{R}^d \times \mathrm{O}_d(\mathbb{R})$, and its Lie algebra at the identity is isomorphic to $\mathbb{R}^d \times \mathrm{A}_d(\mathbb{R})$. Hence, covariance of the volume function with respect to translations and rotations is equivalent to zeroth order consistency and vanishing of the skew-symmetric part of $\mathbb{V}_i\mathbf{x}$ (or equivalently, the symmetry of the matrix $\mathbb{V}_i\mathbf{x}$). In particular if $d = 3$, this is equivalent to the fact that the curl of the coordinate field vanishes: $\mathbb{V}_i \times \mathbf{x} = \mathbf{0}$. We use this property in appendix G, where it is equivalent to conservation of linear and angular momenta.

Volume functions that are invariant with respect to rigid motions are those representable under the form given in expression (3.25), where U_i is an isotropic scalar valued function. As a consequence of Cauchy's representation formula for isotropic scalar valued functions (see [209, 228] for more information about representation theorems for isotropic functions), rigid motion invariant volume weights are expressible only in terms of scalar products between shifted positions: $\forall\ \mathbf{X} : \mathcal{C} \to \mathbb{R}^d$,

$$V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) = U_i\left(((\mathbf{x}_j - \mathbf{x}_i) \cdot (\mathbf{x}_k - \mathbf{x}_i))_{j,k\in\mathcal{C}}\right) \tag{3.26}$$

3.1.4 The price of linear consistency of volume-based primal gradients

This section is devoted to the construction of a counter-example showing that under quite restrictive conditions, it is not possible to choose a volume function yielding a globally first order consistent volume-based gradient.

Conjecture 3.8 (Local primal volume gradients are not first order consistent)**.**
There is no volume function V satisfying:

- Continuity with respect to node positions.
- Locality of the dependency with respect to node positions.
- Global affine covariance with respect to node positions: $\forall\ (\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$, $\forall\ \mathbf{a} \in \mathbb{R}^d$, $\forall\ \mathrm{B} \in \mathrm{M}_d(\mathbb{R})$,
- Global affine covariance with respect to node positions: $\forall\ (\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$, $\forall\ \mathbf{a} \in \mathbb{R}^d$, $\forall\ \mathrm{B} \in \mathrm{M}_d(\mathbb{R})$,

  $$V_i(\mathrm{B}\mathbf{x}_1 + \mathbf{a}, \ldots, \mathrm{B}\mathbf{x}_{n_n} + \mathbf{a}) = \det(\mathrm{B}) V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) \tag{3.27}$$

  This requirement is equivalent to first order consistency of the corresponding volume-based gradient (see proposition 3.7).
- The volume function is not constant with respect to node positions, or equivalently, the corresponding volume-based gradient is non-vanishing.

Even though we believe conjecture 3.8 to hold, maybe under slightly more restrictive assumptions, the technical aspects of the proof only lets us formulate the following proposition:

Proposition 3.9 (Local primal volume gradients are not first order consistent)**.**
Under the following assumptions:

- The space dimension d is larger that 2.
- The volume function is C^1-continuous with respect to node positions and satisfies node indiscernibility (definition 3.2).
- V is local in the following sense: there exists a symmetric smoothing length function $h : (\mathcal{C} \to \mathbb{R}^d) \to \mathbb{R}^+$ such that: $\forall\ (\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$,

  $$\forall\ i, j \in \mathcal{C}, \quad (\|\mathbf{x}_j - \mathbf{x}_i\| > h(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n})) \Rightarrow \left(\frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) = \mathbf{0}\right) \tag{3.28}$$

- The corresponding volume-based gradient is globally consistent of order one: $\forall\ (\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d, \forall\ i \in \mathcal{C}$,

$$\begin{cases} V_i \mathbb{\nabla}_i 1 & = \mathbf{0} \\ V_i (\mathbb{\nabla}_i \mathbf{x} - \mathrm{I}_d) & = 0 \end{cases}$$

we have, $\forall\ i \in \mathcal{C},\ \forall\ f : \mathcal{C} \to \mathbb{R}$:

$$V_i \mathbb{\nabla}_i f = \sum_{j \in \mathcal{C}} \frac{\partial V_i}{\partial \mathbf{x}_j} f_j = 0 \tag{3.29}$$

In other words, volume weights do not vary with node positions: they are constants.

Remark. Before detailing the proof, let us first discuss the assumptions of proposition 3.9:

- The dimension of ambient space is taken to be at least two. This assumption is purely technical and not very restrictive as the case $d = 1$ is of little practical interest. Actually, the special topological properties of $\mathbb{R}$ imply that point clouds in one dimension actually define meshes, limiting the validity of the term "meshless method". Moreover, these very topological properties prevent nodes from "mixing while not crossing", rendering our proof invalid.
- The property is formulated for volume functions satisfying node indiscernibility, but is believed to hold even without this assumption.
- Locality as defined above only really makes sense for infinite point clouds (as in the proof). Indeed, if $\mathcal{C}$ is finite, the smoothing length can be chosen as $h = \max_{i,j \in \mathcal{C}} 1 + \|x_j - x_i\|$ rendering every volume function "local". Practical computing time considerations obviously prohibit using the corresponding differential operator for simulation purposes, and another notion of locality is needed. Moreover, considering the discretization of $\mathbb{R}^d$ allows us to circumvent the problem of the boundary (which we address in section 3.2).
- The locality assumption is essential. Indeed, appendix D.4 provides an example of non-local volume function yielding a first order consistent primal gradient.

In the author's opinion, this theorem is strong enough to forsake any hope of finding a globally defined volume function with associated primal gradient consistent of order $p \geq 1$ to be successfully used in a Lagrangian discretization context (see appendix G).

Proof of proposition 3.9. The proof is given for $d = 2$, but can straightforwardly be extended to higher space dimensions. We build a point cloud so that it remains invariant with respect to a set of affine transformations. From consistency at every node position, we deduce that every partial derivative of the volume function is zero, hence the result. Let $(\mathcal{C} = \mathbb{Z}^2, \mathbf{X})$ a countable infinite cloud of points positioned on the regular lattice of integer coordinate: $\forall\ i \in \mathcal{C},\ \mathbf{x}_i = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix}$. Let $i \neq j \in \mathcal{C}$. For all $p \in \mathbb{Z}$, let us consider the following affine transformations of $\mathbb{R}^2$:

$$\begin{cases} a_p^i(\mathbf{x}) = \mathbf{x}_i + \mathrm{A}_p(\mathbf{x} - \mathbf{x}_i) \\ b_p^i(\mathbf{x}) = \mathbf{x}_i + \mathrm{B}_p(\mathbf{x} - \mathbf{x}_i) \end{cases}$$

Of course, $a_p^i(\mathbf{x_i}) = b_p^i(\mathbf{x}_i) = \mathbf{x}_i$. Only integers appear in their definitions, thus a_p^i and b_p^i are affine homomorphisms. In particular, they map $\mathbb{Z}^2$ to a subset of $\mathbb{Z}^2$. Moreover, $\det(\mathrm{A}_p) = \det(\mathrm{B}_p) = 1$, which is invertible in $\mathbb{Z}$, proving that both a_p^i and b_p^i are affine

isomorphisms mapping $\mathbb{Z}^2$ to itself: the set of node positions remains unchanged and so does h.
Since we assumed primal consistency of order zero and one, we have:

$$V_i((\mathbf{x}_k)_{k\in\mathcal{C}}) = \frac{1}{\det(A_p)} V_i((a_p^i(\mathbf{x}_k))_{k\in\mathcal{C}}) = V_i((a_p^i(\mathbf{x}_k))_{k\in\mathcal{C}})$$
$$\Rightarrow \frac{\partial V_i}{\partial \mathbf{x}_j}((\mathbf{x}_k)_{k\in\mathcal{C}}) = \mathrm{A}_p^T \frac{\partial V_i}{\partial \mathbf{x}_j}((a_p^i(\mathbf{x}_k))_{k\in\mathcal{C}}) \tag{3.30}$$

And similarly for b_p^i. A quick computation gives:

$$a_p^i(\mathbf{x}_j) - \mathbf{x}_i = \begin{pmatrix} y_j - y_i \\ -(x_j - x_i) \end{pmatrix} + p \begin{pmatrix} x_j - x_i + y_j - y_i \\ x_j - x_i + y_j - y_i \end{pmatrix} \tag{3.31}$$

$$b_p^i(\mathbf{x}_j) - \mathbf{x}_i = \begin{pmatrix} -(y_j - y_j) \\ x_j - x_i \end{pmatrix} + p \begin{pmatrix} x_j - x_i - (y_j - y_i) \\ y_j - y_i - (x_j - x_i) \end{pmatrix} \tag{3.32}$$

Thus, both $\{a_p^i(\mathbf{x}_j) - \mathbf{x}_i\}_{p\in\mathbb{N}}$ and $\{b_p^i(\mathbf{x}_j) - \mathbf{x}_i\}_{p\in\mathbb{N}}$ are bounded if and only if:

$$\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x_j - x_i \\ y_j - y_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.33}$$

Whose only solution reads $\mathbf{x}_i = \mathbf{x}_j$. This is not the case (we assumed $i \neq j$). Hence $\exists\ p \in \mathbb{Z} \mid \|a_p^i(\mathbf{x}_j) - \mathbf{x}_i\| > h$ or $\|b_p^i(\mathbf{x}_j) - \mathbf{x}_i\| > h$. Because of our locality assumption, either $\frac{\partial V_i}{\partial \mathbf{x}_j}((a_p^i(\mathbf{x}_k))_{k\in\mathcal{C}}) = \mathbf{0}$ or $\frac{\partial V_i}{\partial \mathbf{x}_j}((b_p^i(\mathbf{x}_k))_{k\in\mathcal{C}}) = \mathbf{0}$, which proves that $\frac{\partial V_i}{\partial \mathbf{x}_j}((\mathbf{x}_k)_{k\in\mathcal{C}}) = \mathbf{0}$ in virtue of (3.30).
Then, $\mathbf{0} = V_i \mathbb{\nabla}_i 1 = \sum_{j\in\mathcal{C}} \frac{\partial V_i}{\partial \mathbf{x}_j}((\mathbf{x}_k)_{k\in\mathcal{C}}) = \frac{\partial V_i}{\partial \mathbf{x}_i}((\mathbf{x}_k)_{k\in\mathcal{C}})$. Thus, V_i is a constant function, proving the desired result. □

Remark (Maximality of rigid motion covariance)**.** Our proof actually shows a slightly stronger result: since all matrices considered in the proof have unit determinant, we in fact showed that the special linear group $\mathrm{SL}_n^+(\mathbb{R})$ cannot be a subgroup of the group of covariant transformation of the volume function. On the other hand, we give in sections 3.5, 3.7 and appendix D.4 and F several examples of volume functions which are covariant with respect to $\mathrm{O}_n^+(\mathbb{R}) \subset \mathrm{SL}_n(\mathbb{R})$, the group of isometric transformations.

A general result of Noll (see [176]) states that $\mathrm{O}_n^+(\mathbb{R})$ is a maximal subgroup of $\mathrm{SL}_n^+(\mathbb{R})$. Applied to our case, this means that it is pointless to expect exactness of primal volume-based gradients on a function space strictly larger than the space of skew-symmetric affine transformations, since such a function space would necessarily include the Lie algebra of volume-preserving affine transformation on which the primal gradient cannot be exact.

Remark (Compatibility of volume-based operators)**.**
The most important take-away from conjecture 3.8 and proposition 3.9 is that the consistency order of volume-based primal gradient cannot exceed zero. The search for a first order consistent compatible set of operators thus apparently ends here.

In section 3.2, we define a boundary integration operator from a volume function, and in section 3.3, we study the resulting dual gradient operator. In particular, we see that the volume-based dual gradient operator does not have a theoretical limit on its consistency order. Since, the labels "primal" and "dual" are completely arbitrary – what is really important is that the differentiation operators form a conjugate pair with respect to the integration operators –, we can for all intents and purposes invert the roles of the primal and dual

gradients. And indeed, we succeed in exhibiting sets of volume-based operators that are first order consistent and compatible in this sense, see section 3.7 and appendix F.

In chapter 4, where we present simulation results, we implicitly invert the roles of these two operators, keeping the highest consistency order for the primal gradient. Note that this also means that we transpose the boundary integration operator, inverting the roles of the two discrete fields.

3.2 Volumes depend on the computational domain: Definition of a boundary integration operator

In this section, we still consider the number of nodes n_n fixed, but we now explicitly write the dependency of the volume weights with respect to the computational domain Ω: $\forall\ f : \mathcal{C} \to \mathbb{R}$:

$$\int\!\!\!\!\!\!\diagup_{\mathcal{C}} f = \sum_{i \in \mathcal{C}} V_i(\Omega, \mathbf{x}_1, \dots, \mathbf{x}_{n_n}) f_i \tag{3.34}$$

Once again, let us clarify that the computational domain Ω is assumed given, and that we do not attempt to retrieve it from the point cloud. While expression (3.34) is clearly more general than expression (3.5), let us explain in this section why nothing from section 3.1 is invalidated.

In order to fall back to the special case where the volume function only depends on node positions, let us suppose for a moment that Ω is itself a function of node positions: $\Omega(\mathbf{x}_1, \dots, \mathbf{x}_{n_n})$. If the nodes are advected as before with the discrete field $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}$, we should be able to make sense of the following chain rule to compute the discrete divergence of $\mathbf{v}$. Starting from expression (3.12), we have: $\forall\ i \in \mathcal{C}$,

$$\begin{aligned} V_i \mathbb{\nabla} \cdot \mathbf{v} &= \frac{\mathrm{d}}{\mathrm{d}t} V_i(\Omega(\mathbf{x}_1(t), \dots, \mathbf{x}_{n_n}(t)), \mathbf{x}_1(t), \dots, \mathbf{x}_{n_n}(t)) \\ &= \sum_{j \in \mathcal{C}} \frac{\mathrm{d}V_i}{\mathrm{d}\mathbf{x}_j}(\Omega(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}), \mathbf{x}_1, \dots, \mathbf{x}_{n_n}) \cdot \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} \\ &= \sum_{j \in \mathcal{C}} \left(\frac{\partial V_i}{\partial \Omega} \cdot \frac{\partial \Omega}{\partial \mathbf{x}_j} + \frac{\partial V_i}{\partial \mathbf{x}_j} \right) \cdot \mathbf{v}_j \end{aligned} \tag{3.35}$$

In section 3.2.2, we show that the term $\frac{\partial V_i}{\partial \Omega} \cdot \frac{\partial \Omega}{\partial \mathbf{x}_j}$ should be understood as the coordinates $\mathbf{\Gamma}_{i,j}$ of a volume-based boundary integration operator and in section 3.3, we interpret the term $\frac{\partial V_i}{\partial \mathbf{x}_j}$ as the coordinates $\mathbf{A}^*_{i,j}$ of the (opposite of the) resulting volume-based dual gradient operator.

3.2.1 Separating the domain dependency from pure node movement

While instructive, the assumption that Ω is a function of the positions of the discretization nodes is flawed. It would mean that we have a procedure to build the underlying computational domain Ω from a point cloud $(\mathcal{C}, \mathbf{X})$. We do not make this very strong assumption in this work. Instead, we replace it with the following strictly weaker one: we suppose given a procedure to define how points of the computational domain $\mathbf{x} \in \Omega$ are dragged around as the nodes of the point cloud move. In other words, we declare that that the movement of space points is driven by the movement of discretization nodes, which amounts to defining the continuous equivalent $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = < \mathbf{v} > (\mathbf{x})$ to the discrete field $\frac{\mathrm{d}\mathbf{x}_j}{\mathrm{d}t} = \mathbf{v}_j$

as a *linear combination*. This is tantamount to assuming the existence of a reconstruction map (definition 2.10) for vector fields.

This representation of dependency is called "transformations generated by velocities" by Delfour and Zolésio. This choice is not the only possible one, and the work of Delfour and Zolésio [70] provides an in-depth exposition of the topic. In this work, we always assume enough regularity of motion for the main results to hold.

More precisely, we assume given some kind of matrix shape functions $\Psi_j : \Omega \times (\mathcal{C} \to \mathbb{R}^d) \to \mathrm{M}_d(\mathbb{R})$ and write:

$$< \mathbf{v} > (\mathbf{x}) \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \Psi_j^T(\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) \cdot \mathbf{v}_j \tag{3.36}$$

As Ψ_j stands for the local evolution of nodes of Ω with respect to variations of the discretization nodes $\mathbf{x}_j$, it makes sense to denote $\Psi_j = \frac{\partial \Omega}{\partial \mathbf{x}_j}$.

Remark. Once the $(\Psi_j)_{j \in \mathcal{C}}$ are given, the volume function can really locally be considered as a function of the node positions only. Notation-wise, the corresponding variation is denoted with a total derivative. The results of section 3.1 still hold if we substitute the partial derivatives in definition 3.1 with full derivatives as:

$$V_i \mathbb{V}_i \cdot \mathbf{v} \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \frac{\mathrm{d}V_i}{\mathrm{d}\mathbf{x}_j} \cdot \mathbf{v}_j \tag{3.37}$$

Remark. In most examples, the definition of the shape functions $(\Psi_j)_{j \in \mathcal{C}}$ is either implicit, or there is an obvious good choice available.

The definition of the shape functions $(\Psi_j)_{j \in \mathcal{C}}$ bridges the gap between the definition of volume functions in terms of node coordinates only (section 3.1) and the introduction of a dependency with respect to the computational domain. This link is only assumed to hold locally, in a tangent form: the movement of points can be deduced from the movement of points through the shape functions $(\Psi_j)_{j \in \mathcal{C}}$, but we do not assume Ω itself to be retrievable from the point cloud. This still allows the decomposition of the primal volume-based gradient into a boundary integration operator and a dual gradient operator as we explain in the following sections.

3.2.2 The structure theorem and its consequences

Once the dependency of the discretization domain with respect to discretization nodes is specified, we still need to compute the sensitivity of the volume function with respect to the computational domain. To this end, we need a generalized form of the Reynolds transport theorem called the "structure theorem". See chapter 9 of [70], and especially theorem 3.6 for a more detailed exposition of the topic.

The structure theorem 3.1.
Let $f : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ be a regular real-valued "shape functional", *i.e.* a function whose argument is a domain $\Omega \subset \mathbb{R}^d$. Then, the support of the gradient of f at Ω is contained in $\partial\Omega$. In other words, for all regular enough $\Omega \subset \mathbb{R}^d$, there exists a scalar boundary field $\phi : \partial\Omega \to \mathbb{R}$ such that the variation of f as Ω is advected by the flow (see definition 3.5) of a continuous vector field $\mathbf{v} : \mathbb{R}^d \to \mathbb{R}^d$ reads:

$$\begin{aligned} \frac{\partial f}{\partial \Omega} \cdot \mathbf{v} &\overset{\text{def}}{=} \frac{\mathrm{d}}{\mathrm{d}\varepsilon} f(\mathrm{Fl}_\varepsilon^{\mathbf{v}}(\Omega))\Big|_{\varepsilon=0} \\ &= \int_{\mathbf{x} \in \partial\Omega} \phi(\mathbf{x}) \mathbf{v}(\mathbf{x}) \cdot \mathbf{n} \, \mathrm{d}S \end{aligned} \tag{3.38}$$

It makes sense to write $\phi = \frac{\partial f}{\partial \Omega}$ since ϕ encodes variations of f as Ω evolves. With this notation, expression (3.38) is reminiscent of the chain rule for computing the derivative of the composition of two functions. The main differences are that the finite sum is replaced by an integral, and more importantly (this is the true meaning of the structure theorem) that the only non-vanishing contributions occur at the boundary $\partial\Omega$.

Let us denote $\phi_i = \frac{\partial V_i}{\partial \Omega}$ the gradient of V_i with respect to the domain variable. Interestingly enough, ϕ_i can also be interpreted as a shape function, namely as the nodal contribution to a continuous reconstruction. To understand that, let us imagine that the computational domain Ω moves in $\mathbb{R}^d$. Loosely speaking, as $\partial\Omega$ progressively reaches $\mathbf{x}$ with an infinitesimal step $\mathrm{d}\mathbf{x}$, this point contributes to the definition of V_i by an amount of $\phi_i(\mathbf{x})\ \mathrm{d}\mathbf{x}\cdot\mathbf{n}\ \mathrm{d}S = \phi_i(\mathbf{x})\ \mathrm{d}V$. This notion is very close to that of "partition of unity", which appears in the definition of the integral over a manifold (see chapter 8 of [2]).

Using the reconstructed velocity field $<\mathbf{v}>$ of expression (3.36) as an advection field for points of Ω, we can write definition 3.1 for all discrete scalar field $f : \mathcal{C} \to \mathbb{R}$ constant with respect to time ($\frac{\mathrm{d}f}{\mathrm{d}t} = 0$) as:

$$\begin{aligned}
\oint\!\!\!\!\!\!\diagup_{\mathcal{C}} f \mathbb{\nabla}\cdot\mathbf{v} &\overset{\text{def}}{=} \frac{\mathrm{d}}{\mathrm{d}t}\oint\!\!\!\!\!\!\diagup_{\mathcal{C}} f \\
&= \sum_{i\in\mathcal{C}} f_i \sum_{j\in\mathcal{C}} \left(\frac{\partial V_i}{\partial\Omega}\cdot\frac{\partial\Omega}{\partial\mathbf{x}_j} + \frac{\partial V_i}{\partial\mathbf{x}_j}\right)\cdot\mathbf{v}_j \\
&= \sum_{i\in\mathcal{C}}\sum_{j\in\mathcal{C}} f_i \left(\int_{\partial\Omega}\phi_i\Psi_j\cdot\mathbf{n}\ \mathrm{d}S\right)\cdot\mathbf{v}_j - \sum_{i\in\mathcal{C}}\mathbf{v}_i\cdot\sum_{j\in\mathcal{C}} -\frac{\partial V_j}{\partial\mathbf{x}_i} f_j
\end{aligned} \tag{3.39}$$

The first term of expression (3.39) vanishes far from the boundary . It is thus tempting to give the following definition:

Definition 3.6 (Volume-based boundary integration operator)**.**
Given a volume function V, the volume-based boundary integration operator is defined as:

$$\oint\!\!\!\!\!\!\diagup_{\partial\mathcal{C}} f\mathbf{v} \overset{\text{def}}{=} \sum_{i,j\in\mathcal{C}} f_i \left(\int_{\partial\Omega}\phi_i\Psi_j\cdot\mathbf{n}\ \mathrm{d}S\right)\cdot\mathbf{v}_j \tag{3.40}$$

In terms of the boundary coordinates $\mathbf{\Gamma}_{i,j}$ (see definition 2.9), the above definition reads:

$$\mathbf{\Gamma}_{i,j} \overset{\text{def}}{=} \int_{\mathbf{x}\in\partial\Omega}\phi_i(\mathbf{x})\Psi_j(\mathbf{x})\cdot\mathbf{n}\ \mathrm{d}S \tag{3.41}$$

A priori, $\oint\!\!\!\!\!\!\diagup_{\partial\mathcal{C}}$ is a non-symmetric bi-linear operator. This is consistent with definition 2.9. Inspecting the cases where $\mathbf{\Gamma}_{i,j} = \mathbf{0}$, this definition conveys two different notions of "being a boundary discretization node":

- A left-sided version: $\frac{\partial V_i}{\partial\Omega} = \mathbf{0} \ \Rightarrow\ \forall\ j\in\mathcal{C}, \mathbf{\Gamma}_{i,j} = \mathbf{0}$. If V_i does not locally depend on Ω, then i is inside, else it is on the boundary.
- A dual right-sided version: $\frac{\partial\Omega}{\partial\mathbf{x}_j} = 0 \ \Rightarrow\ \forall\ i\in\mathcal{C}, \mathbf{\Gamma}_{i,j} = \mathbf{0}$. If Ω does not locally depend on $\mathbf{x}_j$, then j is inside, else (in accordance with the structure theorem) it is a boundary discretization node.

In general, boundary edges are characterized by overlaps of the supports of $\frac{\partial V_i}{\partial\Omega}$ and $\frac{\partial\Omega}{\partial\mathbf{x}_j}$ on the boundary:

$$(i,j)\in\partial\mathcal{C} \quad\Rightarrow\quad \exists\ \mathbf{x}\in\partial\Omega \mid \phi_i(\mathbf{x}) \neq \mathbf{0} \text{ and } \Psi_j(\mathbf{x}) \neq 0 \tag{3.42}$$

We do not give any expression for the error of the volume-based boundary integration operator, or characterization of its consistency. Instead, the properties of this operator should be analyzed through those of the primal and dual volume-based gradients.

In the next section, we identify the leftmost sum of expression (3.39) as the dual differentiation operator.

3.3 Volumes depend on the computational domain: Definition of the resulting dual gradient

In accordance with definition 2.17, the presence of nodal volume $\oint\!\!\!\!\!\!\int_{\mathcal{C}}$ and boundary $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}}$ integration operators and a discrete gradient operator $\mathbb{\nabla}$ readily entail the definition of a dual gradient operator:

Definition 3.7 (Volume-based dual gradient)**.**
As a consequence of expression (3.39) for the primal gradient coefficients and of definition 3.6 of the boundary integration operator, the volume-based dual gradient is fully determined. It reads: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}^*_i f = -\sum_{j\in\mathcal{C}} \frac{\partial V_j}{\partial \mathbf{x}_i} f_j \tag{3.43}$$

The corresponding coordinates of the volume-based dual gradient read: $\forall\ i, j \in \mathcal{C}$,

$$\mathbf{A}^*_{i,j} \overset{\text{def}}{=} -\frac{\partial V_j}{\partial \mathbf{x}_i} \tag{3.44}$$

Remark. Ignoring boundary effects, the graph of the volume-based dual gradient is the transpose of that of the primal gradient: the edge (i, j) is present in this graph if $\frac{\partial V_j}{\partial \mathbf{x}_i} \neq \mathbf{0}$, *i.e.* if the volume of node j does not locally depend on the position of node i as Ω is kept fixed. As a consequence, the value of the dual gradient is related to the variation of the contributions of each function value (through the volume weights) to the total integral as a *single* node is moved. This behavior is exactly the opposite of that of the primal gradient

If the primal gradient emerges from the advection of both the computational domain Ω and the discretization nodes $(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n})$ (linked through the definition of shape functions Ψ), then separating the respective influences of both terms allows the definition of the boundary operator $\oint\!\!\!\!\!\!\int_{\mathcal{C}}$ and the dual gradient $\mathbb{\nabla}^*$.

In this sense, $\mathbb{\nabla}^*$ and $\oint\!\!\!\!\!\!\int_{\partial\Omega}$ are more fundamental than $\mathbb{\nabla}$, which is a composite operator born from their combination. Hence the most logical way to introduce these operators would have been to first define the volume dependency as $V_i(\Omega, \mathbf{x}_1, \ldots, \mathbf{x}_{n_n})$ and define $\mathbb{\nabla}^*$, then link the movement of the computational space and discretization nodes through $< \mathbf{v} >$ and define $\oint\!\!\!\!\!\!\int_{\partial\mathcal{C}}$. At this point, the definition $\mathbb{\nabla}$ is a direct consequence of the chain rule *i.e.* the primal gradient should be understood as a total derivative. Instead, we chose to first introduce the primal gradient as it chronologically was the first to be discovered. Moreover, we feel that the mathematical tools needed for its definition are easier to manipulate. Incidentally, this might be one of the reasons why the dependency of volume functions with respect to the domain Ω has been completely overlooked in the meshless literature, often leading to *ad hoc* treatments of the natural boundary conditions.

Corollary 3.10 (Point-wise expression of the volume-based dual gradient)**.**
We have seen that the volume-based dual gradient arises from the *intrinsic* dependency of

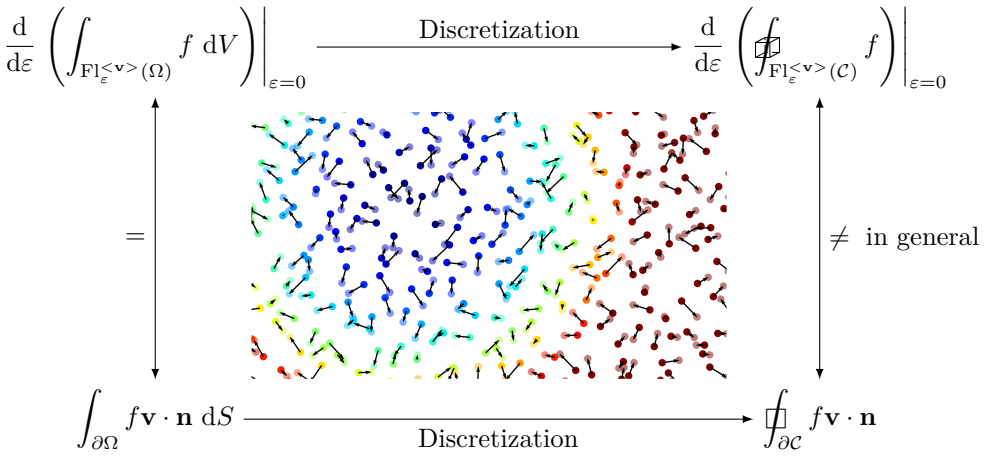


Figure 3.2 – *Reynolds transport theorem is in general not valid at the discrete level. The picture at the center represents a point cloud, and a smooth scalar field defined at the nodes. Each node is nudged slightly in an arbitrary direction and the point values of the scalar field changes accordingly. On the left, the Reynolds transport theorem states that the integral only changes by a boundary flux term. On the right, the discrete correspondent quantities do not match. The associated error is directly linked to the consistency error of the dual gradient as stated in proposition 3.11.*

the discrete integral with respect to discrete node position. Consequently, it is independent of the choice of shape functions $(\Psi_j)_{j\in\mathcal{C}}$ (see section 3.2.1), and it can be expressed as the derivative of the discrete integral, considering the computational domain Ω as motionless: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}^* f = -\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_i} \oiint_{\mathcal{C}} f \Big|_{\Omega=\mathrm{cst}} \tag{3.45}$$

In the next section, we characterize the consistency of the volume-based dual gradient operator in terms of the volume function from which it is derived.

3.3.1 Local error of volume-based dual gradients

In order to translate the consistency results of section 3.1.2 to the dual gradient operator, we need to consider a situation where the two influences do not overlap. In the continuous world, the Reynolds transport theorem states that the variations of the integral of a function over a domain Ω bear the independent contributions of the variation of the domain Ω itself through a boundary integral, and of the intrinsic dependency of the function with respect to the parameter. For a function independent of the parameter ($\frac{\partial f}{\partial \varepsilon} = 0$), this situation is represented on the left side of figure 3.2. In the discrete world, the Reynolds transport theorem does not hold in general (right side of figure 3.2). Interestingly enough, its discrete failure (*i.e.* the lack of commutation of the diagram in figure 3.2.) is related to the consistency of the dual gradient via the following proposition:

Proposition 3.11 (Expression of the error of the dual gradient)**.**
For a regular enough continuous field f and a discrete field $\mathbf{v}$, the following formula holds:

$$\oiint_{\mathcal{C}} (\mathbb{\nabla}^* - \nabla) f \cdot \mathbf{v} = \oiint_{\partial\mathcal{C}} f\mathbf{v} - \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \oiint_{\mathrm{Fl}_\varepsilon^{<\mathbf{v}>}(\mathcal{C})} f \Bigg|_{\varepsilon=0} \tag{3.46}$$

The above formula is valid if we consider that discretization nodes $(\mathbf{x}_k)_{k\in\mathcal{C}}$ are advected with the discrete velocity field $\mathbf{v} : \mathcal{C} \to \mathbb{R}^d$, whereas the points of the computational domain are advected with the reconstructed velocity field $< \mathbf{v} >$.

Proof. The proof is nothing but a direct computation:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\varepsilon} \oiint_{\mathrm{Fl}_\varepsilon^{<\mathbf{v}>}(\mathcal{C})} f \Big|_{\varepsilon=0} &= \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \sum_{i\in\mathcal{C}} V_i(\mathrm{Fl}_\varepsilon^{<\mathbf{v}>}(\Omega), \mathbf{x}_1(\varepsilon), \dots, \mathbf{x}_{n_n}(\varepsilon))\ f(\mathbf{x}_i(\varepsilon)) \Big|_{\varepsilon=0} \\
&= \sum_{i\in\mathcal{C}} \left(\frac{\partial V_i}{\partial \Omega} \cdot \frac{\mathrm{d}\Omega}{\mathrm{d}\varepsilon} + \sum_{j\in\mathcal{C}} \frac{\partial V_i}{\partial \mathbf{x}_j} \cdot \frac{\mathrm{d}\mathbf{x}_j}{\mathrm{d}\varepsilon} \right) f(\mathbf{x}_i) + V_i \nabla f(\mathbf{x}_i) \cdot \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}\varepsilon} \\
&= \sum_{i\in\mathcal{C}} \left(\int_{\partial\Omega} \phi_i(\mathbf{x}) < \mathbf{v} > (\mathbf{x}) \cdot \mathbf{n} + \sum_{j\in\mathcal{C}} \frac{\partial V_i}{\partial \mathbf{x}_j} \cdot \mathbf{v}_j \right) f(\mathbf{x}_i) + V_i \nabla f(\mathbf{x}_i) \cdot \mathbf{v}_i \\
&= \oint_{\partial\mathcal{C}} f\mathbf{v} + \oiint_{\mathcal{C}} (\nabla - \mathbb{V}^*)\, f \cdot \mathbf{v}
\end{aligned}$$

□

While the above form of the theorem highlights the correspondence between the continuous and the discrete world, it might look a bit confusing. It can be re-written under the following node-wise form, in which no flow or reconstruction appear:

Corollary 3.12 (Point-wise expression of the error of the dual gradient)**.**
The point-wise error of the volume-based dual gradient of continuous function $f : \mathbb{R}^d \to \mathbb{R}$ can be expressed as:

$$V_i(\mathbb{V}_i^* f - \nabla f(\mathbf{x}_i)) = -\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_i} \oiint_{\mathcal{C}} f \Big|_{\Omega=\mathrm{cst}} \tag{3.47}$$

The RHS of expression (3.47) is formally identical to that of corollary 3.10, but its meaning is different. Here, f is a continuous function evaluated at discretization nodes and this value changes as nodes are moved. In expression (3.45), f is a discrete function, *i.e.* a collection of values attached to the discretization nodes and these values do not change during advection.

This expression allows the following interpretation of the dual gradient: in sharp contrast with the primal gradient, the dual gradient is not related to advection since the computational space Ω is held fixed as the discretization nodes move. And indeed, the choice of shape functions Ψ_j (relating the advection of nodes of the point cloud and points of space, see equation (3.36)) does *not* play a role in the definition of the dual gradient. Rather, the value of the dual gradient of a function f evaluated at node i depends on the local variation of *every* volume weight V_j as a *single* node i is perturbed. The consistency order of the dual gradient thus characterizes how well the volumes work *together* regardless of the discretization node positions.

Remark. In particular for a constant unit function, expression (3.47) gives a relationship between the error on the dual gradient of a constant field and the variations of the total discretized volume as a node is nudged. This fact is exploited in section 3.6.1 where several common regularization procedures are re-interpreted as total volume maximization procedures, eventually yielding the sought-after zeroth order consistency of the dual gradient.

3.3.2 Global error of volume-based dual gradients

From the previous local result, we can formulate the following global consistency requirement:

Proposition 3.13 (Global consistency conditions of volume-based dual gradients)**.**
The dual gradient of a scalar function is exact at every node position of an open set of configurations if and only if the value of its discrete integral is independent of the said node positions:

$$\oint\!\!\!\!\!\!\diagup_{\mathcal{C}} f = \mathrm{cst}(\Omega) \tag{3.48}$$

In particular, global consistency of the discrete integration operator entails global consistency of the resulting dual gradient operator.

Proof. This is a direct consequence of corollary 3.12. □

Example (Characterization of global dual consistency of order zero and one)**.**
Necessary and sufficient conditions for global zeroth and first order consistency at node $i \in \mathcal{C}$ of the volume-based dual gradient operator are the independence of the total volume and of the centroid of the point cloud with respect to node positions: $\forall\ (\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) : \mathcal{C} \to \mathbb{R}^d$,

$$V_{\mathrm{tot}}(\mathcal{C}) = \sum_{i \in \mathcal{C}} V_i = \mathrm{cst} \tag{3.49}$$

$$V_{\mathrm{tot}}(\mathcal{C})\mathbf{x}_{\mathrm{tot}}(\mathcal{C}) = \sum_{i \in \mathcal{C}} V_i \mathbf{x}_i = \mathbf{cst} \tag{3.50}$$

Contrary to the primal case (see section 3.1.4), there does exist volume functions such that the corresponding dual gradient operator is both local and linear consistent. Actually, dual consistency of any order can be achieved as shown in section 3.7.

3.3.3 Comparison between the primal and dual volume-based gradients

The primal and dual volume-based gradients are two opposite sides of the same coin. As we have seen in earlier sections, the expression of the primal gradient at a given node is given by the variation of one volume weight during the collective movement of all neighboring discretization nodes. Its quality is thus impacted by co-variance properties of the volume method with respect to space transformations. Unfortunately, linear space transformations can stretch neighborhoods at will and consequently, linear consistent volume-based primal gradients are not local.

On the other hand, the expression of the dual gradient at a given node is given by the variation of the opposite of the integral when the node position is perturbed. Hence, its quality properties describe how well volume weights all work together to give an approximation of integrals independently of the position of each individual node. This does not preclude the dual gradient to achieve arbitrarily high orders of consistency while remaining local.

Figure 3.3 recapitulates the important formulae of the previous sections, and compares the behavior of the primal and dual volume-based gradients.

The following proposition investigates the possibility of defining a gradient using the volume method with built-in symmetry (i.e. $\mathbb{\nabla} = \mathbb{\nabla}^*$).

	Primal Gradient: $V_i\mathbb{\nabla}_i \cdot \mathbf{v} = \sum_{j\in\mathcal{C}} \frac{\mathrm{d}V_i}{\mathrm{d}\mathbf{x}_j} \cdot \mathbf{v}_j$	Dual Gradient: $V_i\mathbb{\nabla}_i^* f = -\sum_{j\in\mathcal{C}} \frac{\partial V_j}{\partial \mathbf{x}_i} f_j$
Point-wise expression for discrete fields	$V_i\mathbb{\nabla}_i \cdot \mathbf{v} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} V_i(\mathrm{Fl}_\varepsilon^{\mathbf{v}}(\mathbf{x}_1), \dots, \mathrm{Fl}_\varepsilon^{\mathbf{v}}(\mathbf{x}_{n_n}))\Big\vert_{\varepsilon=0}$ Corollary 3.3	$V_i\mathbb{\nabla}^* f = -\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_i} \oint\!\!\!\!\!\!\square_{\mathcal{C}} f\Big\vert_{\Omega}$ Corollary 3.10
Integral form of the error on continuous fields	$\oint\!\!\!\!\!\!\square_{\mathcal{C}} f\,(\mathbb{\nabla} - \nabla)\cdot\mathbf{v} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left(\oint\!\!\!\!\!\!\square_{\mathrm{Fl}_\varepsilon^{\mathbf{v}}(\mathcal{C})} f \;-\; \oint\!\!\!\!\!\!\square_{\mathcal{C}} f\circ \mathrm{Fl}_\varepsilon^{\mathbf{v}} \det(\mathrm{J}_{\mathrm{Fl}_\varepsilon^{\mathbf{v}}})\right)\Big\vert_{\varepsilon=0}$ Proposition 3.5	$\oint\!\!\!\!\!\!\square_{\mathcal{C}} (\mathbb{\nabla}^* - \nabla)\, f\cdot\mathbf{v} = \oint\!\!\!\!\!\!\square_{\partial\mathcal{C}} f\mathbf{v} - \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \oint\!\!\!\!\!\!\square_{\mathrm{Fl}_\varepsilon^{<\mathbf{v}>}(\mathcal{C})} f\;\Big\vert_{\varepsilon=0}$ Proposition 3.11
Point-wise form of the error on continuous fields	$V_i(\mathbb{\nabla}_i\cdot\mathbf{v} - \nabla\cdot\mathbf{v}(\mathbf{x}_i)) = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\frac{V_i(\mathrm{Fl}_\varepsilon^{\mathbf{v}}(\mathbf{x}_1),\dots,\mathrm{Fl}_\varepsilon^{\mathbf{v}}(\mathbf{x}_{n_n}))}{\det(\mathrm{J}_{\mathrm{Fl}_\varepsilon^{\mathbf{v}}}(\mathbf{x}_i))}\Big\vert_{\varepsilon=0}$ Corollary 3.6	$V_i(\mathbb{\nabla}_i^* f - \nabla f(\mathbf{x}_i)) = -\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_i} \oint\!\!\!\!\!\!\square_{\mathcal{C}} f\Big\vert_{\Omega=\mathrm{cst}}$ Corollary 3.12
Exactness conditions on continuous fields	$V_i(\mathrm{Fl}_\varepsilon^{\mathbf{v}}(\mathbf{x}_1),\dots,\mathrm{Fl}_\varepsilon^{\mathbf{v}}(\mathbf{x}_{n_n})) = \det(\mathrm{J}_{\mathrm{Fl}_\varepsilon^{\mathbf{v}}}(\mathbf{x}_i)) V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})$ Proposition 3.7	$\oint\!\!\!\!\!\!\square_{\mathcal{C}} f = \mathrm{cst}(\Omega)$ Proposition 3.13
Consistency of order zero	$V_i(\mathbf{x}_1+\mathbf{a},\dots,\mathbf{x}_{n_n}+\mathbf{a}) = V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})$ (3.23)	$\sum_{i\in\mathcal{C}} V_i = \mathrm{cst}$ (3.49)
Consistency of order one	$V_i(\mathrm{B}\mathbf{x}_1,\dots,\mathrm{B}\mathbf{x}_{n_n}) = \det(\mathrm{B}) V_i(\mathbf{x}_1,\dots,\mathbf{x}_{n_n})$ (3.24)	$\sum_{i\in\mathcal{C}} V_i\mathbf{x}_i = \mathbf{cst}$ (3.50)
Possibility of local linear consistent gradient?	✗ Conjecture 3.8	✓ See section 3.7.1

Figure 3.3 – *Compared summary of the properties of the primal and dual volume-based gradient*

Proposition 3.14 (Characterization of volume-based symmetric gradients)**.**
Ignoring the boundary, demanding that the primal and dual volume-based gradients are equal reads in terms of coefficients: $\forall\ i, j \in \mathcal{C}$,

$$\frac{\partial V_i}{\partial \mathbf{x}_j} + \frac{\partial V_j}{\partial \mathbf{x}_i} = \mathbf{0} \tag{3.51}$$

These equations formally look like Killing equations in differential geometry. If we assume each $V_i,\ i \in \mathcal{C}$ to be globally twice continuously differentiable, we have the following representation property: $\forall\ i, j, k \in \mathcal{C},\ \exists\ \mathrm{Z}_{\{i,j,k\}} \in \mathrm{M}_d(\mathbb{R}) \mid \mathrm{Z}^T_{\{i,j,k\}} = -\mathrm{Z}_{\{i,j,k\}}$ and:

$$\frac{\partial^2 V_i}{\partial \mathbf{x}_j \mathbf{x}_k} = \frac{\partial^2 V_k}{\partial \mathbf{x}_i \mathbf{x}_j} = \frac{\partial^2 V_j}{\partial \mathbf{x}_k \mathbf{x}_i} = \mathrm{Z}_{\{i,j,k\}} \tag{3.52}$$

Proof. Differentiation of expression (3.51) with respect to $\mathbf{x}_k$ and permutation of the derivatives in the first term (to restore cyclic invariance) gives: $\forall\ i, j, k \in \mathcal{C}$,

$$\left(\frac{\partial^2 V_i}{\partial \mathbf{x}_j \partial \mathbf{x}_k}\right)^T + \frac{\partial V_j}{\partial \mathbf{x}_k \partial \mathbf{x}_i} = 0 \tag{3.53}$$

Then the permutation $i \mapsto k,\ j \mapsto i,\ k \mapsto j$ reads: $\forall\ i, j, k \in \mathcal{C}$,

$$\left(\frac{\partial^2 V_k}{\partial \mathbf{x}_i \partial \mathbf{x}_j}\right)^T + \frac{\partial V_i}{\partial \mathbf{x}_j \partial \mathbf{x}_k} = 0 \tag{3.54}$$

Combining expressions (3.53) and (3.54), and completing the cyclic permutation, we get: $\forall\ i, j, k \in \mathcal{C}$,

$$\frac{\partial V_j}{\partial \mathbf{x}_k \partial \mathbf{x}_i} = \frac{\partial^2 V_k}{\partial \mathbf{x}_i \partial \mathbf{x}_j} = \frac{\partial^2 V_i}{\partial \mathbf{x}_j \partial \mathbf{x}_k} \tag{3.55}$$

Furthermore, comparing expression (3.55) and expression (3.54), we conclude that all these matrices are skew symmetric. □

Remark. In the case of one dimensional discretizations ($d = 1$), we have $\mathrm{Z}_{\{i,j,k\}} = 0$ because of skew-symmetry, and thus each V_i is linear: $\exists\ w_{i,j} | w_{i,j} = -w_{j,i}$ and

$$V_i = V_{o\ i} + \sum_{j \in \mathcal{N}(i)} w_{i,j} x_j \tag{3.56}$$

Order zero consistency reads $\forall\ i \in \mathcal{C},\ \sum_{j \in \mathcal{N}(i)} w_{i,j} = 0$ and order one consistency reads $\forall\ \ i \in \mathcal{C},\ V_{o\ i} = 0$ Thus, such a volume formula reads:

$$V_i = \sum_{j \in \mathcal{N}(i)} w_{i,j}(x_j - x_i) \tag{3.57}$$

The properties of these symmetric one-dimensional discretization operators are discussed in great lengths in [129].

3.4 Volumes depend on the number of nodes: Definition and characterization of Sibson shape functions

In the previous section, we investigated the consequences of considering the computational domain as a variable in the definition of volume functions. In this section, we now

consider that the number of nodes itself is a variable coming into play in the definition of volume functions. This allows us on the one hand to generalize the well-known Sibson shape functions to arbitrary volume functions and on the other hand to provide a new perspective to the study of the dual gradient.

3.4.1 Definition and properties of Sibson shape functions

Let us consider the two following point clouds ($\mathcal{C} = [\![1, n_n]\!], (\mathbf{x}_1, \ldots, \mathbf{x}_{n_n})$) and ($\mathcal{C}' = [\![1, n_n + 1]\!], (\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})$), with associated volume weights V and V'. Clearly, $\mathcal{C}'$ extends $\mathcal{C}$ by the addition of a $(n_n + 1)$th "fictitious" node at position $\mathbf{x}$.

Definition 3.8 (Sibson shape functions)**.**
With the above definitions, we define the evaluation of Sibson shape functions on $\mathcal{C}$ at point $\mathbf{x}$ as: $\forall\, i \in \mathcal{C}$, $\forall\, \mathbf{x} \in \mathbb{R}^d \mid V'_{n_n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x}) \neq 0$,

$$\phi_i^S(\mathbf{x}) \overset{\text{def}}{=} \frac{V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) - V'_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})}{V'_{n_n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})} \tag{3.58}$$

This generic construction is identical to that of Sibson in [208] for the particular case of Voronoi tessellations (see section F), and for this reason we call them the Sibson shape functions. Although elaborating shape functions from volumes is a recurring theme (see [99] for several examples), we did not come across any occurrence of the above general construction in the dedicated literature.

Proposition 3.15 (Characterization of reproducibility of Sibson shape functions)**.**
Let us assume that the discrete volume integral of a given continuous function $f : \mathbb{R}^d \to \mathbb{R}$ is invariant with respect to the addition of a node at position $\mathbf{x} \in \Omega$ into the point cloud. Then, the Sibson shape functions reproduce f at $\mathbf{x}$.

Proof. The key point is to assume that the discrete volume integral of a given function $f : \mathbb{R}^d \to \mathbb{R}$ is invariant with respect to this cloud manipulation: $\forall\, \mathbf{x} \in \mathbb{R}^d \mid V'_{n_n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x}) \neq 0$,

$$\begin{aligned}
\int\!\!\!\!\!\!\diagup_{\mathcal{C}'} f &= \int\!\!\!\!\!\!\diagup_{\mathcal{C}} f \\
\sum_{i \in \mathcal{C}'} V'_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x}) f(\mathbf{x}_i) &= \sum_{i \in \mathcal{C}} V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) f(\mathbf{x}_i) \\
V'_{n_n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x}) f(\mathbf{x}) &= \sum_{i \in \mathcal{C}} \left(V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) - V'_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})\right) f(\mathbf{x}_i) \\
f(\mathbf{x}) &= \sum_{i \in \mathcal{C}} \frac{V_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}) - V'_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})}{V'_{n_n+1}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})} f(\mathbf{x}_i) \\
&= < f >_{\mathcal{C}}^S (\mathbf{x})
\end{aligned}$$

Hence, f is reproduced at point $\mathbf{x}$ thus concluding the proof. □

Remark. Proposition 3.15 essentially states that the Sibson reconstruction is exact on $f : \Omega \to \mathbb{R}$ if $\int\!\!\!\!\!\!\diagup_{\mathcal{C}} f$ does not depend on the number of nodes in the discretization. This characterization of exactness is very close to that of the dual gradient: $\mathbb{\nabla}^*$ is exact on $f : \Omega \to \mathbb{R}$ if $\int\!\!\!\!\!\!\diagup_{\mathcal{C}} f$ does not depend on the positions of the nodes. The link between these two operators is further discussed in section 3.4.2.

Proposition 3.16 (Support of Sibson shape functions)**.**
With the above notation, the volume of a node $i \in \mathcal{C}$ is left unchanged by the addition of a $(n_n + 1)$th node at position $\mathbf{x}$, if and only if $\phi_i^S(\mathbf{x}) = 0$. In particular, the Sibson shape function ϕ_i^S has a compact support if and only if the addition of a node at $\mathbf{x}$ in $\mathcal{C}$ only influences nearby nodes volume weights.

Moreover, assuming node indiscernibility (see definition 3.2), we have the additional characterization: If (j, i) is an edge of the graph of $\mathbb{V}^{\mathcal{C}*}$, then $(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}_j)$ is in the support of ϕ_i^S.

Proof. The first result is a trivial consequence of the definition of Sibson shape functions (definition 3.8). To prove the second result, suppose $(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}_j)$ is not in the support of ϕ_i^S. Then, $\forall\ k \in \mathcal{C}$,

$$\frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) = \frac{\partial V_i'}{\partial \mathbf{x}_j}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}_j) \tag{3.59}$$

And,

$$\mathbf{0} = \frac{\partial V_i'}{\partial \mathbf{x}}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}_j) \tag{3.60}$$

Let us consider the transposition $\pi \in Sym(\mathcal{C}')$ defined as $\pi(j) = n_n + 1$, $\pi(n_n + 1) = j$, and $\pi(i) = i\ \forall\ i \neq j \in \mathcal{C}$. Then using proposition 3.2, expression (3.60) entails:

$$\begin{aligned}
\mathbf{0} &= \frac{\partial V'_{\pi(i)}}{\partial \mathbf{x}_{\pi(j)}}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}_j) \\
&= \frac{\partial V_i'}{\partial \mathbf{x}_j}(\mathbf{x}_{\pi(1)}, \dots, \mathbf{x}_{\pi(n_n)}, \mathbf{x}_{\pi(j)}) \\
&= \frac{\partial V_i'}{\partial \mathbf{x}_j}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}_j) \\
&= \frac{\partial V_i}{\partial \mathbf{x}_j}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n})
\end{aligned}$$

Hence , (j, i) is not an edge of $\mathbb{V}^{\mathcal{C}*}$, thus concluding the proof by contrapositive. □

Proposition 3.17 (Delta-property of Sibson shape functions)**.**
The Sibson shape functions satisfies the delta property if and only if the introduction of a new node in the point cloud at an existing node position $\mathbf{x}_i$ splits the corresponding volume weight V_i into two parts, and does not change the value of any other volume weights. More precisely, we have:

$$\phi_i^S(\mathbf{x}_j) = \delta_{i,j} \quad \forall\ j \in \mathcal{C} \tag{3.61}$$

If and only if:

$$V_i(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) = \lim_{\mathbf{x} \to \mathbf{x}_j} V_i'(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}) \quad \forall\ j \neq i \in \mathcal{C} \tag{3.62}$$

And,

$$V_i(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}) = \lim_{\mathbf{x} \to \mathbf{x}_j} V_i'(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}) + V'_{n_n+1}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}) \tag{3.63}$$

The split condition is stated with a limit to account for the possibility of a discontinuity of the volume function when two node positions are identical.

Proof. This result is a straightforward consequence of definition 3.8. □

Proposition 3.18 (Positivity of Sibson shape functions)**.**
If node volumes weights are positive and the introduction of a node does not increase the value of any volume weight, then the Sibson shape functions are non-negative.

Proof. Again, this result is a direct consequence of definition 3.8. □

The original Sibson shape functions fully exploit the underlying Voronoi tessellation of a point cloud to satisfy the conditions of both propositions 3.17 and 3.18, and are consequently positive and satisfy the delta property.

3.4.2 Link between Sibson shape functions and dual gradient

We have seen in proposition 3.13 that invariance of the discrete integral of a function with respect to node positions meant exactness of the its dual gradient, and in proposition 3.15 that its invariance with respect to node addition meant reproducibility by Sibson shape functions. It is hence natural to expect a direct link between the Sibson shape functions and the dual gradient, given by the following proposition.

Proposition 3.19 (Expression of the gradient of Sibson shape functions)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud equipped with volume function denoted $V^{\mathcal{C}}$. The corresponding dual gradient is denoted $\mathbb{V}^{\mathcal{C}*}$. Let us denote $\mathcal{C}\backslash j$ the cloud made of nodes of $\mathcal{C}$ except j and $< f >^S_{\mathcal{C}\backslash j}$ the Sibson reconstruction of a discrete function f on $\mathcal{C}\backslash j$. Then we have the following relationship between the dual gradient on $\mathcal{C}$ and the evaluation of the gradient of the Sibson reconstruction on $\mathcal{C}\backslash j$:

$$V_j^{\mathcal{C}} \left(\mathbb{V}_j^{\mathcal{C}*} f - \nabla < f >^S_{\mathcal{C}\backslash j} (\mathbf{x}_j) \right) = \left(< f >^S_{\mathcal{C}\backslash j} (\mathbf{x}_j) - f_j \right) \frac{\partial V_j^{\mathcal{C}}}{\partial \mathbf{x}_j} \tag{3.64}$$

Proof. Let us first compute $\nabla \phi_i^S(\mathbf{x}_j)$:

$$\begin{aligned}
\nabla \phi_i^S(\mathbf{x}_j) &= \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_j} \frac{V_i^{\mathcal{C}\backslash j}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_n) - V_i^{\mathcal{C}}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n})}{V_j^{\mathcal{C}}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n})} \\
&= -\frac{V_i^{\mathcal{C}\backslash j} - V_i^{\mathcal{C}}}{V_j^{\mathcal{C}\ 2}} \frac{\partial V_j^{\mathcal{C}}}{\partial \mathbf{x}_j} - \frac{1}{V_j^{\mathcal{C}}} \frac{\partial V_i^{\mathcal{C}}}{\partial \mathbf{x}_j} \\
&= \frac{-1}{V_j^{\mathcal{C}}} \left(\phi_i^S(\mathbf{x}_j) \frac{\partial V_j^{\mathcal{C}}}{\partial \mathbf{x}_j} + \frac{\partial V_i^{\mathcal{C}}}{\partial \mathbf{x}_j} \right)
\end{aligned}$$

And finally,

$$\begin{aligned}
\nabla < f >^S_{\mathcal{C}\backslash j} (\mathbf{x}_j) &= \sum_{\substack{i \in \mathcal{C} \\ i \neq j}} \nabla \phi_i^S(\mathbf{x}_j) f_i \\
&= \frac{-1}{V_j^{\mathcal{C}}} \sum_{i \in \mathcal{C}} \frac{\partial V_i^{\mathcal{C}}}{\partial \mathbf{x}_j} f_i + \frac{1}{V_j^{\mathcal{C}}} \frac{\partial V_j^{\mathcal{C}}}{\partial \mathbf{x}_j} f_j + \frac{-1}{V_j^{\mathcal{C}}} \frac{\partial V_j^{\mathcal{C}}}{\partial \mathbf{x}_j} \sum_{\substack{i \in \mathcal{C} \\ i \neq j}} \phi_i^S(\mathbf{x}_j) f_i \\
&= \mathbb{V}_i^{*\ \mathcal{C}} + \frac{1}{V_j^{\mathcal{C}}} (f_j - < f >^S_{\mathcal{C}\backslash j} (\mathbf{x}_j)) \frac{\partial V_j^{\mathcal{C}}}{\partial \mathbf{x}_j}
\end{aligned}$$

Hence the result. □

Remark. Proposition 3.19 confirms the equivalence between reproducibility of a function f by the Sibson shape functions ϕ^S and the exactness of $\mathbb{\nabla}^{\mathcal{C}*} f$. Moreover, it reveals that the deviation of the dual gradient on $\mathcal{C}$ from the gradient of the Sibson approximation on $\mathcal{C}\backslash j$ is directly controlled by the deviation of the discrete value f_j to the reconstruction at $\mathbf{x}_j$ on $\mathcal{C}\backslash j$. It can thus be shown that under the following (very reasonable for an approximation of order p, see [233] chapter 3) assumptions,

- $V_j^{\mathcal{C}} = \mathcal{O}(h^d)$
- $\frac{\partial V_j^{\mathcal{C}}}{\partial \mathbf{x}_j} = \mathcal{O}(h^{d-1})$
- $\| < f >_{\mathcal{C}\backslash j} (\mathbf{x}_j) - f(\mathbf{x}_j)\| = \mathcal{O}(h^{p+1})$
- $\|\nabla < f >_{\mathcal{C}\backslash j} (\mathbf{x}_j) - \nabla f(\mathbf{x}_j)\| = \mathcal{O}(h^p)$

we can derive:

$$\|\mathbb{\nabla}_j^{\mathcal{C}*} f - \nabla f(\mathbf{x}_j)\| = \mathcal{O}(h^p) \tag{3.65}$$

And thus, the approximation power of the dual gradient is the same as that of the Sibson reconstruction.

In the next sections of this chapter as well as in appendices E and F , we give both well-known and new concrete examples of volume-based meshless operators. In particular, we give formulae for the computation of consistent dual gradients of any given order, effectively showing that contrary to the volume-based primal gradient, the dual gradient suffers no limit of consistency order.

3.5 Smooth Particle Hydrodynamics revisited

We have seen in section 2.5 that the SPH formalism defines volume weights as a function of particle position and constant mass nodal weights. This fits our definition of volume function, and we now study these volume-based SPH operators:

Definition 3.9 (Volume Smooth Particle Hydrodynamics operators)**.**
We have seen in definition 2.19 that given nodal weights m_i, the SPH methodology introduces the following volume function:

$$V_i^{SPH}(\mathbf{x}_1 \ldots, \mathbf{x}_n) = \frac{m_i}{\sum_{j\in\mathcal{C}} m_j W_h(\|\mathbf{x}_j - \mathbf{x}_i\|)} \tag{3.66}$$

V_i^{SPH} is node indiscernible only if all the weights m_i are equal. Furthermore, node j contributes to V_i^{SPH} only if the distance $\|\mathbf{x}_j - \mathbf{x}_i\|$ is lower than h. The volume weights are neither explicitly nor implicitly functions of the computational domain Ω, so the corresponding volume-based boundary integration operator vanishes. In particular, this means that simulation using the SPH operators will always be dependent on *ad hoc* tricks for the imposition of boundary conditions.

Because of their $\|\mathbf{x}_j - \mathbf{x}_i\|$ dependency which is translation and rotation invariant and following the global characterization of primal consistency (proposition 3.7), $\mathbb{\nabla}^{VSPH}$ is consistent of order zero, and $\mathbb{\nabla}^{VSPH}\mathbf{x}$ is a symmetric matrix. Differentiation of the SPH volume function gives: $\forall\ i \neq j$,

$$\mathbf{A}_{i,j}^{VSPH} = -\left(V_i^{SPH}\right)^2 \frac{m_j}{m_i} \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \tag{3.67}$$

In SPH-friendly form, the associated primal gradient thus reads: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$\rho_i \mathbb{\nabla}_i^{VSPH} f = -\sum_{j\in\mathcal{N}(i)} m_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(f_j - f_i) \tag{3.68}$$

This form of the SPH gradient is very similar to the renormalized SPH gradient of order zero (see expression 2.49), the main difference being that the summation includes density weights and not only volume weights.

The volume-based SPH dual gradient reads: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$\rho_i \mathbb{\nabla}_i^{*\ VSPH} f = -\sum_{j \neq i \in \mathcal{C}} m_j \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \left(\frac{f_j}{\rho_j^2} + \frac{f_i}{\rho_i^2} \right) \tag{3.69}$$

Just like its counterparts of section 2.5, the volume-based dual SPH gradient is not even zeroth-order consistent for general node positions since the total volume is not independent of particle positions. It should thus be surprising to know that $\mathbb{\nabla}^{*\ VSPH}$ is the preferred gradient for the evaluation of stresses in the SPH community (see [161] for instance). One of the main reasons for its success is that the consistency properties of the its dual operator (*i.e.* the primal volume-based SPH gradient $\mathbb{\nabla}^{VSPH}$) translate into conservation properties of Lagrangian-based formulations (see appendix G for more details).

Even if the volume SPH operators are not compatible for general node positions, the question remains whether we can find particular node positions for which the compatibility error either vanishes or remains low enough to retain optimal convergence. This idea, which we develop in the next section, amounts to introducing a stabilizing term into the discrete equations. We consider it as one of the most promising way around the meshless curse (see conjecture 2.4).

3.6 Achieving approximate compatibility through the adjustment of nodal positions

Several well-known volume functions (like the SPH volume function) do not provide an approximation of the total volume of the simulation domain which is independent of nodal positions. As a consequence of the global characterization of dual consistency (see proposition 3.13), these volume functions do not yield compatible discrete operators. However, compatibility for arbitrary node position is a stronger requirement than needed for simulation: what we really want is compatibility (or even approximate compatibility as explained in section 4.3.2) for one particular set of nodal positions $\mathbf{X}$. In this section, we take advantage of the point-wise expression of the error of the dual gradient of corollary 3.12 to re-interpret several known stabilization methods as procedures seeking to lower the compatibility error.

3.6.1 Maximizing the total discrete volume

We have seen in corollary 3.12 that the compatibility error of volume-based operators can be written in terms of the total discrete volume as:

$$V_i \mathbb{\nabla}_i^* 1 = -\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_i} \sum_{j \in \mathcal{C}} V_j \Bigg|_{\Omega = \mathrm{cst}} \tag{3.70}$$

Note that in the above expression, only discretization nodes are supposed to move, while the underlying computational domain Ω remains motionless. Hence, in the case of volume-based operators, finding node configurations such that $\mathbb{\nabla}^* 1 = 0$ essentially comes down to finding extremal points of the total volume considered as a function of $\mathbf{X}$. Such configurations can either be extrema of the total volume (either minima or maxima), or saddle points.

In order to understand why maxima of the volume function are the most desirable alternative, let us take the example of the volume-based SPH volumes (see expression 3.66). If every node of the discretization is set at a distance greater than the smoothing length h from every other node in the discretization, then each volume weight is maximal, with value $V_{\max}^{SPH} = \frac{1}{W_h(0)}$. In particular, this shows that the total volume is bounded from above. On the other end of the spectrum, the total volume is minimal when every node is in the same position, and in this case, it has the same value as that of an isolated node: $V_{\text{tot min}}^{SPH} = \frac{1}{W_h(0)}$.

Of course, in such extreme situations, either the density of nodes compared to the smoothing length is not well suited to represent the computational domain, since every node should "be responsible for" a volume greater than the volume enclosed in its "field of view" (the ball centered in the node position of radius h in the case of SPH), or the node distribution does not allow a good representation of continuous fields. Nevertheless, this example hints at the following idea: volume functions are designed so that nodes have greater volumes when they are isolated and far apart from other nodes than when they are clumped together. Maximizing node volumes while constraining node positions inside Ω will thus tend to space out nodes evenly in an harmonious fashion, whereas minimizing node volume tends to group nodes together. Saddle points are in-between configurations. In fact maximization of SPH node volumes has been proposed as a method to generate "blue noise" configuration, *i.e.* configuration of nodes that are both harmonious and unstructured (see [92] for instance).

One of the best known methods to solve this type of optimization problem is the gradient descend, which consists in following gradient flow lines of the function to optimize. In its time-continuous version, it can be stated in the following form: in order to find a local maximum of V_{tot}, advect nodes i of the point cloud with velocity proportional to the gradient of the cost function. In our case, this reads:

$$\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \alpha \frac{\partial V_{\text{tot}}}{\partial \mathbf{x}_i} = -\alpha V_i \mathbb{\nabla}_i^* 1 \tag{3.71}$$

The total volume function indeed increases with (fictitious) time since we have:

$$\begin{aligned} V_{\text{tot}}(t) - V_{\text{tot}}(0) &= \int_0^t \frac{\mathrm{d}V_{\text{tot}}}{\mathrm{d}t}\ \mathrm{d}t \\ &= \int_0^t \sum_{i\in\mathcal{C}} \frac{\partial V_{\text{tot}}}{\partial \mathbf{x}_i} \cdot \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}\ \mathrm{d}t \\ &= \int_0^t \sum_{i\in\mathcal{C}} \left(\frac{\partial V_{\text{tot}}}{\partial \mathbf{x}_i}\right)^2\ \mathrm{d}t\ \geq 0 \end{aligned}$$

As a consequence of monotony and boundedness, $V_{\text{tot}}(t)$ converges to a finite value denoted V_{tot}^{∞} as $t \to +\infty$. Of course, we want to constrain node to stay within the computational domain and avoid the fact that nodes drift as far as not to be in smoothing length range from each other. To that end, we introduce a few layers of regularly spaced nodes around Ω as represented in figure 3.4.

As the nodes move, the value of $V_{\text{tot}}^{\infty} - V_{\text{tot}}(t) = \int_t^{\infty} \sum_{i\in\mathcal{C}} \left(\frac{\partial V_{\text{tot}}}{\partial \mathbf{x}_i}\right)^2\ \mathrm{d}t$ decreases. With an additional regularity assumption on the volume function (such as a Lipschitz condition on the partial derivatives of V_{tot} for instance), we can prove that the value of $\frac{\partial V_{\text{tot}}}{\partial \mathbf{X}} = -V_i \mathbb{\nabla}^* 1$ tends to vanish. In other words, we asymptotically reach a node distribution with exact compatibility.

Several correction procedures present in recent SPH formulations found in the meshless fluid simulation literature bear striking similarities with this idea. In fact, the Fick-based shifting of Lind et al. (see equation (24) of [143]) can be reinterpreted in the light of the

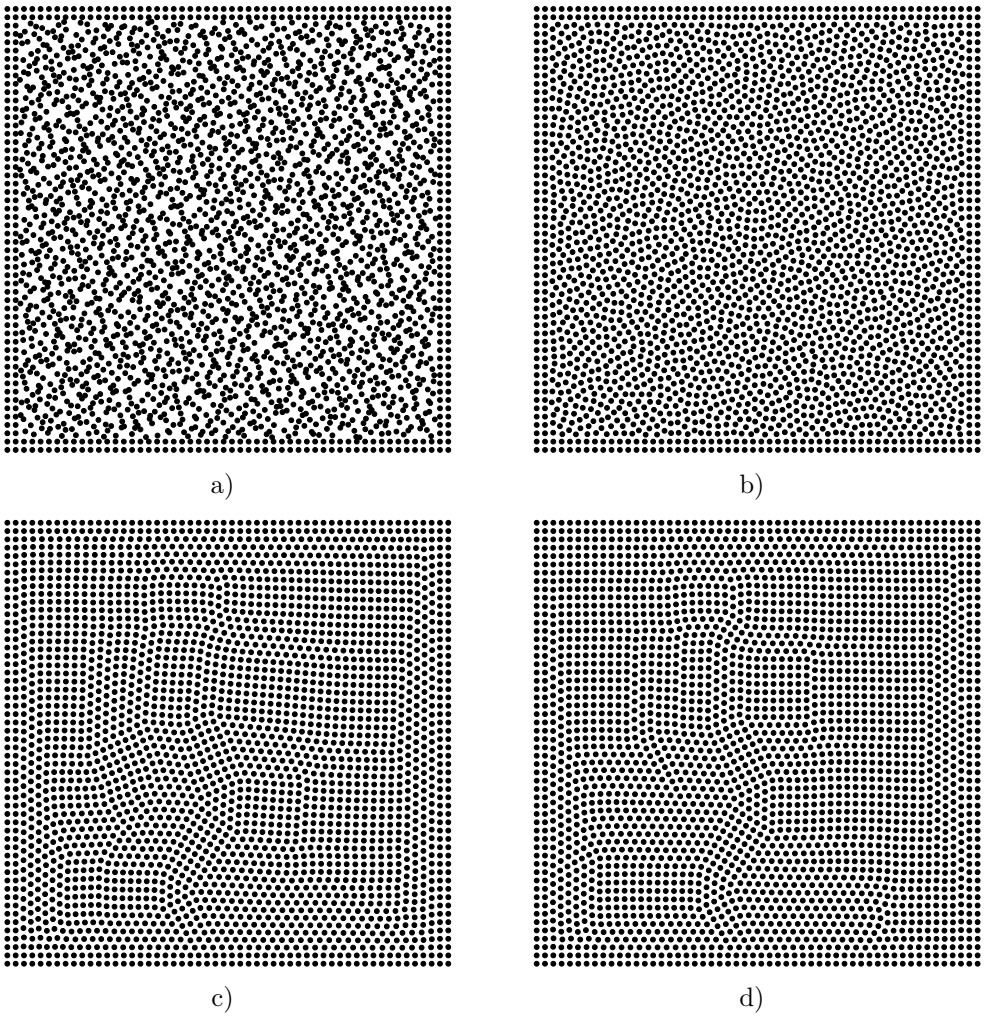


Figure 3.4 – *The generation of point clouds well-suited for meshless simulations is somehow similar to the meshing procedure in mesh-based methods: it should not be performed regardless of the numerical method that follows. The advection method presented in this section adapts a point cloud to the chosen set of operators, lowering the error on the discrete dual gradient on uniform fields.*
a) Initial Halton sequence made of 2500 nodes, enclosed in two layers of regularly distributed nodes. This boundary layer of motionless nodes ensures that interior nodes stay within the computational domain, and that the total volume has local maxima. On this non-optimized node distribution, the Stokes error is $\|\nabla^{VSPH} {}^*1\|_{L^2(\mathcal{C})} \approx 29.1$.
b) The point cloud after a few iterations of the gradient descent. Note how regular the node distribution has become: the inter-node spacing looks much more homogeneous, but periodic or quasi-periodic structures have not yet started to appear. Convergence was stopped as soon as the Stokes error reached $\|\nabla^{VSPH} {}^*1\|_{L^2(\mathcal{C})} \lessapprox 1$.
c) The point cloud after more iterations of the gradient descent algorithm. The node distribution now presents a grain-like structure of regularly arranged nodes reminiscent of atomic arrangement in crystals. Convergence was stopped as soon as the Stokes error reached $\|\nabla^{VSPH} {}^*1\|_{L^2(\mathcal{C})} \lessapprox 5 \times 10^{-3}$.
d) The point cloud after a more strict convergence of the gradient descent algorithm. Visually, the situation is very close to c). Convergence was stopped as soon as the Stokes error reached $\|\nabla^{VSPH} {}^*1\|_{L^2(\mathcal{C})} \lessapprox 5 \times 10^{-8}$.

gradient descent method described above. Indeed, it is tantamount to the subtraction of an Arbitrary Lagrangian-Eulerian (ALE) velocity term of the form $\mathbf{c}_i = \alpha V_i \mathbb{\nabla}_i^* 1$. The final system of Ordinary Differential Equations (ODEs) thus becomes:

$$\forall\ i \in \mathcal{C}, \qquad \begin{cases} m_i \dfrac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \mathbf{f}_i \\ \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_i - \alpha V_i \mathbb{\nabla}_i^* 1 \end{cases} \tag{3.72}$$

This idea is itself a modification of the classical modified SPH (or XSPH) of Monaghan (see equation (2.4) of [164]). More recently, the addition of a uniform background pressure in the conservation of momentum equation has been a recurring theme, with for instance the contributions of Adami et al. (see equation (13) of [4]) and Litvinov et al. (see equation (17) of [146]). Roughly speaking, all these proposition come down to writing the additional term as a force, and not as a velocity:

$$\forall\ i \in \mathcal{C}, \qquad \begin{cases} m_i \dfrac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \mathbf{f}_i - \beta V_i \mathbb{\nabla}_i^* 1 \\ \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{v}_i \end{cases} \tag{3.73}$$

In order to understand why this has a beneficial effect on compatibility, we remark that in the presence of dissipative forces ($\mathbf{f}_i \cdot \mathbf{v}_i < 0$ in expression (3.73), for instance $\mathbf{f}_i = -\mathrm{D}\mathbf{v}_i$ where D denotes a positive definite tensor), the following expression is decreasing:

$$\mathcal{L}\left(\mathbf{X}, \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}t}\right) = \sum_{i \in \mathcal{C}} \frac{1}{2}\mathbf{v}_i^2 - \beta V_i(\mathbf{X}) \tag{3.74}$$

Any equilibrium will thus be a local minimum of $\mathcal{L}$ and verify $\frac{\partial \mathcal{L}}{\partial \mathbf{X}} = \mathbf{0}$, which are exactly the compatibility conditions: $\mathbb{\nabla}^* 1 = \mathbf{0}$.

In the next section, we devise a numerical experiment to test the volume maximization procedure in a simple case and derive guidelines for the most adequate values of the stabilization coefficients α and β.

3.6.2 Numerical experiments

In the context of Lagrangian simulation of a fluid flow, the point cloud is evolving at every time step following the velocity streamlines. In the previous section, we reinterpreted the Fick-based shifting procedure of Lind et al. (see [143]) as an ALE term corresponding to the gradient descent optimization procedure. The aim of this section is to investigate how the addition of this regularizing source term quantitatively impacts the kinematics of the fluid, and to find a balance between the desired compatibility level and the magnitude of the ALE source field.

In order to design a numerical experiment, let us suppose that the physical velocity field is extremely well captured by the simulation. The top left plot of figure 3.5 shows an initially Cartesian nodal distribution advected by a smooth divergence-free Taylor-Green flow after two full turns. Even if the divergence-free condition ensures that the transformation does not change the density at the continuous level, the corresponding node distribution does not look regular any more. The direction of the flow is then reversed, and the nodes come back in their initial positions (top right plot). The middle and bottom rows of figure 3.5 show the very same process, but with non-vanishing gradient descent coefficient α. The node

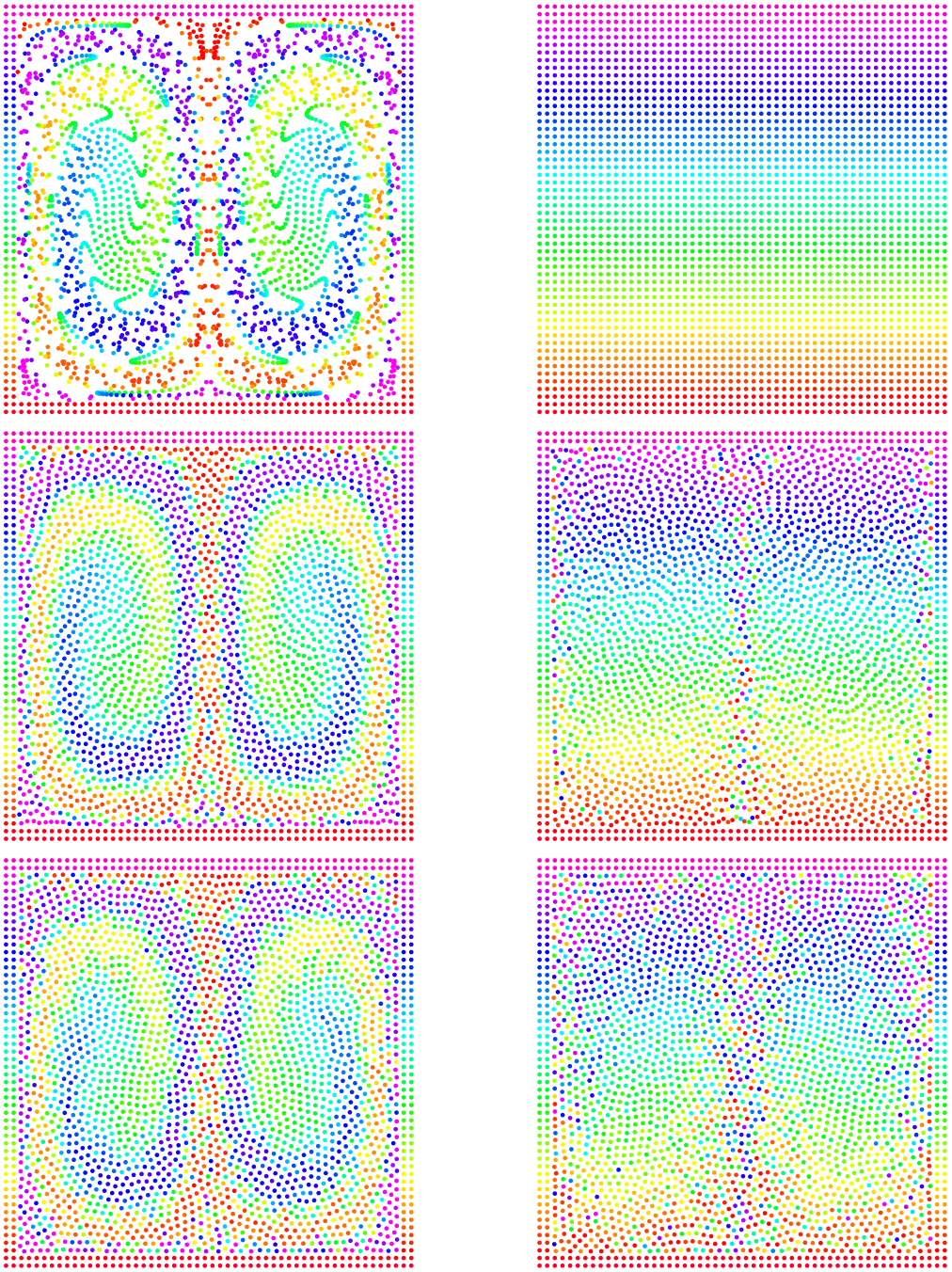


Figure 3.5 – *Even in the case of the divergence-free Taylor-Green flow, a stabilization procedure is necessary to avoid node clumping and creation of voids. The above plots shows the result of the modified advection (see equation (3.72)) of an initially Cartesian distribution on nodes. On the left column, the advection is shown after two full turns. The advection velocity is then reversed and two full turns are again carried out. In order to follow the distribution, nodes are colored according to their initial y-coordinate.*
Top row: the descent coefficient is set to $\alpha = 0$*: the advection is exact. Notice how the initial Cartesian regularity is completely lost at mid-time but then completely recovered at the final time.*
Middle row: the descent coefficient is set to $\alpha = 0.003$*. The node distribution is much more regular and does not present periodic substructures.*
Bottom row: the descent coefficient is set to $\alpha = 0.1$*. Notice how the nodes organize in locally structured patterns reminiscent of atomic arrangement in crystals.*

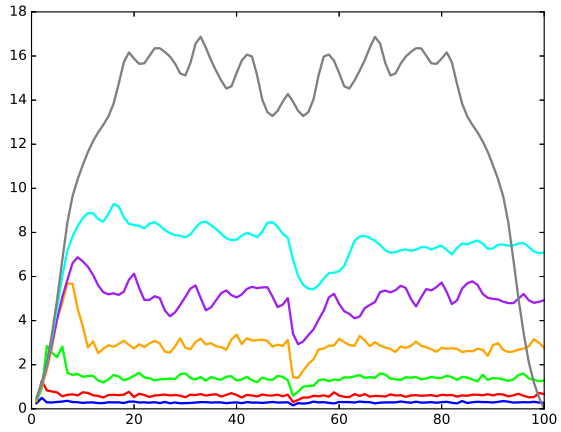


Figure 3.6 – *Stokes L^2 error of an initially Cartesian point cloud of 400 nodes advected by the Taylor-Green flow, and relaxed by formulation 3.72 as a function of time. At mid-time, the Taylor-Green flow is reversed. The system of ODEs is only reversible in the case $\alpha = 0$. The values of α used are: 0.3, 0.1, 0.03, 0.01, 0.003, 0.001 and 0.*

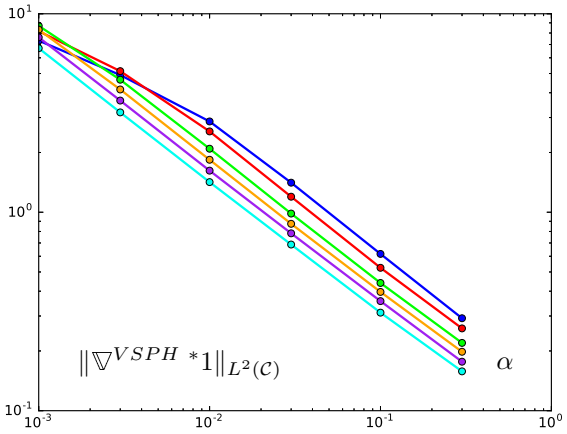
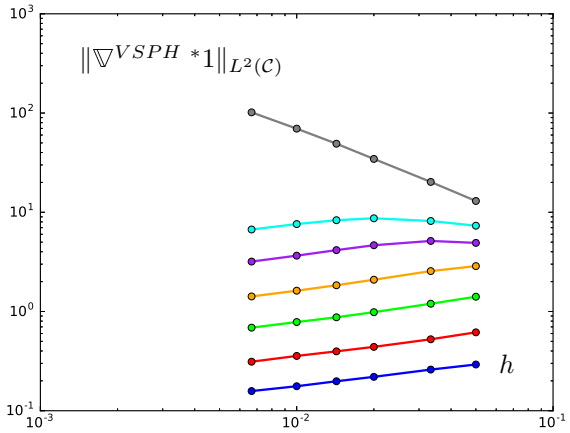


Figure 3.7 – *Time-averaged Stokes' error as a function of the parameter α in the case of the modified Taylor-Green vortex flow. The number of nodes used are: 400, 900, 2500, 4900, 10000 and 22500.*



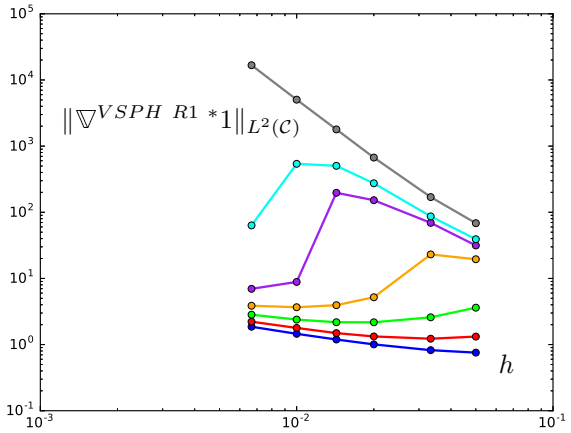


Figure 3.8 – *Time-averaged Stokes error as a function of discretization length in the case of the modified Taylor-Green vortex flow with the SPH volume function. The values of γ used are: 0.3, 0.1, 0.03, 0.01, 0.003, 0.001 and 0.*
Left: Stokes error of the volume-based SPH gradient.
Right: Stokes error of the renormalized volume-based SPH gradient of order one.

distribution remains more regular throughout the process, but nodes do not come back to their original positions after being advected with the reversed flow. Yet, the original colors are still more or less retrieved.

Figure 3.6 gives the instantaneous Stokes error during the advection process for a particular point cloud. In the non-modified case, the flow is exactly reversed. Notice how in the ALE case, the Stokes error manages to stay close to a constant plateau, except at the very beginning, and at mid-time when the flow is reversed. Figure 3.7 gives the time-averaged Stokes error throughout the test for several point cloud refinement levels as a function of the ALE coefficient α. A least-square fit reveals the following approximate dependency: $\|\mathbb{V}^{VSPH\ *}\|_{L^2(\mathcal{C})} \propto \alpha^{-0.65}$.

The left plot of figure 3.8 gives the time-averaged Stokes error of the volume-based SPH operators as a function of the discretization length for several values of the ALE coefficient α. Notice how the behavior is completely different for $\alpha = 0$ and $\alpha \neq 0$. In the first case, the Stokes error grows as $\propto h^{-1}$ whereas as soon as $\alpha \neq 0$, the Stokes error decreases with the smoothing length. A least square fit gives the following approximate dependency: $\|\mathbb{V}^{VSPH\ *}\|_{L^2(\mathcal{C})} \propto h^{0.35}$. Even if these exponents are probably case-dependent, they still seem to indicate that we can choose a value $\alpha = \mathcal{O}(\sqrt{h})$ in order to keep bounded values of the Stokes error as the smoothing length decreases. This choice would ensure that the advection velocity goes to zero as the cloud is refined as $\|\mathbf{c}_i\| = \alpha\|V_i\mathbb{V}_i^*1\| = \mathcal{O}(\sqrt{h})$, so that the method is asymptotically Lagrangian instead of fully ALE. Of course, other compromises could also be made.

Finally, the right plot of figure 3.8 gives the time-averaged Stokes error of the renormalized volume-based SPH operators as a function of the discretization length for point clouds advected in the Taylor-Green flow modified with an ALE term based on the SPH volume. Even if this methodology does not guarantee that the Stokes error should converge to zero – or even decrease at all –, we do observe such a behavior. Unfortunately, its value still increases with h so that the method cannot be made asymptotically Lagrangian while keeping the Stokes error bounded.

The results of this section again confirm that approximate compatibility is sufficient to ensure second order convergence of meshless weak discretizations of elliptic systems. Moreover, the particle arrangement has a big impact on Stokes's error. Even though classical stabilization procedures can be re-interpreted as trying to restore dual consistency of order zero of volume-based gradient operators. We observed that they decrease the Stokes error of other formulations as well, justifying their widespread use.

3.7 The Smoothed Finite Element Method operators revisited in the volume-based approach

In this section, we investigate the operators generated by the SFEM volume function defined in section 2.7.2. In a nutshell, these operators have extremely good properties, achieving consistency of any order for the integration and dual gradient operator as well as compatibility. The mesh-based background integration procedure of section 2.7.3 is adapted to the present context, and is shown to be able to preserve first order consistency and compatibility of the discrete operators.

3.7.1 A volume function from the reconstruction operator

In this section, we assume given a reconstruction operator. This might be the Moving Least Squares (MLS) reconstruction (see section D.2), the P-1 reconstruction on a simplex

mesh (see section 2.7.1), the Reproducing Kernel Hilbert Space (RKHS) reconstruction (see section E), *et cetera* ... The following definition provides volume function from the reconstruction operator. This definition is identical to that of the SFEM integration operator (definition 2.24). We recall it here with slightly different notations to insist on the fact that the shape functions are functions of the evaluation point $\mathbf{x}$, but also of the node positions $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{n_n})$.

Definition 3.10 (Smoothed Finite Element Method volume function)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud, and $< . >: (\mathcal{C} \to \mathbb{R}) \to (\mathbb{R}^d \to \mathbb{R})$ a reconstruction operator on the point cloud. We define the SFEM discrete integration operator as: $\forall\, f : \mathcal{C} \to \mathbb{R}$,

$$\int\!\!\!\!\!\!\diagup_{\mathcal{C}}^{VSFEM} f \overset{\text{def}}{=} \int_{\mathbf{x}\in\Omega} < f > (\mathbf{x})\ \mathrm{d}V \tag{3.75}$$

The corresponding volume weights read:

$$V_i^{VSFEM}(\Omega, \mathbf{x}_1, \dots, \mathbf{x}_{n_n}) \overset{\text{def}}{=} \int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x})\ \mathrm{d}V \tag{3.76}$$

The volume function depends on node positions through the shape functions, and on Ω through the integration domain.

Although these are exactly the classical SFEM volume weights introduced in section 2.7.2, the next proposition shows that applying the theoretical results of sections 3.1-3.4 concerning volume functions to the SFEM volume weights yields differentiation operators that are akin, but not identical to the classical SFEM differentiation operators in general. In particular, while classical SFEM (primal or dual) gradients typically do not achieve high order consistency (see proposition 2.9), we can already state the following result:

Proposition 3.20 (Volume-based dual gradient of arbitrary consistency order)**.**
If $< . >$ reproduces a continuous function f on Ω, then $\mathbb{\nabla}^{VSFEM\ *}$ is exact on f. In particular, there exists volume functions whose derived dual gradient operator is local and consistent of any order $p \in \mathbb{N}$.

This result concerning volume-based dual gradients comes in stark contrast with the primal case, which cannot achieve first order consistency while remaining local (see section 3.1.4).

Proof. There exist local reconstruction operators that reproduce polynomials of any order (see for instance the MLS shape functions and proposition D.4). Since the exactness of the integration operator stated in proposition 2.8 is valid for any node configuration, the value of the discrete integral of a reproduced function is always equal to the value of the exact integral. In particular, this value does not depend on the node positions, and as a consequence of the global consistency conditions given in proposition 3.13, the dual gradient is exact. □

Since it heavily uses previous results, this proof might not seem very convincing. We give an alternative proof using dual gradient coordinates on page 111.

Proposition 3.21 (Expression of the volume-based dual SFEM gradient)**.**
The associated dual gradient operator reads:

$$V_i \mathbb{\nabla}_i^{VSFEM*} f = -\int_{\mathbf{x}\in\Omega} \sum_{j\in\mathcal{C}} \frac{\partial \phi_j}{\partial \mathbf{x}_i}(\mathbf{x}_1, \dots, \mathbf{x}_{n_n}, \mathbf{x}) f_j\ \mathrm{d}V \tag{3.77}$$

Or in coordinate form:

$$\mathbf{A}_{i,j}^{VSFEM*} = -\int_{\mathbf{x}\in\Omega} \frac{\partial \phi_j}{\partial \mathbf{x}_i}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})\ \mathrm{d}V \tag{3.78}$$

Using these expressions, we conclude that (i, j) is an edge of the graph of $\mathbb{V}^{SFEM\ *}$ if and only if the shape function ϕ_j, locally depends on the position of node i. Figure 3.9 goes further in the description of this graph in the case of the Shepard shape functions.

Proof. This is a direct application of the definition of volume-based dual gradient (definition 3.7). □

Using the coordinate form of the dual gradient, let us give an alternative proof of proposition 3.20.

Coordinate proof of proposition 3.20. Differentiating the reproducibility conditions with respect to $\mathbf{x}_i$, we get: $\forall\ \mathbf{x} \in \Omega$,

$$\phi_j(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})\nabla f(\mathbf{x}_i) = -\sum_{j\in\mathcal{C}} \frac{\partial \phi_j}{\partial \mathbf{x}_i}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x}) f(\mathbf{x}_j) \tag{3.79}$$

Integration of the above expression over Ω yields:

$$\left(\int_{\mathbf{x}\in\Omega} \phi_i(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})\ \mathrm{d}V\right) \nabla f(\mathbf{x}_i) = \sum_{j\in\mathcal{C}} \left(-\int_{\mathbf{x}\in\Omega} \frac{\partial \phi_j}{\partial \mathbf{x}_i}(\mathbf{x}_1, \ldots, \mathbf{x}_{n_n}, \mathbf{x})\ \mathrm{d}V\right) f(\mathbf{x}_j) \tag{3.80}$$

Which is exactly the coordinate form of the exactness conditions for the SFEM volume-based dual gradient on f. □

Remark. The expression of the volume-based SFEM dual gradient coefficients incorporates the term $-\frac{\partial \phi_j}{\partial \mathbf{x}_i}$, replacing the term $\phi_i \nabla \phi_j$ of the classical SFEM operator. These two terms behave quite differently of a reproduced function f:

- The classical term $\phi_i \nabla \phi_j$ allows the reproduction of ∇f under the integral sign. If f is affine (as in expression (2.73)), its gradient is constant and matches the nodal value, which is sufficient to ensure first order consistency. Else, the exact gradient gets smoothed away by the ϕ_i factor in the integral and the nodal value is not exactly captured.
- In contrast, the $-\frac{\partial \phi_j}{\partial \mathbf{x}_i}$ term of the volume SFEM allows the exact value of the gradient at node $\mathbf{x}_i$ to appear in expression (3.79) and survive the integration process in expression (3.80), yielding exactness of the dual gradient for any reproduced function.

Remark (Sibson shape functions of SFEM volumes)**.** The methodology described in this section allows the construction of a volume function given a reconstruction operator. As we have seen in section 3.4, it is also possible to generate the Sibson reconstruction operator from a volume function. In both cases, reproducibility of a continuous function f is equivalent to constancy (a slightly weaker notion than exactness) of both the discrete integration and dual gradient operators. It is thus natural to wonder whether the Sibson shape functions of a SFEM volume function are the original shape functions. Such a result would truly be amazing: Integration and addition of a node in the point cloud would thus become two inverse operations. The examples detailed in appendices E and F show that it might or might not be the case depending on the specificities of the initial shape functions.

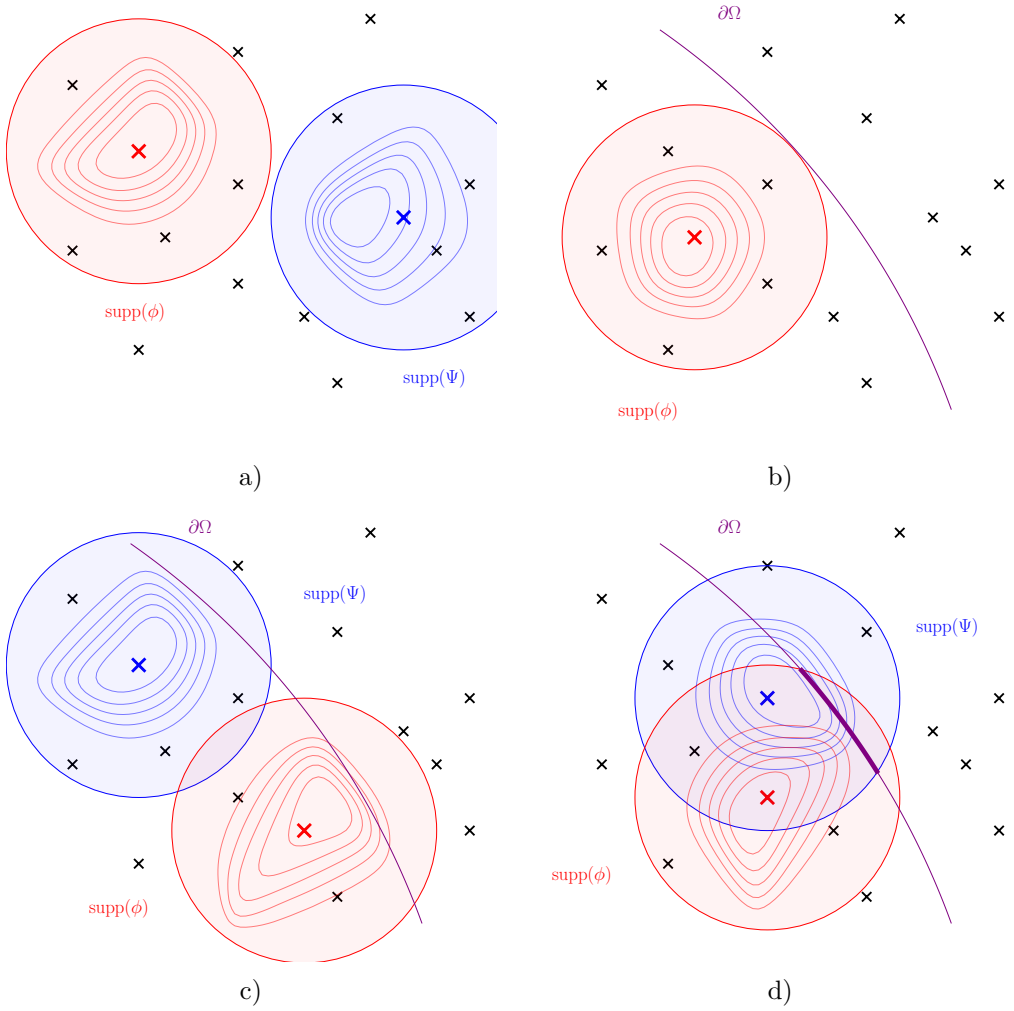


Figure 3.9 – *The intersection of the supports of shape functions give the graph of the SFEM operators. Several iso-values of the Shepard shape functions are represented above, both for the integration reconstruction and the advection reconstruction.*
a) The supports of the integration shape function ϕ and the advection shape function Ψ do not intersect, thus $\mathbf{A}_{red,blue} = \mathbf{0}$. In other words, (red, blue) is not an edge of the graph of $\mathbb{\nabla}^{VSFEM\ *}$. In particular, locality of the shape functions imply locality of $\mathbb{\nabla}^{VSFEM\ *}$.
b) The red node is not a boundary node since the intersection of $\mathrm{supp}(\phi)$ with the boundary $\partial\Omega$ has non-vanishing measure.
c) The support of the integration shape function ϕ intersects the boundary $\partial\Omega$, so the corresponding node is a left-sided boundary node. Similarly, the support of the advection shape function Ψ intersects the boundary, so the corresponding node is a right-sided boundary node.
d) The intersection of the supports of the two shape functions and the boundary is non-empty, thus $\mathbf{\Gamma}_{red,blue} = \mathbf{\Gamma}_{blue,red} \neq \mathbf{0}$.

Proposition 3.22 (Volume SFEM boundary integration and primal gradient operators)**.** Let us denote $< . >^{\Psi}$ the advection reconstruction operator, *i.e.* the operator defining the domain local dependency with respect to node movement (see section 3.2.1). The corresponding shape functions are denoted $\Psi_j : \mathbb{R}^d \to \mathrm{M}_d(\mathbb{R})$. As a consequence of the Reynolds transport theorem, the boundary operator reads: $\forall\, f : \mathcal{C} \to \mathbb{R}$, $\forall\, \mathbf{v} : \mathcal{C} \to \mathbb{R}^d$,

$$\oint_{\partial\mathcal{C}}^{VSFEM} f\mathbf{v} = \int_{\mathbf{x}\in\partial\Omega} < f > (\mathbf{x}) < \mathbf{v} >^{\Psi} (\mathbf{x}) \cdot \mathbf{n} \, \mathrm{d}S \tag{3.81}$$

In coordinate form, this reads: $\forall\, i, j \in \mathcal{C}$,

$$\mathbf{\Gamma}_{i,j}^{VSFEM} = \int_{\mathbf{x}\in\partial\Omega} \phi_i(\mathbf{x}) \Psi_j(\mathbf{x}) \cdot \mathbf{n} \, \mathrm{d}S \tag{3.82}$$

The graph of the boundary integration operator is interpreted on figure 2.7.

Remark. The simplest choice for the advection reconstruction operator is obviously $< . >^{\Psi} = < . >$, *i.e.* $\forall\, i \in \mathcal{C}$, $\forall\, \mathbf{x} \in \Omega$, $\Psi_i(\mathbf{x}) = \phi_i(\mathbf{x})\mathrm{I}_d$. With this choice, the boundary operator is symmetric and coincides with the classical SFEM boundary integration operator (see definition 2.26): $\forall\, i, j \in \mathcal{C}$,

$$\mathbf{\Gamma}_{i,j}^{VSFEM} = \mathbf{\Gamma}_{j,i}^{VSFEM} = \int_{\mathbf{x}\in\partial\Omega} \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) \mathbf{n} \, \mathrm{d}S$$

Proposition 3.23 (Expression and consistency of the volume SFEM primal gradient)**.** The corresponding primal gradient reads: $\forall\, \mathbf{v} : \mathcal{C} \to \mathbb{R}^d$,

$$V_i \mathbb{\nabla}_i^{VSFEM} \cdot \mathbf{v} = \sum_{j\in\mathcal{C}} \left(\int_{\mathbf{x}\in\Omega} \frac{\partial \phi_i}{\partial \mathbf{x}_j}(\mathbf{x}) \, \mathrm{d}V \right) \cdot \mathbf{v}_j + \int_{\mathbf{x}\in\partial\Omega} \phi_i(\mathbf{x}) < \mathbf{v} > (\mathbf{x}) \cdot \mathbf{n} \, \mathrm{d}S \tag{3.83}$$

Or in coordinate form: $\forall\, i, j \in \mathcal{C}$,

$$\mathbf{A}_{i,j}^{VSFEM} = \int_{\mathbf{x}\in\Omega} \frac{\partial \phi_i}{\partial \mathbf{x}_j}(\mathbf{x}) \, \mathrm{d}V + \int_{\mathbf{x}\in\partial\Omega} \phi_i(\mathbf{x}) \Psi_j(\mathbf{x}) \cdot \mathbf{n} \, \mathrm{d}S \tag{3.84}$$

As we have seen in section 3.1.4, we cannot expect affine covariance of the volume function equivalent to first order consistency of $\mathbb{\nabla}^{VSFEM}$ while retaining locality. In this case, the best one can expect is to achieve covariance with respect to isometric transformations. As a straightforward consequence of proposition 3.7, rigid motion covariance is achieved as long as the integration shape functions themselves are rigid motion covariant and the advection shape functions reproduce vector fields of the form $\mathbf{v}(\mathbf{x}) = \mathrm{A}\mathbf{x} + \mathbf{b}$ where $\mathrm{A} = -\mathrm{A}^T$.

Similarly to the classical SFEM case, the volume-based SFEM construction is mostly theoretical: there is for instance no known closed-form formula for the integration of meshless shape functions such as the MLS shape function, and it is doubtful that such formulae even exist. However, it is still possible to devise a mesh-based integration procedure that preserves the compatibility of the operators, as developed in section 3.7.2. Once again, this construction does not break the meshless curse (see conjecture 2.4) since the integration is performed using an underlying mesh.

3.7.2 A background mesh integration procedure for the volume-based SFEM operators

In this section, we develop a discrete integration procedure for the volume-based SFEM operators introduced in section 3.7.1 using a mesh. As we have seen in propositions 3.7 and

3.13, primal consistency is a consequence of covariance properties of the volume weights with respect to space transformations, whereas dual consistency is a consequence of the fact that the discrete integral of polynomials does not depend on the specific values of the positions of the discretization nodes. The challenge of a discrete integration procedure for the volume-based SFEM operator is to find a way to keep these properties from the continuous world inside the discrete world.

Let us thus suppose given an integration mesh denoted $\mathcal{M}$. We have seen in section 2.7.3 that the following discretization of the SFEM volume weights yields a first-order consistent volume integration operator on a point cloud: $\forall\, i \in \mathcal{C}$,

$$V_i^{VSFEM\ \mathcal{M}} \stackrel{\text{def}}{=} \sum_{\mathcal{C}^d \in \mathcal{M}} V(\mathcal{C}^d)\phi_i(\mathbf{x}(\mathcal{C}^d)) \tag{3.85}$$

Consistency does not appear out of thin air: rather, it is transferred from the reconstruction procedure $< . >$ to the volume integration operator $\oint_{\mathcal{C}}^{VSFEM}$ through another discrete volume integration operator (here, we used the mesh-based one point formula). Hence, if we wanted to transfer a higher consistency order to the mesh-integrated SFEM integration operator $\oint_{\mathcal{C}}^{VSFEM}$, we would need to use a discrete integration method with a higher consistency order.

Following the mantra of the volume-based meshless operator framework, the primal gradient is obtained by differentiation of the nodal volume weights considering that the underlying space is dragged along with the discretization nodes through the advection reconstruction $< . >^{\Psi}$. Here, the computational space is present in its discrete version: $\mathcal{M}$. Advection of the mesh is achieved through the advection of its nodes, we thus define: $\forall\, i, j \in \mathcal{C}$,

$$\begin{aligned}
\mathbf{A}_{i,j}^{VSFEM\ \mathcal{M}} &\stackrel{\text{def}}{=} \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_j} \sum_{\mathcal{C}^d \in \mathcal{M}} V(\mathcal{C}^d)\phi_i(\mathbf{x}(\mathcal{C}^d)) \\
&= \sum_{\mathcal{C}^d \in \mathcal{M}} V(\mathcal{C}^d) \frac{\partial \phi_i}{\partial \mathbf{x}_j}(\mathbf{x}(\mathcal{C}^d)) \\
&\quad + \sum_{\mathcal{C}^0 \in \partial \mathcal{C}^d} \Psi_j^T(\mathbf{x}(\mathcal{C}^0)) \cdot \left(\phi_i(\mathbf{x}(\mathcal{C}^d)) \frac{\partial V(\mathcal{C}^d)}{\partial \mathbf{x}(\mathcal{C}^0)} + V(\mathcal{C}^d) \frac{\partial \mathbf{x}(\mathcal{C}^d)}{\partial \mathbf{x}(\mathcal{C}^0)}^T \cdot \nabla \phi_i(\mathbf{x}(\mathcal{C}^d)) \right)
\end{aligned} \tag{3.86}$$

In the above expression, the derivatives of the volumes and centroids of cells of the mesh with respect to node movement appear. Since the advection field is very general, we cannot guarantee that planar p-cells of the mesh (*i.e.* cells of dimension p that are contained in an affine subspace of dimension p) are kept planar in the process. For this reason, we need a way to be able to compute volumes and centroids of cells of a mesh, as well as their derivatives with respect to node position in a very general setting. This task is carried on in section 1.2.

Proceeding similarly to the continuous case, a first choice for the discretization of the dual gradient is to recognize the first term of expression (3.86) as the opposite of the dual gradient operator. Equivalently, this means we use the same discrete integration on $\mathcal{M}$ as for volume weights, which gives: $\forall\, i, j \in \mathcal{C}$,

$$\mathbf{A}_{i,j}^{VSFEM\ \mathcal{M}\ *} = -\sum_{\mathcal{C}^d \in \mathcal{M}} V(\mathcal{C}^d) \frac{\partial \phi_j}{\partial \mathbf{x}_i}(\mathbf{x}(\mathcal{C}^d)) \tag{3.87}$$

A straightforward adaptation of the coordinate proof of proposition 3.20 given on page 111 shows that – as in the exactly integrated case – this definition converts the reproducibility

of a continuous function f by $< . >$ into exactness of ${}^{VSFEM}\mathbb{V}^{*}f$, without any restriction on f! This astonishing result however comes at a price: the resulting boundary integration operator is the second term of expression (3.86), which – unlike its name suggests – involves every vertex in the mesh. Consequently, we have $\Gamma_{i,j} \neq \mathbf{0}$ for nodes i and j of the point cloud arbitrarily far from the boundary.

In order to avoid this awkward behavior, we prefer to forsake the surprising possibility of dual consistency of arbitrarily high order, and instead include nodes that do not belong to the boundary of the mesh in the definition of the dual gradient: $\forall\, i, j \in \mathcal{C}$,

$$\begin{aligned}\mathbf{A}_{i,j}^{VSFEM\ \mathcal{M}\ *} \overset{\text{def}}{=} &- \sum_{\phi^d \in \mathcal{M}} V(\phi^d) \frac{\partial \phi_j}{\partial \mathbf{x}_i}(\mathbf{x}(\phi^d)) \\ &+ \sum_{\substack{\phi^0 \notin \phi^d \\ \phi^0 \in \partial \mathcal{M}}} \Psi_i^T(\mathbf{x}(\phi^0)) \cdot \left(\phi_j(\mathbf{x}(\phi^d)) \frac{\partial V(\phi^d)}{\partial \mathbf{x}(\phi^0)} + V(\phi^d) \frac{\partial \mathbf{x}(\phi^d)}{\partial \mathbf{x}(\phi^0)}^T \cdot \nabla \phi_j(\mathbf{x}(\phi^d)) \right)\end{aligned} \tag{3.88}$$

The first term of expression (3.88) is the consistent term of expression (3.87), while the second is a differential integration error that does not break consistency of maximal order $p = 1$. This order limitation would be pushed back if we were to use a discrete integration formula of higher order for the volume weights.

This choice ensures that the boundary integration operator vanishes for edges that are located far away from the boundary since the corresponding boundary integration coefficients read: $\forall\, i, j \in \mathcal{C}$,

$$\mathbf{\Gamma}_{i,j}^{VSFEM\ \mathcal{M}} = \sum_{\phi^d \in \mathcal{M}} \sum_{\substack{\phi^0 \in \partial \phi^d \\ \phi^0 \in \partial \mathcal{M}}} \Psi_j^T(\mathbf{x}(\phi^0)) \cdot \left(\phi_i(\mathbf{x}(\phi^d)) \frac{\partial V(\phi^d)}{\partial \mathbf{x}(\phi^0)} + V(\phi^d) \frac{\partial \mathbf{x}(\phi^d)}{\partial \mathbf{x}(\phi^0)}^T \cdot \nabla \phi_i(\mathbf{x}(\phi^d)) \right) \tag{3.89}$$

These definitions for the mesh-integrated volume-based SFEM operators allow us to state the following result:

Proposition 3.24 (Consistency of mesh-integrated volume-based SFEM operators)**.** The mesh-integrated volume-based SFEM operators whose coefficients are given in expressions (3.85), (3.86), (3.88) and (3.89) inherit all consistency properties of their exactly integrated analogs up to order $p = 1$. Moreover, the boundary cloud is restricted to nodes that are close to the boundary compared to the size of the supports of shape functions and the integration mesh.

Let us now compare the mesh-integrated classical and volume-based SFEM operators. First of all, as in the exactly integrated case, the corresponding volume integration operators are identical. Indeed, the only concern when discretizing the volume integration operator is to transfer first order consistency of the reconstruction operator via the mesh-based integration.

Secondly, even though the exactly integrated discrete boundary integration operators coincide (in the case $\Psi = \phi \mathrm{I}_d$), their two mesh-integrated equivalents differ. The reason for this difference is that the two methods choose different paths to exploit the topological structure of the mesh in order to pass on the properties of the exactly integrated discrete operators to their mesh-integrated analogues:

On the one hand, the mesh-integrated classical SFEM operators take advantage of the cell-face relation of the mesh using Gauss-type integration for the gradient – an approach reminiscent of FVM-type discretizations – and the oriented face vector areas $\mathbf{\Gamma}(\phi^{d-1})$. This approach makes the most of the chain complex structure of the mesh (see proposition 1.5):

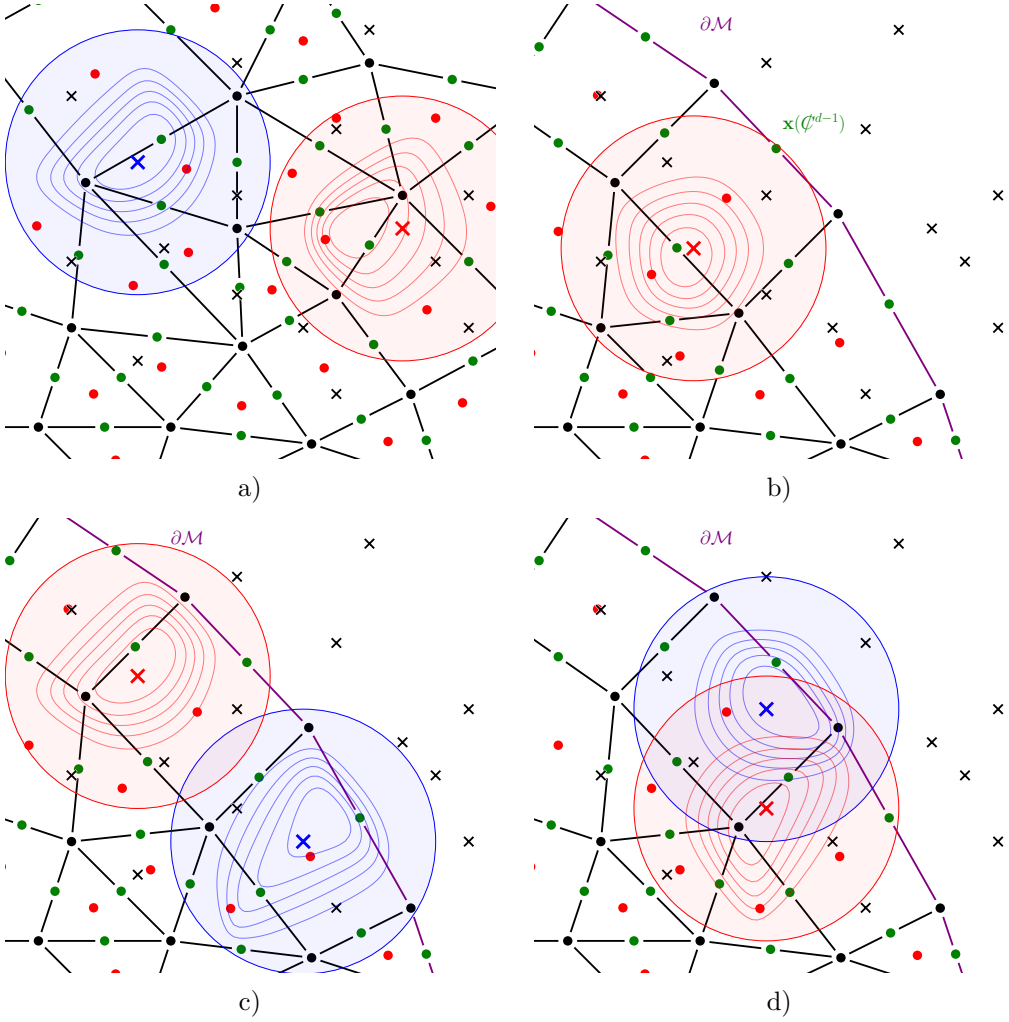


Figure 3.10 – *Discrete mesh-integrated volume-based SFEM operators can have a slightly different graph from their exact analogues (see figure 3.9), especially if the mesh $\mathcal{M}$ is coarse compared to the point cloud. On each subgraph, we represent in red the support of a reconstruction basis function ϕ, and in blue the support of an advection basis function Ψ.*
a) $\mathrm{supp}(\phi)$ and $\mathrm{supp}(\Psi)$ do not intersect. Still, there is a cell whose centroid is in $\mathrm{supp}(\phi)$ and which has a node in $\mathrm{supp}(\Psi)$. Hence, (red, blue) is an edge of the graph of $\mathbb{V}^{VSFEM}\ \mathcal{M}$.
b) Contrary to the same situation using the classical mesh-integrated SFEM operator (see figure 2.8 b)), the red node is not a boundary node.
c) Even though $\mathrm{supp}(\phi)$, $\mathrm{supp}(\Psi)$ and $\partial\mathcal{M}$ have an empty triple intersection, $\mathrm{supp}(\Psi)$ contains a node of $\partial\mathcal{M}$ which is in the boundary of a cell whose centroid is in $\mathrm{supp}(\phi)$. Consequently, (red, blue) is a boundary edge.
d) For the same reason, (red, blue) is an edge of the volume-based mesh-integrated SFEM boundary integration operator contrary to the classical mesh-integrated case (see figure 2.8 d)).

compatibility is ensured thanks to the closure property, its immediate geometric consequence, which certifies that faces of the boundary of a cell in a mesh adequately enclose the cell volume (see expression (1.31)).

On the other hand, the mesh-integrated volume-based SFEM operators capitalize on the fact that the formulae for the volumes and centroids of cells in a mesh developed in section 1.2 (in particular proposition 1.13) take their validity from the fact that the topological cell-tuple structure of the mesh retains its soundness for a wide range of node positions. In particular, the fact that p-cells are not assumed to be embedded in a linear subspace of dimension p lets us move the nodes of the mesh with an arbitrary velocity field, enabling an appropriate transfer of the volume-based methodology from the exact integration case to the discrete integration level. As a consequence, the relevant topological relation is no longer the cell-face relation, but the full cell-node relation through the computation of the derivatives of cell volumes and centroids with respect to node positions.

As a result of this discrete integration, the graphs of mesh-integrated volume-based SFEM operators are slightly different from the exactly integrated operators. For instance, (i,j) is an edge of the graph of $\mathbb{V}^{VSFEM\ \mathcal{M}}$ (*i.e.* $\mathbf{A}_{i,j}^{VSFEM\ \mathcal{M}} \neq \mathbf{0}$) if there is a cell $\phi^d \in \mathcal{M}$ with centroid $\mathbf{x}(\phi^d)$ in the support of ϕ_i and either with centroid in the support of Ψ_j as well, or with a node $\phi^0 \in \partial\phi^d$ in the support of Ψ_j. Figure 3.10 shows several examples of subtle configurations using the MLS reconstruction.

In order to complete the discretization of Neumann and Dirichlet boundary conditions using Nitsche's weak form (as presented in section 4.2.1), we need to give discrete versions of $\oiint_{\partial\mathcal{C}}^{VSFEM\ \mathcal{M}} f\mathbf{v}$ where one of the fields is discrete, and the other one is continuous. This discretization must coincide with $\oiint_{\partial\mathcal{C}} f\mathbf{v}\ \mathrm{d}S$ for linear fields, and above all, it needs to *only use values of the continuous function on* $\partial\Omega$. This last requirement is there because boundary condition fields (Dirichlet values and Neumann fluxes) are only defined on the boundary.

First, let us rewrite the formula for the vector boundary integration weights (expression (3.89)) as:

$$\mathbf{\Gamma}_{i,j}^{VSFEM\ \mathcal{M}} = \sum_{\phi^0 \in \partial\mathcal{M}} \Psi_j^T(\mathbf{x}(\phi^0)) \cdot \frac{\partial}{\partial \mathbf{x}(\phi^0)} \oiint_{\mathcal{M}} \phi_i \tag{3.90}$$

Hence, $\oiint_{\partial\mathcal{C}}^{VSFEM\ \mathcal{M}} f\mathbf{v}$ can be re-written as:

$$\oiint_{\partial\mathcal{C}}^{VSFEM\ \mathcal{M}} f\mathbf{v} = \sum_{\phi^0 \in \partial\mathcal{M}} <\mathbf{v}>^{\Psi}(\mathbf{x}(\phi^0)) \cdot \frac{\partial}{\partial \mathbf{x}(\phi^0)} \oiint_{\mathcal{C}}^{VSFEM\ \mathcal{M}} f \tag{3.91}$$

This suggests the following discretization if $\mathbf{v}$ is continuous and f is discrete:

$$\oiint_{\partial\mathcal{C}}^{VSFEM\ \mathcal{M}} f\mathbf{v} = \sum_{\phi^0 \in \partial\mathcal{M}} \mathbf{v}(\mathbf{x}(\phi^0)) \cdot \frac{\partial}{\partial \mathbf{x}(\phi^0)} \oiint_{\mathcal{C}}^{VSFEM\ \mathcal{M}} f \tag{3.92}$$

The above expression only involves evaluations of $\mathbf{v}$ at the boundary and coincides with expression (3.91) if $\mathbf{v}$ is reproduced by $<.>^{\Psi}$. If on the other hand f is continuous and $\mathbf{v}$ discrete, we need to find a linear consistent approximation of $\frac{\partial}{\partial \mathbf{x}(\phi^0)} \int_{\mathbf{x}\in\mathcal{M}} f(\mathbf{x})\ \mathrm{d}V$ that only uses values of f on the boundary. This highly non-trivial task requires an in-depth understanding of the geometry of polygonal meshes, and especially of non-planar cells. It is performed in section 1.3.2.3.

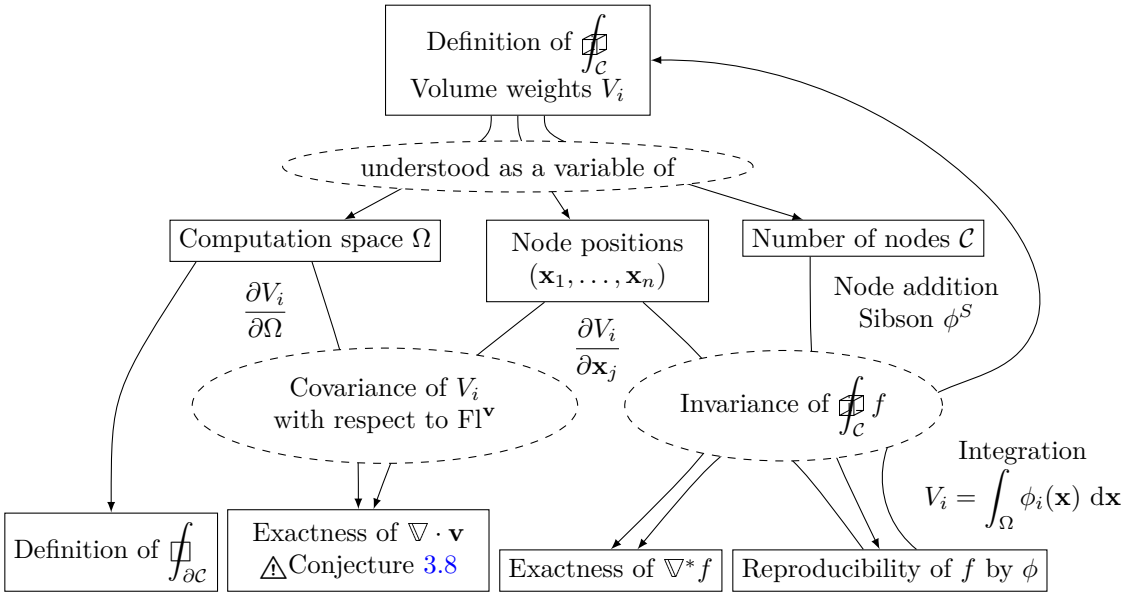


Figure 3.11 – *This diagram recapitulates the objects defined via the volume function through variations with respect to every possible variable, and links the desirable properties of the volume integration method with their consequences on the derived operators.*

A quick summary

In this chapter, we have presented a general method to exploit the dependency of volume integration weights with respect to its geometrical variables in order to generate derived primal and dual differential operators, a discrete boundary integration operator and nodal shape functions. We then related local and global consistency characterizations for these derived operators to properties of the initial volume weights, providing guidelines for the development of reliable discrete structures. The main definitions and theoretical results are summarized on figure 3.11.

We found that primal consistency at a given node was related to covariance of the corresponding volume weights with respect to transformations of space. Furthermore, we proved that first order consistency of the primal gradient could not be reconciled with locality.

We then proposed to decouple the different contributions of discretization nodes and computational space: considering variations of discrete volume weights with respect to the computational space only allowed us to define a boundary integration operator and to describe the corresponding dual gradient. This is the first time that a similar treatment has been given in the literature. In contrast to its primal counterpart, we showed that dual consistency at a given node was related to the invariance of the value of the integral as this node is moved, and that this notion was equivalent with the reproducibility property of Sibson shape functions. With this idea in mind, we could re-interpret classical stabilization procedures in SPH as actually trying to find a particular set of node positions that maximizes the discrete total volume, making it locally invariant, thus locally achieving dual consistency.

These theoretical results were highlighted with the examples of the SPH and volume-

based SFEM operators. In particular, we developed a simple discrete integration procedure allowing the computation of compatible linear consistent nodal meshless operators.

Chapter 4

Application to the discretization of elliptic partial differential equations

Contents

In this chapter, we use the meshless framework and operators developed in earlier chapters to solve systems of elliptic differential equations.

In section 4.1, we recall two continuous formulations of elliptic PDEs: the classical weak form and the symmetric Nitsche weak form. Since the two formulations only differ in their way to handle boundary conditions, their nodal discretization have one central patch test necessary conditions: compatibility. We verify throughout this chapter that compatibility – or at least approximate compatibility – is indeed necessary for optimal convergence.

On the one hand, the Nitsche formulation imposes Dirichlet boundary conditions weakly.

Even though it is somewhat cumbersome with its numerous added boundary terms and the fact that it requires a specific boundary stabilization to ensure coercivity, its weak nodal discretization is almost immediate in our meshless framework once every discrete operator is available.

In section 4.2, we compare the performance of the classical and the volume-based SFEM operators on this formulation. We realize that the classical formulation is less prone to under-integration issues, but its volume-based alternative does not require any additional bulk stabilization. We then give numerical evidence that these advantages can be combined in an element-free Galerkin discretization with non-nodal integration.

In section 4.3 on the other hand, we investigate the possibility of handling Dirichlet boundary conditions in an essential way. A naive first proposal necessitates the positioning of boundary nodes exactly on the boundary of the computational domain. In our effort to circumvent this inconvenient restriction, we augment our meshless framework with a general method to build a discrete trace operator and H_0^1 Sobolev spaces and treat the Dirichlet boundary conditions essentially. Provided adequate discrete operators can be built, we give numerical evidence that this new formulation performs equally well as a linear finite element method.

Finally, these two ways of handling boundary conditions are compared on a common linear elasticity test case.

4.1 Several continuous formulations of elliptic partial differential equations

In this section, we detail the several continuous formulations of elliptic equations. See Gilbarg and Trudinger [103] for a thorough theoretical treatment of the Poisson problem.

4.1.1 The scalar diffusion equation

Let Ω be a compact subset of $\mathbb{R}^d$ with non-empty interior and regular enough boundary $\partial\Omega$. The boundary is split into two disjoint sets: $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ where $\partial\Omega_D \cap \partial\Omega_N = \varnothing$ and $\partial\Omega_D \neq \varnothing$. Let $\mathrm{D} : \Omega \to \mathrm{S}_d^{++}(\mathbb{R})$, $s : \Omega \to \mathbb{R}$, $g : \partial\Omega_N \to \mathbb{R}$ and $u_D : \partial\Omega_D \to \mathbb{R}$ respectively denote a regular positive definite diffusion tensor field on Ω, an integrable source field on Ω, a Neumann source field on $\partial\Omega_N$ and Dirichlet boundary values on $\partial\Omega_D$ (see [88] for a detailed account of the influence of the regularity of the boundary and the different fields on the analysis).

The strong form of the Poisson equation reads: Find $u \in C^2(\Omega)$ such that:

$$\begin{cases} -\nabla \cdot \mathrm{D} \cdot \nabla u = s & \\ u = u_D & \text{on } \partial\Omega_D \\ \nabla u \cdot \mathrm{D} \cdot \mathbf{n} = g & \text{on } \partial\Omega_N \end{cases} \tag{4.1}$$

In the case $\mathrm{D} = \mathrm{I}_d$, the Left Hand Side (LHS) of the bulk equation is the opposite of the Laplace operator $\Delta \overset{\text{def}}{=} \nabla \cdot \nabla$.

The usual weak form of the Poisson equation treats Dirichlet boundary conditions in an essential way, *i.e.* with the introduction of $H^1_{0,D}(\Omega)$, the Sobolev space of weakly derivable functions with vanishing trace on the Dirichlet boundary $\partial\Omega_D$. It reads: Find $u \in H^1(\Omega)$ such that for all $v \in H^1_{0,D}(\Omega)$, we have:

$$\begin{cases} \displaystyle\int_\Omega \nabla u \cdot \mathrm{D} \cdot \nabla v \, \mathrm{d}V = \int_\Omega sv \, \mathrm{d}V + \int_{\partial\Omega_N} gv \, \mathrm{d}S \\ u = u_D \qquad\qquad \text{on } \partial\Omega_D \end{cases} \tag{4.2}$$

It is also possible to treat Dirichlet boundary conditions in a natural way, *i.e.* with a weak form that does not explicitly mention the trace operator and does not need the definition of H^1_0 spaces. For instance, the symmetric Nitsche's weak form [97, 175] reads:

$$\begin{aligned} \int_\Omega \nabla u \cdot \mathrm{D} \cdot \nabla v \, \mathrm{d}V - \int_{\partial\Omega_D} (\nabla u \cdot \mathrm{D} \cdot \mathbf{n})\, v \, \mathrm{d}S - \int_{\partial\Omega_D} u \nabla v \cdot \mathrm{D} \cdot \mathbf{n} \, \mathrm{d}S + a_{\partial\Omega_D \text{ stab}}(u, v) \\ = \int_\Omega sv \, \mathrm{d}V + \int_{\partial\Omega_N} gv \, \mathrm{d}S - \int_{\partial\Omega_D} u_D \nabla v \cdot \mathrm{D} \cdot \mathbf{n} \, \mathrm{d}S + a_{\partial\Omega_D \text{ stab}}(u_D, v) \end{aligned} \tag{4.3}$$

The stabilization bilinear form $a_{\partial\Omega_D \text{ stab}}$ is devised so that the formulation is coercive. The classical choice, which we also make here consists in defining:

$$a_{\partial\Omega_D \text{ stab}}(u, v) \overset{\text{def}}{=} \gamma \int_{\partial\Omega_D} uv \, \mathrm{d}S \tag{4.4}$$

In the above expression, γ denotes a positive scalar penalization factor. It should be large enough to ensure coercivity of the formulation, but very large values are not desirable in practice as they lead to ill-conditioned discretizations. A compromise thus needs to be found on a case-by-case basis.

4.1.2 Derivation of the weak forms of the scalar diffusion equation

The computation here is only formal and is not concerned with regularity of the functions. Starting from $-\nabla \cdot \mathrm{D} \cdot \nabla u = s$, we multiply both sides by any test function v and integrate on Ω, yielding: $\forall\, v$,

$$-\int_\Omega (\nabla \cdot \mathrm{D} \cdot \nabla u)\, v \, \mathrm{d}V = \int_\Omega sv \, \mathrm{d}V$$

Integrating the LHS of the above expression by parts, we have:

$$\int_\Omega \nabla u \cdot \mathrm{D} \cdot \nabla v \, \mathrm{d}V - \int_{\partial\Omega} (\nabla u \cdot \mathrm{D} \cdot \mathbf{n})\, v \, \mathrm{d}S = \int_\Omega sv \, \mathrm{d}V$$

Using additivity of integration on disjoints subsets, the boundary integral can be split into a Dirichlet part on $\partial\Omega_D$ and a Neumann part on $\partial\Omega_N$. The value of $\mathrm{D} \cdot \nabla u \cdot \mathbf{n}$ is replaced with the Neumann flux field g on the Neumann boundary $\partial\Omega_N$ and transferred to the RHS, yielding:

$$\int_\Omega \nabla u \cdot \mathrm{D} \cdot \nabla v \, \mathrm{d}V - \int_{\partial\Omega_D} (\nabla u \cdot \mathrm{D} \cdot \mathbf{n})\, v \, \mathrm{d}S = \int_\Omega sv \, \mathrm{d}V + \int_{\partial\Omega_N} gv \, \mathrm{d}S \tag{4.5}$$

The derivations of the usual and symmetric Nitsche weak forms bifurcate here. On the one hand, if we make the essential choice, we restrict ourselves to test functions which vanish on the Dirichlet boundary $\partial\Omega_D$, hence the term $\int_{\partial\Omega_D} v \nabla u \cdot \mathrm{D} \cdot \mathbf{n} \, \mathrm{d}S$ vanishes. This gives:

$$\int_\Omega \nabla u \cdot \mathrm{D} \cdot \nabla v \, \mathrm{d}V = \int_\Omega sv \, \mathrm{d}V - \int_{\partial\Omega_N} gv \, \mathrm{d}S$$

Which is nothing but the first equation of (4.2).

On the other hand, starting back from equation (4.5), we can restore symmetry with the addition of a transposed Dirichlet boundary term on both sides of the equation. The value

of the solution is replaced in the RHS with its imposed value, which yields:

$$\int_\Omega \nabla u \cdot \mathrm{D} \cdot \nabla v \, \mathrm{d}V - \int_{\partial\Omega_D} (\nabla u \cdot \mathrm{D} \cdot \mathbf{n})\, v \, \mathrm{d}S - \int_{\partial\Omega_D} u \nabla v \cdot \mathrm{D} \cdot \mathbf{n} \, \mathrm{d}S = \int_\Omega sv \, \mathrm{d}V + \int_{\partial\Omega_N} gv \, \mathrm{d}S - \int_{\partial\Omega_D} u_D \nabla v \cdot \mathrm{D} \cdot \mathbf{n} \, \mathrm{d}S$$

Finally, adding a stabilization term on the Dirichlet boundary gives the Nitsche weak formulation:

$$\int_\Omega \nabla u \cdot \mathrm{D} \cdot \nabla v \, \mathrm{d}V - \int_{\partial\Omega_D} (\nabla u \cdot \mathrm{D} \cdot \mathbf{n})\, v \, \mathrm{d}S - \int_{\partial\Omega_D} u \nabla v \cdot \mathrm{D} \cdot \mathbf{n} \, \mathrm{d}S + a_{\partial\Omega_D \text{ stab}}(u, v) = \int_\Omega sv \, \mathrm{d}V + \int_{\partial\Omega_N} gv \, \mathrm{d}S - \int_{\partial\Omega_D} u_D \nabla v \cdot \mathrm{D} \cdot \mathbf{n} \, \mathrm{d}S + a_{\partial\Omega_D \text{ stab}}(u_D, v)$$

4.1.3 Linear elasticity: a vector elliptic equation

We now detail different formulations of the equations of linear elasticity. These equations are extremely similar to their scalar counterpart given in section 4.1.1. The only significant change is the possibility to have different types of boundary conditions at the same boundary location for the normal and tangential components of the unknown or boundary stress. Physically speaking, this for instance allows the modeling of roller boundary conditions, but also symmetry and anti-symmetry planes.

More precisely, the boundary of the domain is supposed split twice into non-overlapping regular domains $\partial\Omega = \partial\Omega_D^\perp \cup \partial\Omega_N^\perp = \partial\Omega_D^\shortparallel \cup \partial\Omega_N^\shortparallel$ where $\partial\Omega_D^\perp \cap \partial\Omega_N^\perp = \partial\Omega_D^\shortparallel \cap \partial\Omega_N^\shortparallel = \varnothing$, appropriate boundary fields $u_D^\perp$,$g^\perp$,$\mathbf{u}_D^\shortparallel$ and $\mathbf{g}^\shortparallel$ and interior fields $\mathbb{H}$ and $\mathbf{s}$ given.

The orthogonal projection on the normal space is denoted $\mathrm{P}^\perp = \mathbf{n}\otimes\mathbf{n}$ and the orthogonal projection on the tangential space is denoted $\mathrm{P}^\shortparallel = \mathrm{I}_d - \mathrm{P}^\perp$.

Defining $\mathcal{E}\mathbf{v} \overset{\text{def}}{=} (\nabla\mathbf{v} + \nabla\mathbf{v}^T)/2$, the strong form of the elasticity equations reads: Find $\mathbf{u} \in C^{1,d}(\Omega)$ such that:

$$\begin{cases} -\nabla \cdot (\mathbb{H} : \mathcal{E}\mathbf{u}) = \mathbf{s} & \\ \mathbf{u} \cdot \mathbf{n} = u_D^\perp & \text{on } \partial\Omega_D^\perp \\ \mathrm{P}^\shortparallel \cdot \mathbf{u} = \mathbf{u}_D^\shortparallel & \text{on } \partial\Omega_D^\shortparallel \\ ((\mathbb{H} : \mathcal{E}\mathbf{u}) \cdot \mathbf{n}) \cdot \mathbf{n} = g^\perp & \text{on } \partial\Omega_N^\perp \\ \mathrm{P}^\shortparallel ((\mathbb{H} : \mathcal{E}\mathbf{u}) \cdot \mathbf{n}) = \mathbf{g}^\shortparallel & \text{on } \partial\Omega_N^\shortparallel \end{cases} \tag{4.6}$$

The standard weak form of the elasticity equations read: Find $\mathbf{u} \in H^{1,d}(\Omega)$ such that for all $\mathbf{v} \in H^{1,d}_{0,D^\perp,D^\shortparallel}(\Omega)$ (the space of square integrable vector valued functions on Ω with square integrable derivatives and whose trace has a vanishing orthogonal (resp. parallel) component on $\partial\Omega_D^\perp$ (resp. $\partial\Omega_D^\shortparallel$)), we have:

$$\begin{cases} \displaystyle\int_\Omega \mathcal{E}\mathbf{u} : \mathbb{H} : \mathcal{E}\mathbf{v} \, \mathrm{d}V = \int_\Omega \mathbf{s} \cdot \mathbf{v} \, \mathrm{d}V + \int_{\partial\Omega_N^\perp} g^\perp \mathbf{n} \cdot \mathbf{v} \, \mathrm{d}S + \int_{\partial\Omega_N^\shortparallel} \mathbf{g}^\shortparallel \cdot \mathbf{v} \, \mathrm{d}S & \\ \mathbf{u} \cdot \mathbf{n} = u_D^\perp \quad \text{on } \partial\Omega_D^\perp & \\ \mathrm{P}^\shortparallel \cdot \mathbf{u} = \mathbf{u}_D^\shortparallel \quad \text{on } \partial\Omega_D^\shortparallel & \end{cases} \tag{4.7}$$

The corresponding Nitsche weak form reads:

$$\begin{aligned}
&\int_\Omega \mathcal{E}\mathbf{u} : \mathbb{H} : \mathcal{E}\mathbf{v} \, \mathrm{d}V - \int_{\partial\Omega_D^\perp} \mathbf{n} \cdot (\mathbb{H} : \mathcal{E}\mathbf{u}) \cdot \mathrm{P}^\perp \cdot \mathbf{v} \, \mathrm{d}S - \int_{\partial\Omega_D^{\shortparallel}} \mathbf{n} \cdot (\mathbb{H} : \mathcal{E}\mathbf{u}) \cdot \mathrm{P}^{\shortparallel} \cdot \mathbf{v} \, \mathrm{d}S \\
&- \int_{\partial\Omega_D^\perp} \mathbf{u} \cdot \mathrm{P}^\perp \cdot (\mathbb{H} : \mathcal{E}\mathbf{v}) \cdot \mathbf{n} \, \mathrm{d}S - \int_{\partial\Omega_D^{\shortparallel}} \mathbf{u} \cdot \mathrm{P}^{\shortparallel} \cdot (\mathbb{H} : \mathcal{E}\mathbf{v}) \cdot \mathbf{n} \, \mathrm{d}S \\
&+ a_{\partial\Omega_D^\perp \text{ stab}}(\mathbf{u}, \mathbf{v}) + a_{\partial\Omega_D^{\shortparallel} \text{ stab}}(\mathbf{u}, \mathbf{v}) \qquad (4.8) \\
&= \int_\Omega \mathbf{s} \cdot \mathbf{v} \, \mathrm{d}V + \int_{\partial\Omega_N^\perp} g^\perp \mathbf{n} \cdot \mathbf{v} \, \mathrm{d}S + \int_{\partial\Omega_N^{\shortparallel}} \mathbf{g}^{\shortparallel} \cdot \mathbf{v} \, \mathrm{d}S \\
&- \int_{\partial\Omega_D^\perp} \mathbf{u}_D \cdot \mathrm{P}^\perp \cdot (\mathbb{H} : \mathcal{E}\mathbf{v}) \cdot \mathbf{n} \, \mathrm{d}S - \int_{\partial\Omega_D^{\shortparallel}} \mathbf{u}_D \cdot \mathrm{P}^{\shortparallel} \cdot (\mathbb{H} : \mathcal{E}\mathbf{v}) \cdot \mathbf{n} \, \mathrm{d}S \\
&+ a_{\partial\Omega_D^\perp \text{ stab}}(u_D^\perp \mathbf{n}, \mathbf{v}) + a_{\partial\Omega_D^{\shortparallel} \text{ stab}}(\mathbf{u}_D^{\shortparallel}, \mathbf{v})
\end{aligned}$$

Where the stabilization forms are usually defined as: $\forall \, \square \in \{\perp, \shortparallel\}$,

$$a_{\partial\Omega_D^\square \text{ stab}}(\mathbf{u}, \mathbf{v}) \overset{\text{def}}{=} \gamma^\square \int_{\partial\Omega_D^\square} \mathbf{u} \cdot \mathrm{P}^\square \cdot \mathbf{v} \, \mathrm{d}S \qquad (4.9)$$

In the sections 4.2 and 4.3, we develop meshless discretization methods based on the previously detailed continuous formulations.

4.1.4 Discretization principles and a common test case

The governing principle behind the elaboration of discrete formulations is the following: we translate one of the many forms of the continuous equation into a discrete set of equations using the tools introduced in chapter 2. In most cases, we bluntly replace the continuous operators with their discrete counterparts. The main difficulty of this step is to take the boundary conditions – especially the essential boundary conditions – into account while imposing as few constraints on the point cloud as possible. This is the first major hurdle when designing meshless methods for the discretization of PDEs.

Then we analyze the properties of the resulting discrete problem by means of an altogether widely celebrated (see [217, 224]) and highly criticized (see [85, 218]) criterion: the patch test. In a nutshell, we want to make sure that in cases where the exact solution is a linear field, it is exactly captured by the discretization. This analysis always reveals the necessity for the discrete structure to verify some kind of relation between the discrete integration and differentiation, which takes a form reminiscent of Stokes's theorem (see expression (1.9)). In most cases, these requirements are exactly the compatibility conditions of definition 2.18, justifying our intense search for compatible sets of discrete operators.

The quality of the different discretizations is assessed on a common two-dimensional ($d = 2$) smooth analytical test-case with simple square geometry ($\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$) with unit scalar diffusivity $\mathrm{D} = \mathrm{I}_d$. Source and boundary fields are chosen so that the exact manufactured solution reads: $\forall \, \mathbf{x} \in \Omega$,

$$u^{\text{ex}}(\mathbf{x}) = \prod_{i=1}^{d} \sin(k_i x_i) \qquad (4.10)$$

We choose $\mathbf{k} = 2\pi \begin{pmatrix} 2 \\ 4 \end{pmatrix}$. The results are then compared based on the following relative L^2 discrete error:

$$\mathrm{Err}^2_{L^2(\mathcal{C})} \overset{\text{def}}{=} \frac{\int\!\!\!\!\!\!\diagup_{\mathcal{C}} (u^{\text{ex}} - u^{\text{num}})^2}{\int\!\!\!\!\!\!\diagup_{\mathcal{C}} u^{\text{ex}\ 2}} \qquad (4.11)$$

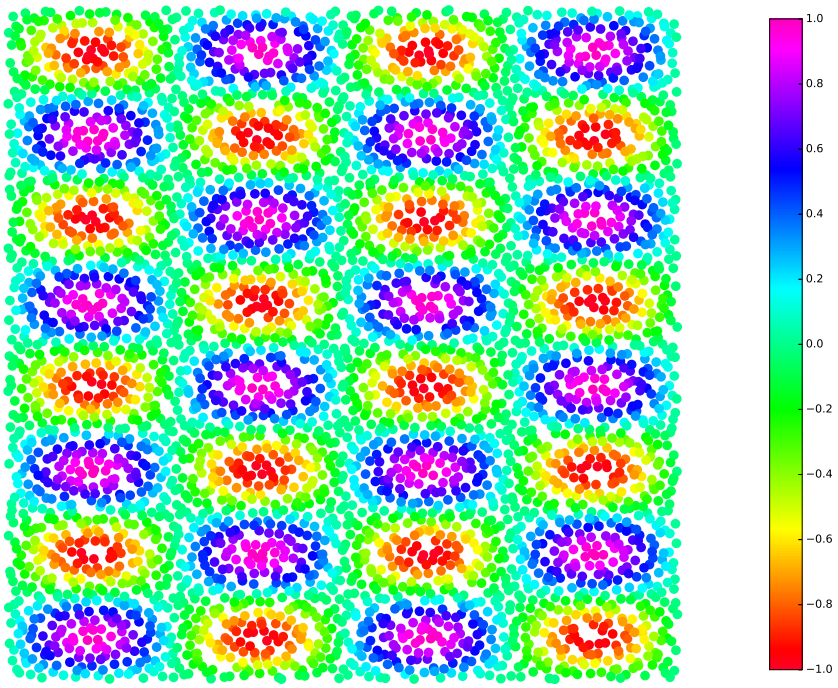


Figure 4.1 – *Exact analytical solution of the diffusion test case on a Halton point cloud.*

This exact solution is represented on a Halton point cloud with 4900 nodes on figure 4.1.

4.2 Meshless discretizations of the Nitsche weak form

In this section, we build discretizations starting from the Nitsche weak form of elliptic equations (see expression (4.3) in the scalar case). In section 4.2.1, we brutally substitute every continuous operator for its discrete analogue to form nodal operator-based discretizations, and in section 4.2.2, we detail meshless Bubnov-Galerkin discretizations.

4.2.1 Nodal operator-based discrete formulation

In this section, we give a nodal operator-based discretization of the Nitsche formulation of the diffusion equation.. The discretization process is simply the following: start from the continuous Nitsche weak form (expression (4.3)) and replace every continuous operator by their discrete analogue defined in chapter 2. More precisely, we give the following definitions:

Definition 4.1 (Nitsche discretization of the Poisson problem)**.**
Let $\mathrm{D} : \Omega \to \mathrm{S}_d^+(\mathbb{R})$ be a diffusion tensor field, $s : \Omega \to \mathbb{R}$ a scalar source field, $u_D : \partial\Omega_D \to \mathbb{R}$ and $g : \partial\Omega_N \to \mathbb{R}$ Dirichlet boundary values and Neumann boundary fluxes. Then, the nodal meshless discretization of the Nitsche weak form reads: Find $u : \mathcal{C} \to \mathbb{R}$ such that for all $v : \mathcal{C} \to \mathbb{R}$, we have:

$$\fint_{\mathcal{C}} \mathbb{\nabla} u \cdot \mathrm{D} \cdot \mathbb{\nabla} v - \fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} u) v - \fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} v) u + a_{\partial \mathcal{C}_D \text{ stab}}(u, v) \tag{4.12}$$
$$= \fint_{\mathcal{C}} s v + \fint_{\partial \mathcal{C}_N} g v - \fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} v) u_D + a_{\partial \mathcal{C}_D \text{ stab}}(u_D, v)$$

We insist on the fact that the order of the fields under a discrete boundary integration is important. The right order is the one given in the above expression. The resulting discretization is symmetric.

In expression (4.12), we use the notation $\fint$ for three distinct meanings:

- The terms $\fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} u) v$ and $\fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} v) u$ use the regular bilinear boundary integration operator of definition 2.7 since both $\mathrm{D} \cdot \mathbb{\nabla} u$ and v are discrete fields.
- The term $\fint_{\partial \mathcal{C}_N} g v$ contains both a continuous field g (on the left) defined on $\partial \Omega_N$ only and a discrete field v (on the right). Discretization of this kind of terms as a linear form acting on discrete fields is briefly mentioned in sections 2.2.1, 2.7.3 and 3.7.2.
- Symmetrically, the term $\fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} v) u_D$ uses the other linear form, with the continuous field u_D on the right, and the discrete field $\mathrm{D} \cdot \mathbb{\nabla} v$ on the left.

Remark. In fact, the boundary integration operators need to be restricted to integrate on $\partial \Omega_N$ and $\partial \Omega_D$ respectively. The specifics on how to proceed vary with the set of discrete operators and the representation of the boundary.

In the case of the classical mesh-integrated SFEM operators, we have seen that the boundary is represented by boundary faces, and a restriction of the sum in expression (2.82) is a good choice.

The case of the volume-based mesh-integrated SFEM operators is slightly different, as we have seen that the relevant object to represent the boundary are its nodes. In order to stay consistent and not attribute regions of the boundary twice, the restriction of the sum in expression (3.92) to Neumann boundary nodes entails that expression (1.42) should be restricted to Dirichlet boundary nodes.

Proposition 4.1 (Linear patch test conditions for the nodal Nitsche weak form)**.**
Let $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Let us consider the discrete system defined above with a uniform diffusion tensor field, vanishing source field and boundary conditions given by $u_D = \mathbf{a} \cdot \mathbf{x} + b$ and $g = \mathbf{a} \cdot \mathrm{D} \cdot \mathbf{n}$. If the following conditions are met, then the nodal Nitsche weak discretization of the Poisson problem admits the affine field $u = \mathbf{a} \cdot \mathbf{x} + b$ as a solution:

- The gradient operator is first order consistent.
- $\left(\fint_{\mathcal{C}}, \fint_{\partial \mathcal{C}}, \mathbb{\nabla} \right)$ form a compatible set of discrete operators.
- If the left field in the bi-linear boundary is constant, then it reduces to the corresponding linear boundary integration operator.
- If the right field in the bi-linear boundary is affine, then it reduces to the corresponding linear boundary integration operator.

Proof. We replace the boundary conditions by their value and u by $\mathbf{a} \cdot \mathbf{x} + b$ in expression (4.12). Using the first order consistency of $\mathbb{\nabla}$, the system becomes: $\forall\, v : \mathcal{C} \to \mathbb{R}$,

$$\fint_{\mathcal{C}} \mathbf{a} \cdot \mathrm{D} \cdot \mathbb{\nabla} v - \fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbf{a}) v - \fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} v)(\mathbf{a} \cdot \mathbf{x} + b) + a_{\partial \mathcal{C}_D \text{ stab}}(\mathbf{a} \cdot \mathbf{x} + b, v) \tag{4.13}$$
$$= \fint_{\partial \mathcal{C}_N} (\mathrm{D} \cdot \mathbf{a}) v - \fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} v)(\mathbf{a} \cdot \mathbf{x} + b) + a_{\partial \mathcal{C}_D \text{ stab}}(\mathbf{a} \cdot \mathbf{x} + b, v)$$

Both the boundary stabilization terms $a_{\partial \mathcal{C}_D \text{ stab}}(\mathbf{a} \cdot \mathbf{x} + b, v)$ and the symmetric Nitsche consistent terms $\fint_{\partial \mathcal{C}_D} (\mathrm{D} \cdot \mathbb{\nabla} v)(\mathbf{a} \cdot \mathbf{x} + b)$ in the LHS and the RHS cancel each other out. The

equation can now be re-written as:

$$\oint\!\!\!\!\!\!\square_{\mathcal{C}} (\mathrm{D}\cdot\mathbf{a})\cdot\mathbb{\nabla} v = \oint\!\!\!\!\!\!\square_{\partial\mathcal{C}} (\mathrm{D}\cdot\mathbf{a}) v \tag{4.14}$$

Since the field $\mathrm{D}\cdot\mathbf{a}$ is uniform, the above expression is exactly the compatibility of the discrete operators (see definition 2.18). □

Convergence study

Let us now compare the results of this formulation using the classical mesh-integrated SFEM set of discrete meshless operators defined in section 2.7.3 and their volume-based analogues of section 3.7.2 using the MLS shape functions with a linear reconstruction basis. Both these sets of operators satisfy the linear patch test sufficient conditions of proposition 4.1. Let us note that both sets of discrete operators use the same volume integration weights, and consequently the cloud-integrated errors (see expression 4.11) are immediately comparable.

The test-case is the diffusion problem with exact solution given in expression (4.10), with $\mathbf{k} = 2\pi\,(2,4)$. The top half of the boundary is treated with Neumann boundary conditions (*i.e.* $\partial\Omega_N = \{\mathbf{x}\in\partial\Omega \mid x_2 \geq \frac{1}{2}\}$ and the bottom half is treated with Dirichlet boundary conditions. The Nitsche boundary stabilization term $a_{\partial\mathcal{C}_D\ \mathrm{stab}}$ is given by the mesh-integrated discretization of expression (4.4) using expression (2.79) in the LHS and expression (4.24) in the RHS. The boundary stabilization γ is chosen just high enough so that the formulation is stable but no too high as to avoid ill-conditioning in accordance with recommended usage. Specifically, we used the value $\gamma = \frac{1\times 10^1}{h}$ for this test-case. The smoothing length is set as $4\sqrt{\frac{V(\Omega)}{\pi n_n}}$, so that the volumes of the supports of the MLS shape functions decrease linearly with the number of particles.

On the top row of figure 4.2, we compare the cloud-integrated error of the Nitsche formulation for both classical and volume-based mesh integrated SFEM operators.

The classical SFEM formulation gives extremely poor results for all meshes and point clouds, the error level is almost always $> 100\%$. The volume-based formulation has a very different behavior: if the background integration mesh is fine enough, then we have a second order convergence as the point cloud is refined (the measured rate is $h^{1.9}$). But if the integration mesh is too coarse compared to the point cloud, our linear solver (an in-house deflated GMRES solver) simply cannot solve the corresponding equations, and the error is obviously very high.

In order to improve the behavior of the simulation, especially for the classical SFEM case, we take full advantage of the fact that the MLS reconstruction does not satisfy the delta property (see definition 2.11) and define the following stabilization term. Notice that it is completely independent of the mesh.

$$a_{\mathrm{stab}}(u,v) = \sum_{i\in\mathcal{C}} w_i (u_i - < u > (\mathbf{x}_i))(v_i - < v > (\mathbf{x}_i)) \tag{4.15}$$

This stabilization term is symmetric and non-negative provided the nodal stabilization weights w_i are non-negative. It vanishes when u is a reproduced function, thus does not change the patch test conditions since we use linear MLS shape functions. The right scaling for the stabilization weights is $w_i = \delta\frac{V_i}{h^2}$. The error on the test case using this stabilization is given on the medium and bottom row of figure 4.2.

For a low value of the stabilization factor ($\delta = 1\times 10^{-3}$), the convergence profile of the simulation using the classical SFEM operators with a given mesh is clearly separated in

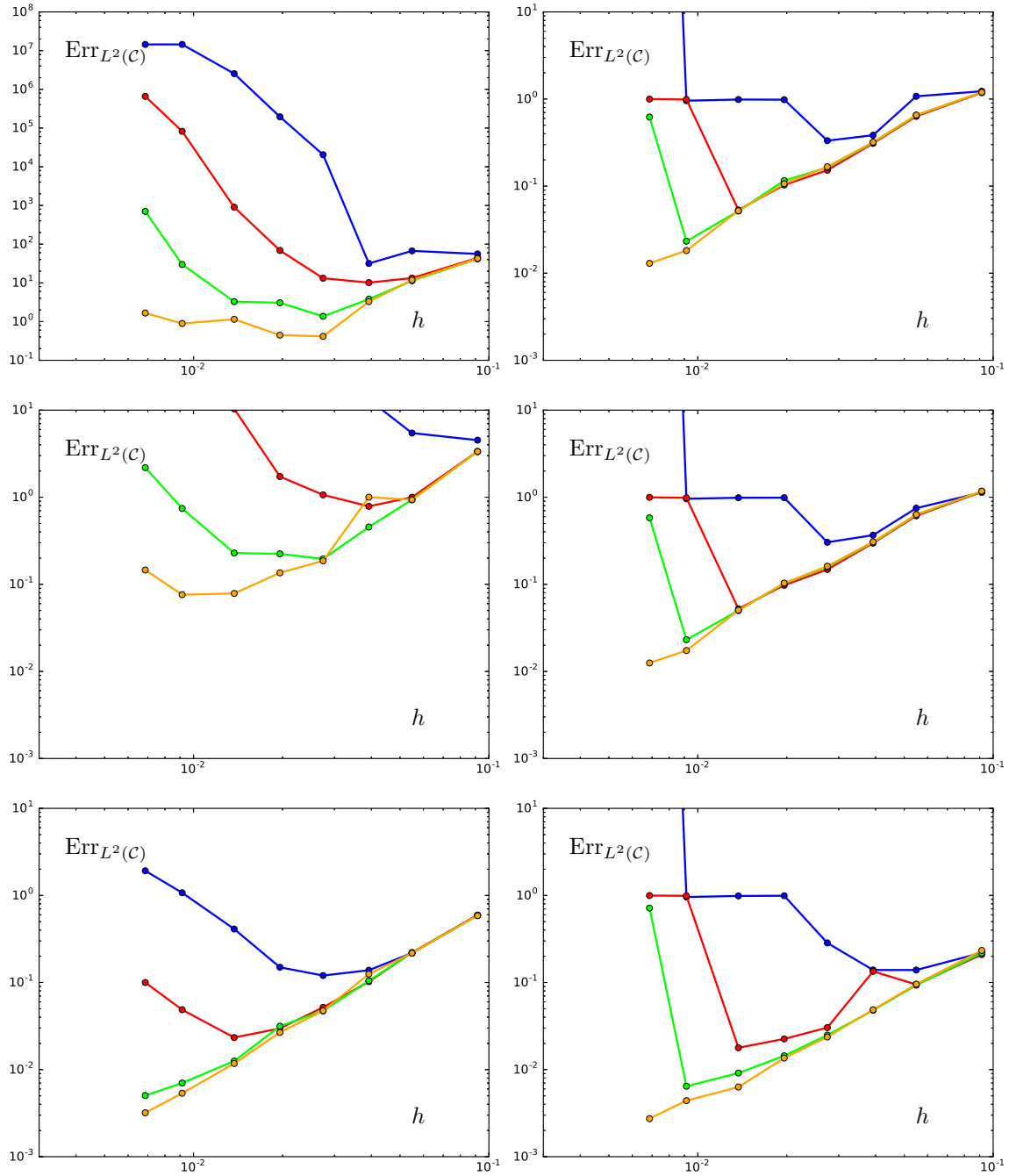


Figure 4.2 – *Relative error on the nodal meshless discretization of the Nitsche weak form of the diffusion equation as a function of the smoothing length. Integration is performed on a mesh made of 2574 cells, 11480 cells, 47492 cells and 124422 cells.*
Top row: unstabilized simulation $\delta = 0$. ⚠: the scale is different on the top left plot.
Middle row: low stabilization factor $\delta = 1 \times 10^{-3}$.
Bottom row: medium stabilization factor $\delta = 2 \times 10^{0}$.
Left column: mesh-integrated classical SFEM operators (see section 2.7.3).
Right column: mesh-integrated volume-based SFEM operators (see section 3.7.2).

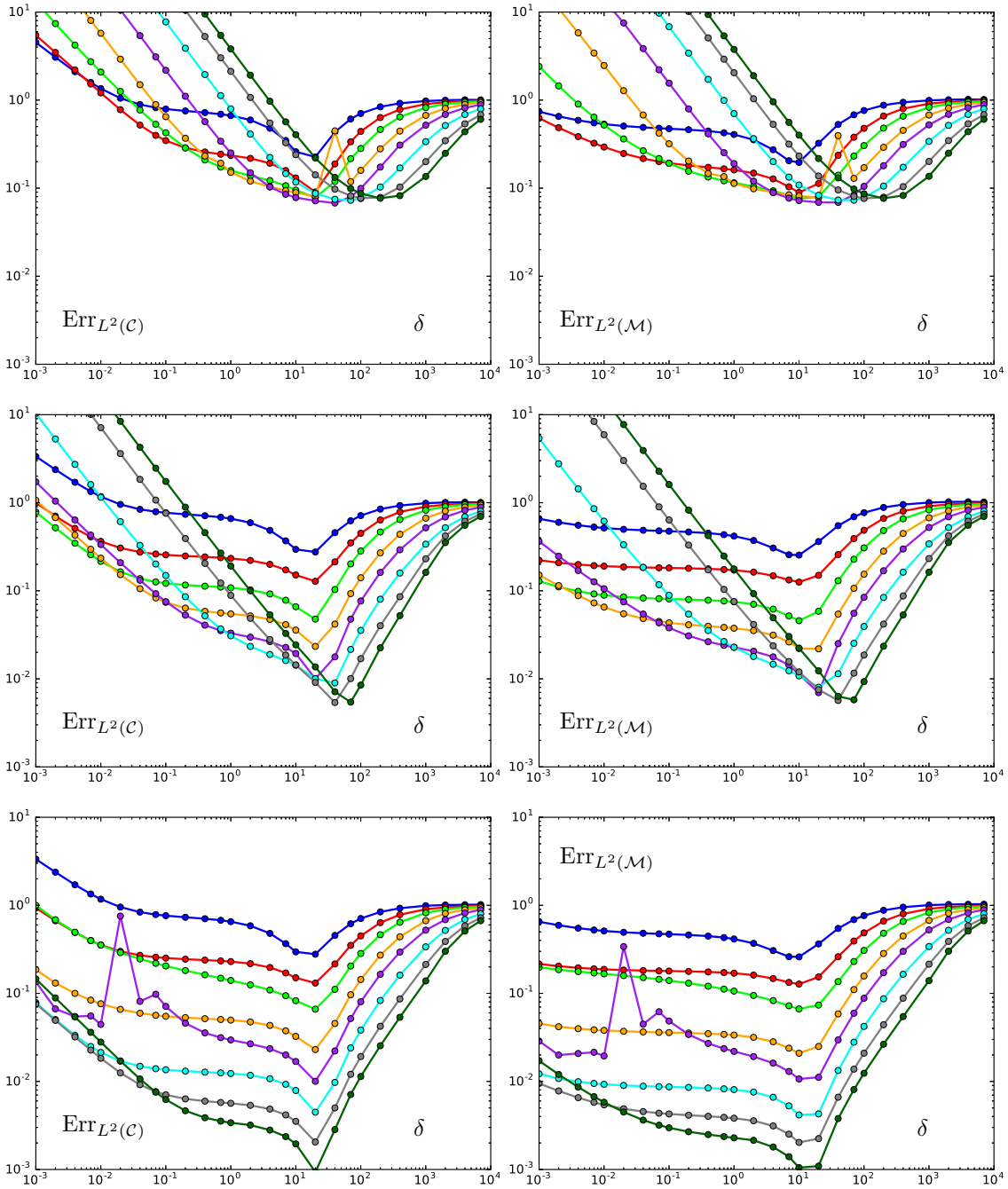


Figure 4.3 – *Relative error on the nodal meshless discretization of the Nitsche weak form of the diffusion equation as a function of the nodal stabilization coefficients (formulation of expression (4.15)). The meshless operators are the mesh-integrated classical SFEM operators defined in section 2.7.3. The point clouds follow a Halton distribution with 900 nodes, 2500 nodes, 4900 nodes, 10000 nodes, 19600 nodes, 40000 nodes, 90000 nodes, 160000 nodes.*
Left column: node-integrated error. Right column: mesh-integrated error.
Top row: integration mesh with 2574 cells. Middle row: integration mesh with 11480 cells. Bottom row: integration mesh with 124422 cells.

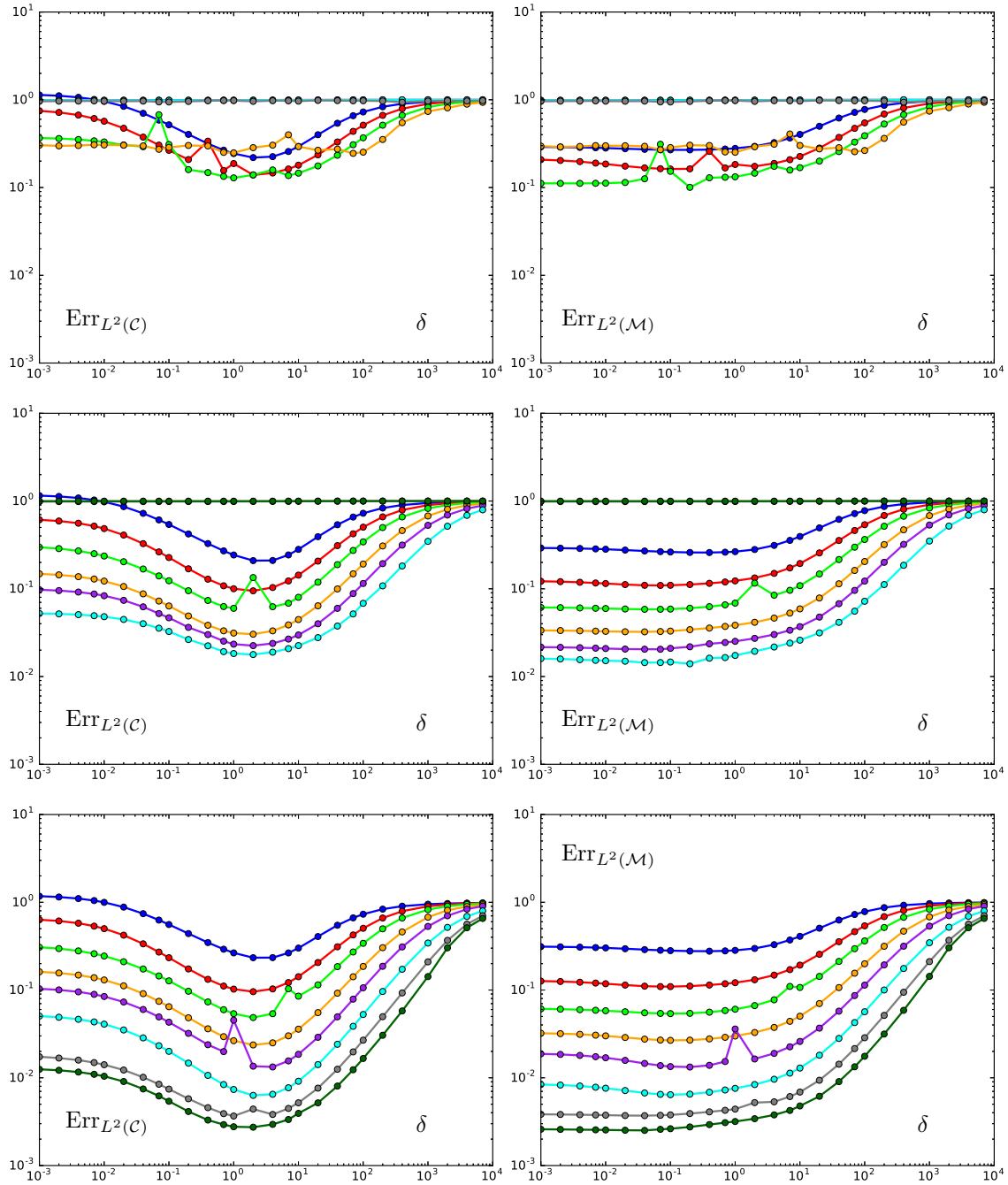


Figure 4.4 – *Relative error on the Galerkin discretization of the Nitsche weak form of the diffusion equation as a function of the nodal stabilization coefficients (formulation of expression (4.15)). The meshless operators are the mesh-integrated volume-based SFEM operators defined in section 3.7.2. The point clouds follow a Halton distribution with 900 nodes, 2500 nodes, 4900 nodes, 10000 nodes, 19600 nodes, 40000 nodes, 90000 nodes, 160000 nodes.*
Left column: node-integrated error. Right column: mesh-integrated error.
Top row: integration mesh with 2574 cells. Middle row: integration mesh with 11480 cells. Bottom row: integration mesh with 124422 cells.

two regions: first a convergent region where the mesh is sufficiently fine to provide a good approximation of the meshless operators. In this region, the measured convergence rate is $h^{1.9}$. Then, an under-integrated region characterized by slower convergence and then a divergence. On the other hand, the results of the simulation using the volume-based SFEM operators are hardly affected by the stabilization.

A higher stabilization ($\delta = 2 \times 10^0$, bottom row of figure 4.2) still greatly improves the behavior of the classical SFEM operators, regardless of the integration level. For the finest mesh (124422 cells), the convergence is clearly second order (the measured rate is $h^{2.05}$) for all point clouds. The addition of the stabilization term delays the under-integration phenomenon: In general, for a given mesh, the error strictly decreases with the cloud refinement level until the number of nodes is $3-4$ times higher than the number of cells. Moreover, the convergence keeps a second order rate until the number of nodes is $1.5-2$ times higher than the number of cells.

For the volume-based SFEM operators, the stabilization decreases the error in the well-integrated cases by a factor $\approx 5-8$, but does not delay the effect of under-integration. The error level still undergoes a very sharp transition in the under-integration case since our linear solver still cannot solve the corresponding equations.

The choice of stabilization parameter

It is clear that a good choice of this stabilization parameter is crucial for a good behavior of the simulation. We now investigate its effect in details. In addition to the cloud-based error, we also compare the linear MLS reconstruction $< u >$ of the discrete solution with the following mesh-based measure of the error:

$$\mathrm{Err}^2_{L^2(\mathcal{M})} \overset{\text{def}}{=} \frac{\fint_{\mathcal{M}} (u^{\text{ex}} - < u^{\text{num}} >)^2}{\fint_{\mathcal{M}} u^{\text{ex}\ 2}} \tag{4.16}$$

On figure 4.3, we represent the evolution of the cloud and mesh-integrated errors with the reconstruction-based stabilization coefficient δ using the classical SFEM operators with linear consistent MLS functions. These plots clearly show that the formulation is not stable since the error increases rapidly as δ gets closer to zero. The error behavior is unimodal: it first rapidly decreases, then reaches a slightly slanted plateau. It slowly decreases again, then again quickly decreases to reach a minimum located in a narrow valley, and then quickly increases to a value of $\approx 100\%$ in the over-stabilized cases.

Both the refinement level of the point cloud and the integration mesh have an influence on the value of the minimum of the error. For a given point cloud, the minimum of the error first decreases as the mesh is refined (under-integration), and then stagnates (over-integration). For a given integration mesh, the behavior is a bit more subtle: in the over-integrated regime, the minimum of the error first decreases as the cloud gets finer, and is reached for a constant value of the stabilization factor ($\delta \approx 2 \times 10^1$ for this test case). Then, as the cloud gets finer, we enter the under-integrated regime. In this case, the minimum of the error is reached for higher values of the stabilization factor. The value of this minimum does not depend on the point cloud, only on the integration mesh. We interpret this effect as a saturation of the integration capability of the point cloud. We also see that there is not much of a difference between the cloud-integrated error and the mesh-integrated error for the classical SFEM operators.

On figure 4.3, we represent the evolution of the cloud and mesh-integrated errors with the reconstruction-based stabilization coefficient δ using the volume-based SFEM operators with linear consistent MLS functions. Contrary to the classical SFEM operators, we see that the formulation is stable (the error does not diverge for low stabilization levels). The general

behavior of the mesh-integrated and cloud-integrated errors are a bit different: in the over-integrated case, the cloud-integrated error first decreases, reaches a minimum in a wide valley, and then increases. On the other hand, the mesh-integrated error is at its minimum value for low values of the stabilization factor and only increases with it, first very slowly, and then more quickly. Even with such a behavior, it is still interesting not to use a zero stabilization factor as the resulting system is better conditioned and between two and three times fewer iterations are needed to achieve convergence. On figure 4.3, we represent the evolution of the cloud and mesh-integrated errors with the reconstruction-based stabilization coefficient δ using the volume-based SFEM operators with linear consistent MLS functions. Contrary to the classical SFEM operators, we see that the formulation is stable (the error does not diverge for low stabilization levels). The general behavior of the mesh-integrated and cloud-integrated errors are a bit different: in the over-integrated case, the cloud-integrated error first decreases, reaches a minimum in a wide valley, and then increases. On the other hand, the mesh-integrated error is at its minimum value for low values of the stabilization factor and only increases with it, first very slowly, and then more quickly. Even with such a behavior, it is still interesting not to use a zero stabilization factor as the resulting system is better conditioned and between two and three times fewer iterations are needed to achieve convergence. The behavior in the under-integrated case is very bad: the linear solver simply does not converge at all and the error is $\approx 100\%$.

The values of the minimal error of the volume-based SFEM formulation is very similar to that of the plateau of the classical SFEM formulation. The minimal value of the error of the classical SFEM formulation is approximately two times lower, but the sensitivity to the stabilization coefficient near the minimum is so high that devising a reliable heuristic to aim for it seems difficult.

In conclusion, the classical and volume-based SFEM operators can both achieve second order convergence and comparable error levels on this test case if a proper stabilization is used. Each method has its strengths and weaknesses: the volume-based version does not require stabilization at all, especially if we consider the reconstructed solution $< u >$. However, its behavior regarding under-integration is very brutal: as soon as the background mesh is too coarse, the discrete system gives a worthless solution.

This is not the case of the classical SFEM operators, which can always give a sound solution even in the under-integrated case, provided adequate stabilization is used. The problem with this formulation is precisely that the stabilization needs a fine-tuning to perform the best given the fineness of the point cloud and the background integration mesh, contrary to its volume-based analogue.

In section 4.2.2, we continue our investigations related to the Nitsche weak form, and compare several mesh-integrated Galerkin formulations of the diffusion equation.

4.2.2 Meshless Galerkin discretization of elliptic equations

In this section, we develop the basics of Element Free Galerkin Methods (EFGs) using weakly imposed Dirichlet boundary conditions with Nitsche boundary conditions. We insist in particular on the corresponding patch test conditions and give numerical evidence of the potential gains in verifying them.

Definition 4.2 (Galerkin method)**.**
Suppose we are interested in finding a solution of the following weak abstract problem: Find $u \in U$ (the trial space) such that $\forall\ v \in V$ (the test space), we have:

$$a(u, v) = L(v) \tag{4.17}$$

Existence and unicity of the solution of the above problem are usually studied through Lax-Milgram type theorems (see section 2.7 of [41] for instance).

Let U^h and V^h be finite dimensional subspaces of U and V respectively. Then a *Petrov-Galerkin method* seeks the solution in the discrete trial space as the restriction of the weak requirements to the discrete test space, namely: Find $u^h \in U^h$ such that for all $v^h \in V^h$, we have:

$$a(u^h, v^h) = L(v^h) \tag{4.18}$$

Assuming the discrete trial and test spaces are equipped with known bases, equation (4.18) reduces to a linear system of size n_{DOF}. In the case where a is symmetric, $U = V$ and $U^h = V^h$, the system is symmetric and the method is called a *Bubnov-Galerkin method*.

Galerkin methods are at the heart of mesh-based simulation. Indeed a Finite Element Method (FEM) is a Galerkin method using function spaces of finite dimension built using geometric considerations on the mesh. For instance, the classical linear finite element method uses the piecewise affine reconstruction on a simplex mesh (see definition 1.15).

The meshless Garlerkin discretization

The so-called Element Free Galerkin Method (EFG) introduced by Belytschko et al. in [27] is a Galerkin method using bases of meshless functions. In other words, it is defined using the image of the reconstruction operator as the discrete test and trial spaces of a Galerkin method: $U^h = V^h = \mathrm{Im}(< . >) = \mathrm{span}((\phi_i^h)_{i\in\mathcal{C}})$.

In most meshless situations however, there is no analytical formula to compute the values of $a(\phi_i, \phi_j)$ and $L(\phi_i)$, for all $i, j \in \mathcal{C}$, and thus, the discrete system (4.18) itself cannot be formed exactly. This is where approximate integration comes in: we need to find discrete versions of the continuous operators a and L to approximate the Galerkin system of equations.

In this section, we introduce several such procedures using the MLS shape functions (defined in appendix D.2). First, we adapt the mesh-based integration procedure for SFEM operators (see section 2.7.3) to the context of EFG, which satisfies exactly the compatibility conditions (the Gauss MLS operator below). This method is compared to the one point integration procedure, which does not satisfy the compatibility condition (the 1 point FMLS operator below), and to the one point integration of the DMLS approximation of the exact derivatives (the 1 point DMLS operator below). Let us thus define $\forall\, u, v \in \mathcal{C} \to \mathbb{R}$:

Gauss MLS This is the Gauss-type one point compatible integration using cell faces (see definition 1.17):

$$\oint_{\mathcal{M}}^{\text{Gauss}} \mathbb{\nabla} u \cdot \mathrm{D} \cdot \mathbb{\nabla} v \overset{\text{def}}{=} \sum_{i,j\in\mathcal{C}} u_i \left(\sum_{\phi^d \in \mathcal{M}} \frac{\mathbf{A}^{\text{Gauss}}_{\phi^d,i} \cdot \mathrm{D}(\mathbf{x}(\phi^d)) \cdot \mathbf{A}^{\text{Gauss}}_{\phi^d,j}}{V(\phi^d)} \right) v_i \tag{4.19}$$

Where we denoted: $\forall\, \phi^d \in \mathcal{M}$, $\forall\, i \in \mathcal{C}$

$$\mathbf{A}^{\text{Gauss}}_{\phi^d,i} \overset{\text{def}}{=} \oint_{\phi^d}^{\text{Gauss 1pt}} \nabla < \delta_i >= \sum_{\phi^{d-1}\in\phi^d} o(\phi^{d-1}, \phi^d) \mathbf{\Gamma}(\phi^{d-1}) \phi_i(\mathbf{x}(\phi^{d-1})) \tag{4.20}$$

Let us recall that the Gauss integration method does not require explicit computation of the gradient of shape functions at any point.

One point FMLS This is the one point integration formula on cells of the mesh (see proposition 1.17) using the exact derivatives of the linear MLS shape functions (see definition D.6):

$$\fint_{\mathcal{M}}^{1 \text{ pt } FMLS} \mathbb{\nabla} u \cdot \mathrm{D} \cdot \mathbb{\nabla} v \overset{\text{def}}{=} \sum_{i,j \in \mathcal{C}} u_i \left(\sum_{\phi^d \in \mathcal{M}} V(\phi^d) \nabla \phi_i(\mathbf{x}(\phi^d)) \cdot \mathrm{D}(\mathbf{x}(\phi^d)) \cdot \nabla \phi_j(\mathbf{x}(\phi^d)) \right) v_j \quad (4.21)$$

One point DMLS This is the one point integration formula on cells of the mesh (see proposition 1.17) using only the consistent part of the MLS derivatives (see definition D.5) denoted $\boldsymbol{\psi}^{DMLS}$ here:

$$\fint_{\mathcal{M}}^{1 \text{ pt } DMLS} \mathbb{\nabla} u \cdot \mathrm{D} \cdot \mathbb{\nabla} v \overset{\text{def}}{=} \sum_{i,j \in \mathcal{C}} u_i \left(\sum_{\phi^d \in \mathcal{M}} V(\phi^d) \boldsymbol{\psi}_i^{DMLS}(\mathbf{x}(\phi^d)) \cdot \mathrm{D}(\mathbf{x}(\phi^d)) \cdot \boldsymbol{\psi}_j^{DMLS}(\mathbf{x}(\phi^d)) \right) v_j \quad (4.22)$$

Remark. All three discrete diffusion operators defined above can be put under the form given in expression (4.19) reminiscent of expression (4.27) of the node-integrated weak diffusion operator. The coefficients $\mathbf{A}_{\phi^d,i}$ form a discrete gradient operator mapping scalar valued fields defined on the point cloud to vector valued fields defined on cells of the mesh $\mathbb{\nabla} : (\mathcal{C} \to \mathbb{R}) \to (\mathcal{M} \to \mathbb{R}^d)$.

The meshless framework of chapter 2 could almost entirely be adapted to accommodate these mixed operators whose integration space (functions defined on cells of a mesh in this case) is different from their degrees of freedom space (functions defined on nodes of a point cloud in this case) as already pointed out in [184]. The only major difference with the node-integrated case would be that we cannot define a dual gradient operator. This remark justifies the notations $\fint_{\mathcal{M}}$ and $\mathbb{\nabla}$ by analogy with the previous node-integrated case of section 4.3.

Going further, the classical SFEM operators of definition 2.27 are in fact nothing but mesh-integrated reconstructions of these mixed operators. Specifically for the mesh-integrated SFEM gradient operator, this remark reads: $\forall\ f : \mathcal{C} \to \mathbb{R}$, $\forall\ i \in \mathcal{C}$

$$V_i \mathbb{\nabla}^{SFEM\ \mathcal{M}} f = \sum_{\phi^d \in \mathcal{M}} V(\phi^d) \phi_i(\mathbf{x}(\phi^d)) \mathbb{\nabla}_{\phi^d}^{\text{Gauss}} f \quad (4.23)$$

In all three discretizations, we use the symmetric SFEM boundary integration operator (see expressions (2.78) and (2.79)) for the bilinear uses, and the following counterpart for linear uses: $\forall\ v : \mathcal{C} \to$, $\forall\ g : \partial\Omega \to \mathbb{R}$,

$$\fint_{\partial\mathcal{M}}^{SFEM\ \mathcal{M}} < u > g = \sum_{i \in \mathcal{C}} \left(\sum_{\phi^{d-1} \in \partial\mathcal{M}} \boldsymbol{\Gamma}(\phi^{d-1}) \phi_i(\mathbf{x}(\phi^{d-1})) g(\mathbf{x}(\phi^{d-1})) \right) u_i \quad (4.24)$$

In other words, there is no need to project the boundary fields onto $\mathrm{Im}(< . >_{|\partial\Omega})$.

Replacing the above discrete operators into expressions (4.3) yields the discrete Nitsche formulation. Similarly to the nodal integration weak nodal discretization of section 4.2, the patch test conditions of the final discrete diffusion system are formally equivalent to Stokes's theorem.

$$\fint_{\mathcal{M}} \mathbb{\nabla} v = \fint_{\partial\mathcal{M}} v \quad (4.25)$$

These are exactly the compatibility conditions, in the non-nodal integration context. These conditions were already pointed out in [12, 14] in the very context of EFG.

The proof of proposition 2.10 can easily be adapted to show that the Gauss integration method satisfies the above requirement exactly. The 1 point FMLS method does not satisfy the requirements exactly, but the integration error decreases to arbitrarily low values with the fineness of the mesh (see proposition 1.17): it is asymptotically compatible. The one point DMLS method does not satisfy expression (4.25) neither exactly nor asymptotically as the integration mesh is refined.

Numerical experiments

In this section, we perform a battery of tests comparing the different integration procedures of the Galerkin weak form introduced in the previous section using the MLS shape functions with a linear reconstruction basis. The problem is the diffusion test-case with exact solution given in expression (4.10), with $\mathbf{k} = 2\pi\,(2, 4)$. The top half of the boundary is treated with Neumann boundary conditions (*i.e.* $\partial\Omega_N = \{\mathbf{x} \in \partial\Omega \mid x_2 \geq \frac{1}{2}\}$ and the bottom half is treated with Dirichlet boundary conditions. The Nitsche boundary stabilization term is given by the discretization of expression (4.4) using expression (2.79) in the LHS and expression (4.24) in the RHS. The boundary stabilization γ is chosen just high enough so that the formulation is stable but no too high as to avoid ill-conditioning in accordance with recommended usage. Specifically, we used the value $\gamma = \frac{1\times10^1}{h}$ for this test-case. In order to compare the numerical result to the exact analytical solution, we use the mesh-integrated error defined in expression (4.16). The smoothing length is set as $4\sqrt{\frac{V(\Omega)}{\pi n_n}}$.

On the left column of figure 4.5, we plot the error on the diffusion problem as a function of the smoothing length for several levels of mesh refinement with a very low stabilization coefficient ($\delta = 1 \times 10^{-3}$). Roughly speaking, for the Gauss integration, the error decreases superlinearly with cloud refinement as long as the number of nodes in the point cloud remains lower than approximately $1 - 2$ times the number of cells in the mesh (in this region, the measure convergence rate is $h^{1.65}$, then increases. For the one point FMLS integration, the error decreases with the same rate, but only as long as the number of nodes remains lower than $\approx 30\%$ of the number of cells. As for the one point DMLS integration method, it does not converge past a $\approx 10\%$ error level, and the error increases approximately as the nodes outnumber the cells. On this first test, we see that the Gauss integration method systematically performs the best. The result using the Gauss and the one point FMLS integration methods are quite similar as long as the point cloud is fine enough. On the other hand, the one point DMLS formulation systematically performs poorly.

On the right column of figure 4.5, we perform the same tests, but with a higher stabilization coefficient ($\delta = 1 \times 10^1$). In all cases, this change postpones the instability due to under-integration with little change in the error of well-integrated cases. The Gauss and one point FMLS integration methods exhibit a clear second order convergence rate (the measured rate is $h^{2.13}$), and the error of the one point DMLS integration method seems to exhibit a shy $\propto h^{1.3}$ behavior when the mesh is fine enough.

On figure 4.6, we further study the influence of the stabilization coefficient on the final error. For the Gauss integration, we see that if the mesh is fine enough, the error decreases with the smoothing length and is never lowered with added stabilization. However, when under-integration occurs, stabilization allows the error to reach a threshold only determined by the fineness of the integration mesh. The behavior of the one point FMLS integration method is roughly the same except for a few important details: under-integration occurs much more quickly, the error threshold is a bit higher and the width of the low error "valleys" is smaller, especially for highly under-integrated cases. This means that in these cases, the sensitivity with respect to the stabilization coefficient is greater. These plots again show that the one point DMLS integration method performs poorly: in the over-integrated cases,

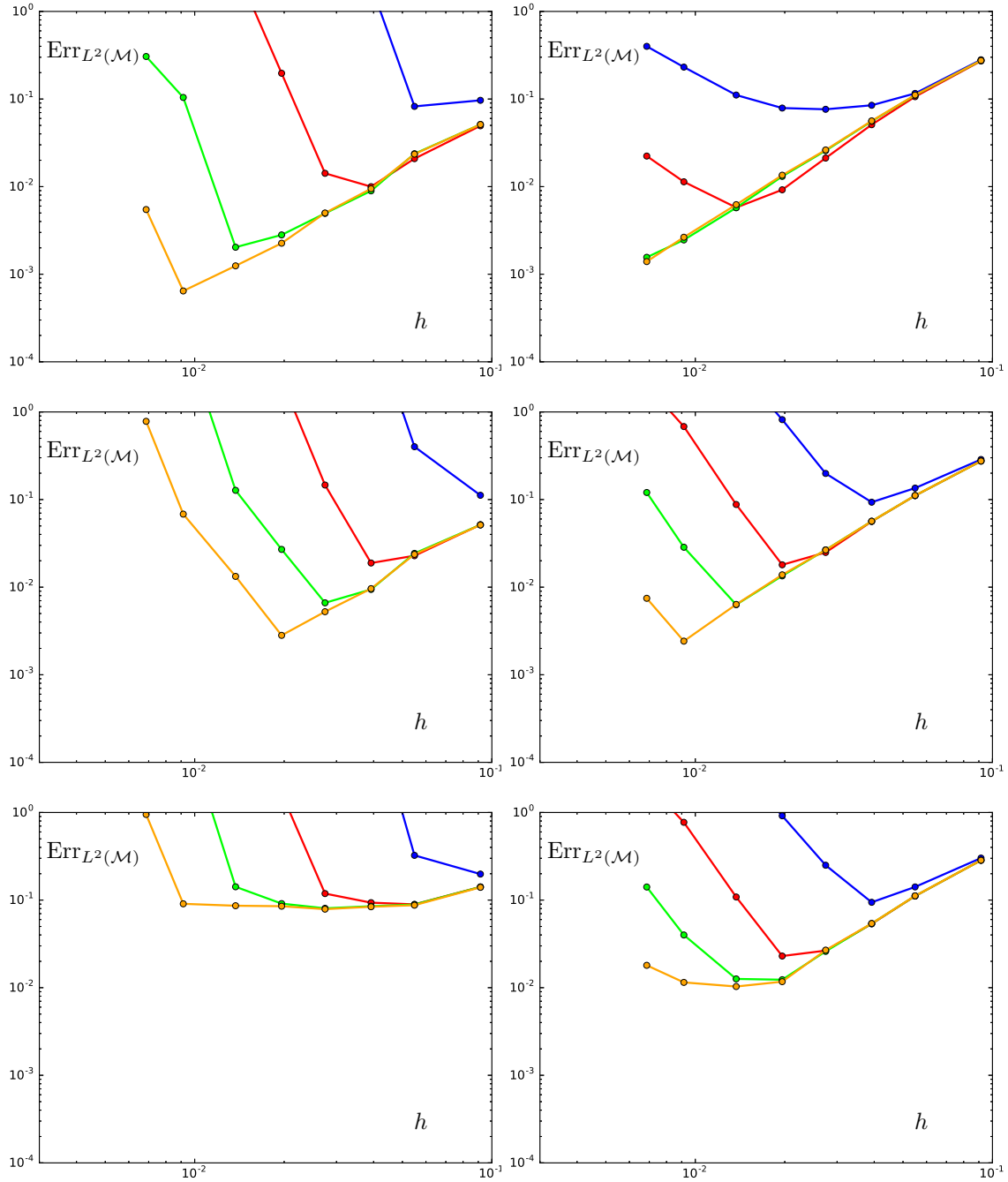


Figure 4.5 – *Relative mesh-integrated error on the Galerkin discretization of the Nitsche weak form of the diffusion equation as a function of the smoothing length. The point clouds follow a Halton distribution. Integration is performed on a mesh made of 2574 cells, 11480 cells, 47492 cells and 124422 cells. Left column: the stabilization coefficient is $\delta = 1 \times 10^{-3}$, which is very low. Right column: the stabilization coefficient is $\delta = 1 \times 10^{1}$. Top row: compatible Gauss MLS operators. Middle row: asymptotically compatible one point FMLS operators. Bottom row: non-compatible one point DMLS operators.*

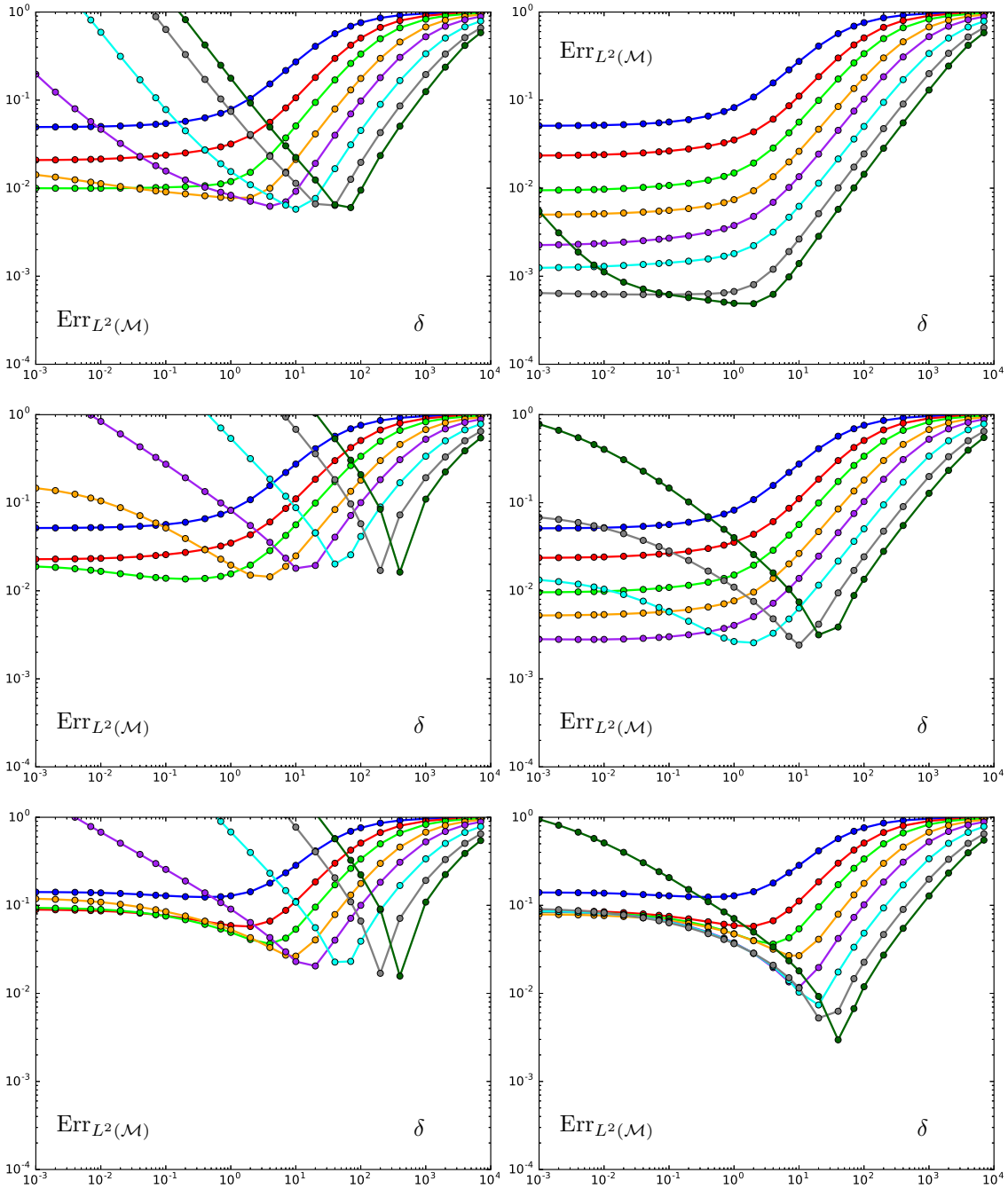


Figure 4.6 – *Relative mesh-integrated error on the Galerkin discretization of the Nitsche weak form of the diffusion equation as a function of the stabilization coefficients. The point clouds follow a Halton distribution with 900 nodes, 2500 nodes, 4900 nodes, 10000 nodes, 19600 nodes, 40000 nodes, 90000 nodes, 160000 nodes.*
Left column: the integration mesh is composed of 11480 cells. Right column: the integration mesh is composed of 124422 cells.
Top row: compatible Gauss MLS operators. Middle row: asymptotically compatible one point FMLS operators. Bottom row: non compatible one point DMLS operators.

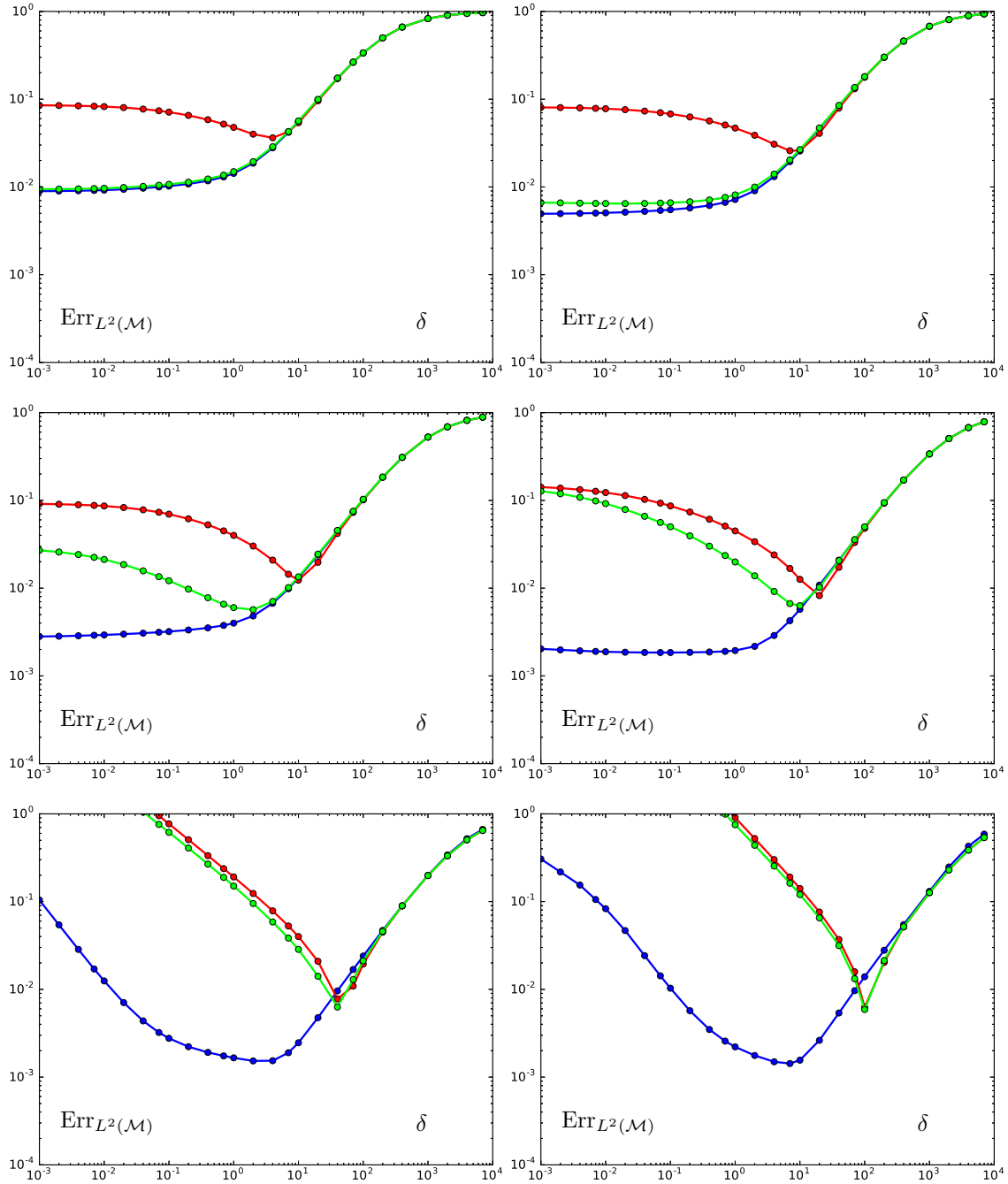


Figure 4.7 – *Relative mesh-integrated error on the Galerkin discretization of the Nitsche weak form of the diffusion equation as a function of the stabilization coefficients. The discrete operators are the exactly compatible Gauss MLS operators, the asymptotically compatible one point FMLS operators and the non compatible one point DMLS operators all using a mesh with 47492 integration cells. The simulation is performed on point clouds following a Halton distribution with 4900 nodes (top left), 10000 nodes (top right), 19600 nodes (middle left), 40000 nodes (middle right) 90000 nodes (bottom left) and 160000 nodes (bottom right).*

the error does not decrease with the smoothing length. Surprisingly enough, the error does decrease with added stabilization though, but sensibility to the stabilization coefficient at the optimal value is extremely high, making it hard to devise reliable heuristics.

We have seen that the classical mesh-integrated SFEM gradient operator is nothing but a reconstruction of the Gauss gradient of the shape functions (see expression 4.23). With this in mind, we would expect the behavior of the Gauss MLS operators to be similar to that of the classical mesh-integrated SFEM on the diffusion test-case. However, a qualitative comparison of figures 4.3, 4.4 and 4.6 show that in the well-integrated case, its behavior is much closer to that of the volume-based SFEM operators. Actually, the Gauss MLS operators seem to reconcile the best of both SFEM formulations: stabilization is not a pre-requisite for convergence when the mesh is fine enough (as with the volume-based SFEM formulation), and the transition to the under-integrated case can effectively be smoothly tempered with an appropriate stabilization (as with the classical SFEM formulation).

Finally, figure 4.7 compares the stabilization profiles of the three integration methods. For over-integrated systems (top left plot), the Gauss and the one point FMLS behaviors are almost identical. This is to be expected: in this case, discrete integration is fine enough to approximate the Galerkin system well enough compared to the approximation level of the (exact) Galerkin system to the continuous diffusion problem. In opposition, the DMLS integration does not approximate the Galerkin system to a reasonable level.

As the point cloud is refined, the exact corresponding Galerkin system better approximates the continuous diffusion problem, but since the discrete integration is performed on the same mesh, the integration error on the Galerkin system increases. This is where the compatibility error of the FMLS integration method reaches too high a value. As the compatibility error increases, the stabilization profile of the FMLS integration method drifts off from that of the exactly compatible Gauss integration method and gradually switches to that of the non-compatible DMLS method. Notice how thin the range of optimal stabilization coefficient is in FMLS and the DMLS case compared to the Gauss case when the ratio of cloud size to mesh size is the highest (bottom right plot).

In conclusion, we can say that the real benefit of using exactly compatible sets of operators as opposed to asymptotically compatible ones or even non-compatible ones is to exploit the discrete integration structure (which is given by the mesh in this case) to its full potential. If the mesh is fine enough, compatible operators yield the optimal second order consistency, and when it is coarse compared to the point cloud, compatible operators reach the lowest possible error permitted by the mesh when combined with an appropriate stabilization procedure. In this regard, a compatible of the Galerkin formulation combines the best of the classical and volume-based SFEM operators of the nodal discretization described in section 4.2.1: it does not suffer the instability of the classical SFEM operators, and still allows for a smooth transition in the under-integrated case contrary to the volume-based SFEM operators. On the other hand, if nothing is implemented to control the error on Stokes's formula, any formulation is very likely to be wildly unstable and require unreasonably fine-tuning of the stability factors in order to yield satisfactory convergence behavior.

Finally, let us point out another computational detail in favor of the Galerkin approach: for a same discretization level, the number of connectivities (hence of stored non-zero values) is lower in the case of the Galerkin discretization than in the case of the nodal weak form with SFEM operators. In the exactly integrated case, two nodes i and j participate to a non-zero coefficient if the support of their shape functions intersect in the Galerkin case, and if these supports each intersect that of a common node k in the SFEM case. Denoting h the radius of the MLS shape functions, this translate to a maximum distance between nodes of $\|\mathbf{x}_j - \mathbf{x}_i\| \leq 2h$ in the Galerkin case and $\|\mathbf{x}_j - \mathbf{x}_i\| \leq 4h$ in the SFEM case. The analysis is a bit more subtle if we take discrete integration into account. Nevertheless, we numerically

verified on our two dimensional test case with Halton point clouds that the number of non-zeros is ≈ 3.2 times greater for the nodal weak form with the SFEM operators than for the Galerkin weak form in the over-integrated case, and this discrepancy increases in the under-integrated case.

4.3 Meshless discretization of the classical weak form: dealing with essential boundary conditions

In this section, we study nodal operator-based discretizations of the classical weak form of elliptic equations. In the scalar case, the continuous formulation is given in expression (4.2). In section 4.3.1, we give a first naive version that has the major drawback of requiring boundary nodes to be placed exactly on the boundary. In section 4.3.4, we address this limitation and propose a formulation that circumvents this problem through the definition of discrete Sobolev spaces.

4.3.1 A first approach with discretization nodes on the boundary

Bluntly replacing continuous operators with their discrete equivalents yields the following nodal discretization:

Definition 4.3 (Weak discretization of the Poisson problem)**.**
We suppose given a set of discrete operators $\left(\int\!\!\!\!\!\!\diagup_{\mathcal{C}}, \int\!\!\!\!\!\!\diagup_{\partial\mathcal{C}}, \mathbb{\nabla}\right)$ on a point cloud $(\mathcal{C}, \mathbf{X})$. Then, the weak nodal discretization of the Poisson problem reads: find $u : \mathcal{C} \to \mathbb{R}$ such that we have:

$$\begin{cases} \int\!\!\!\!\!\!\diagup_{\mathcal{C}} \mathbb{\nabla} u \cdot \mathrm{D} \cdot \mathbb{\nabla} v = \int\!\!\!\!\!\!\diagup_{\mathcal{C}} sv + \int\!\!\!\!\!\!\diagup_{\partial\mathcal{C}_N} gv & \forall\ v : \mathcal{C} \to \mathbb{R} \text{ such that } v_{|\partial\mathcal{C}_D} = 0, \\ u_i = u_D(\mathbf{x}_i) & \forall\ i \in \partial\mathcal{C}_D \end{cases} \tag{4.26}$$

The weak discrete diffusion operator is symmetric. In coordinates, it reads: $\forall\ i, j \in \mathcal{C}$,

$$\int\!\!\!\!\!\!\diagup_{\mathcal{C}} \mathbb{\nabla}\delta_i \cdot \mathrm{D} \cdot \mathbb{\nabla}\delta_j = \sum_{k \in \mathcal{C}} \frac{\mathbf{A}_{k,i} \cdot \mathrm{D}_k \cdot \mathbf{A}_{k,j}}{V_k} \tag{4.27}$$

Moreover, it is non-negative if the volume integration operator is non-negative since in this case, we have: $\forall\ v : \mathcal{C} \to \mathbb{R}$,

$$\int\!\!\!\!\!\!\diagup_{\mathcal{C}} \mathbb{\nabla} v \cdot \mathrm{D} \cdot \mathbb{\nabla} v \geq 0 \tag{4.28}$$

In the special case where we only have Dirichlet boundary conditions ($\partial\mathcal{C}_N = \varnothing$), these equations can be rewritten in the following pointwise form:

$$\begin{cases} -\mathbb{\nabla}_i^* \cdot \mathrm{D} \cdot \mathbb{\nabla} u = s(\mathbf{x}_i) & \forall\ i \in \mathcal{C} \backslash \partial\mathcal{C}, \\ u_i = u_D(\mathbf{x}_i) & \forall\ i \in \partial\mathcal{C} \end{cases} \tag{4.29}$$

Hence, this type of weak symmetric elliptic operator is nothing but the composition of the dual and the primal differentiation operators.

Remark. The space of real valued discrete functions which vanish on the Dirichlet boundary is a discrete version of the Sobolev space of square integrable fields with square integrable

weak derivatives whose trace vanishes on the Dirichlet boundary. A possible notation for this space could thus be:

$$H^1_{0,D}(\mathcal{C}) \overset{\text{def}}{=} \{v : \mathcal{C} \to \mathbb{R} \mid v_{|\partial\mathcal{C}_D} = 0\} \tag{4.30}$$

Combined with our *a priori* choice of reduction operator (see definition 2.4), this essential choice of handling Dirichlet boundary conditions constrains Dirichlet boundary nodes to lie exactly on the boundary since the Dirichlet field u_D is only defined on the boundary of Ω. For this reason, we suppose *in this section only* that boundary nodes lie exactly on the boundary. Of course, this drastically reduces the panel of operators available from chapters 2 and 3. For simplicity reasons, we furthermore suppose the boundary integration operator diagonal, so that we only need to define vector boundary integration weights of the form $(\mathbf{\Gamma}_i)_{i\in\partial\mathcal{C}}$.

In section 4.3.4, we explore the possibility of building discrete Sobolev spaces in the case where nodes of $\partial\mathcal{C}$ do not necessarily lie on the boundary of the continuous domain $\partial\Omega$.

Let us now derive the patch test conditions for the above discrete formulation:

Proposition 4.2 (Patch-test conditions for system (4.26))**.**
If $\mathbb{V}$ is P-1 consistent and $\left(\fint_{\mathcal{C}}, \fint_{\partial\mathcal{C}}, \mathbb{V}\right)$ form a compatible set of discrete operators (see definition 2.18), then the weak discretization of the diffusion equation given in definition 4.3 passes the linear patch test.

Proof. Setting $\mathrm{D} = \mathrm{cst}$, $u = \mathbf{a}\cdot\mathbf{x} + b$, $s = 0$, $g = \mathbf{a}\cdot\mathrm{D}\cdot\mathbf{n}$ in equation (4.26), we get: $\forall\, v : \mathcal{C} \to \mathbb{R}$ such that $v_{|\partial\mathcal{C}_D} = 0$,

$$\mathbf{a}\cdot\mathrm{D}\fint_{\mathcal{C}}\cdot\mathbb{\nabla} v = \mathbf{a}\cdot\mathrm{D}\cdot\fint_{\partial\mathcal{C}} v \tag{4.31}$$

Which is a consequence of compatibility. □

This characterization of the first order consistency of the weak diffusion operator is the reason why we invest so much effort into designing compatible gradients.

Numerical experiments

Let us now compare the results of this formulation. As remarked above, we need the boundary nodes to be placed exactly on the boundary. The point clouds are thus augmented with a row of n_{bou} evenly distributed nodes on the boundary of the domain. These nodes are chosen as boundary nodes and the discrete boundary operator is defined as:

$$\fint_{\partial\mathcal{C}}\mathbf{u} = \sum_{i\in\partial\mathcal{C}} \frac{\mathrm{Area}(\partial\Omega)}{n_{\mathrm{bou}}}\mathbf{u}_i\cdot\mathbf{n}(\mathbf{x}_i) \tag{4.32}$$

In other words, the boundary integration weights are $\mathbf{\Gamma}_{i,j} \overset{\text{def}}{=} \frac{\mathrm{Area}(\partial\Omega)}{n_{\mathrm{bou}}}\delta_{i,j}\mathbf{n}(\mathbf{x}_i)\ \forall\ i,j \in \mathcal{C}$. Similarly, we choose uniform volume weights. This choice is made so that the integration operator satisfy the necessary condition for compatibility (see 2.3).

On figure 4.8, we compare the cloud-integrated error of the weak discretization using several discrete gradients with different consistency properties on the test-case with exact solution given in expression (4.10) using Dirichlet boundary conditions. We use two different nodal arrangements for these simulations: the Cartesian and Halton distributions (see

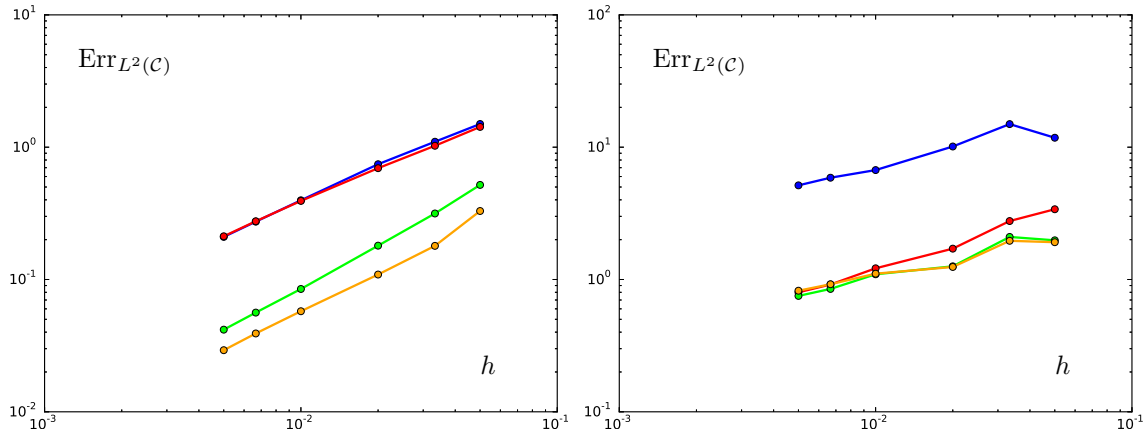


Figure 4.8 – *Relative L^2 errors on the diffusion test case as a function of the smoothing length. The ratio h^d/V is kept constant for all point clouds. Tests were made using the standard SPH gradient (see expression 2.44), the renormalized SPH gradient of order zero (see expression 2.49), the renormalized SPH gradient of order one, and the linear DMLS gradient using a singular weight function.*
Left plot: Cartesian distribution. Right plot: Halton distribution.

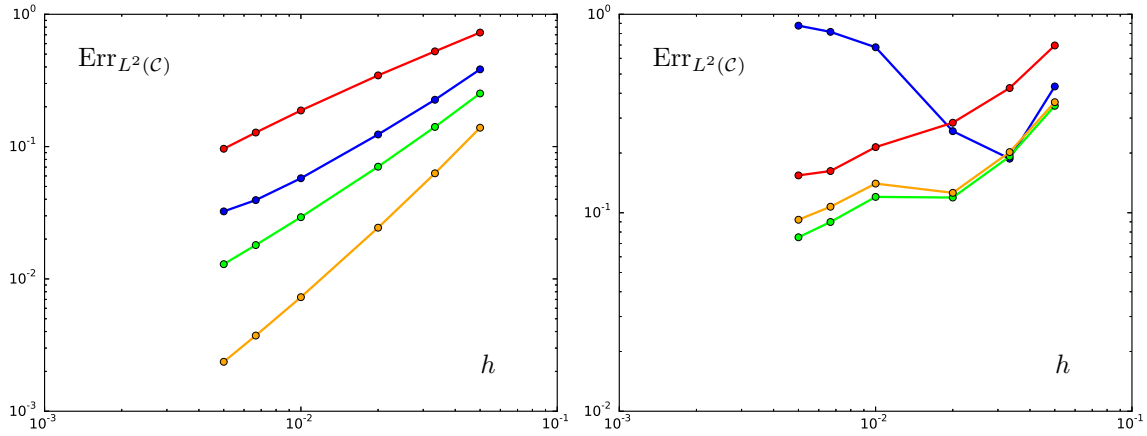


Figure 4.9 – *Relative L^2 errors on the diffusion test case as a function of the smoothing length, with added stabilization. Tests were made using the standard SPH gradient (see expression 2.44), the renormalized SPH gradient of order zero (see expression 2.49), the renormalized SPH gradient of order one, and the linear DMLS gradient using a singular weight function.*
The stabilization factors are chosen as: $w_{i,j} = \delta \frac{\sqrt{V_i V_j}}{(h/\alpha)^2 + \|\mathbf{x}_j - \mathbf{x}_i\|^2}$ with $\alpha = 20$ and $\delta = 5 \times 10^{-2}$.
Left plot: Cartesian distribution. Right plot: Halton distribution.

figure 2.1 a) and c)) and four discrete gradient operators: the classical SPH gradient, its renormalized versions of order zero and one as well as the DMLS gradient.

On the Cartesian arrangement, every discrete gradient achieves first order convergence on the diffusion system measured in the discrete L^2 norm. However, this behavior is not reproduced on the Halton distribution, where no simulation yields a relative error lower than 100%. The highly oscillatory pattern of the error suggests that the formulation is not stable on the Halton distribution, which prevents convergence.

Since the SPH method does not define first order accurate reconstruction, the stabilization given in expression 4.15 would hurt the consistency of the formulation (*i.e.* the patch test conditions would not be the same).

We thus want to devise a stabilization term that does not harm first order consistency of the diffusion system. Supposing that $\mathbb{\nabla}$ is first order consistent (this is one of the patch test conditions), and that u is a discrete fields with a locally linear behavior around a node i, *i.e.* that the restriction of u to $\mathcal{N}(i)$ is linear. Then the first order Taylor approximation at node i is exact on its neighborhood: $\forall\, j \in \mathcal{N}(i)$, $u_j = u_i + (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbb{\nabla}_i u$. This observation suggests the following stabilization term to be added in the weak formulation:

$$a_{\text{stab}}(u, v) = \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{N}(i)} w_{i,j} (u_j - u_i - (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbb{\nabla}_i u)(v_j - v_i - (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbb{\nabla}_i v) \tag{4.33}$$

This term is symmetric and non-negative provided the stabilization weights $w_{i,j}$ are non-negative. It vanishes if u is a linear field, thus does not affect the patch test conditions. Its role is to penalize the solution field from deviating from locally (*i.e.* on each neighborhood of the point cloud) deviating from a linear function.

Remark. Considering physical dimensions, the right scaling for the edge stabilization coefficients is $w_{i,j} \propto \|\mathrm{D}\| h^{d-2}$. Moreover, an appropriate choice of weight stabilization coefficients can be made such that a_{stab} has the same matrix coefficients as those of the least-norm correction of definition 2.22. Exploiting this fact in a simulation code results in reduced memory footprint and CPU effort. While very practical, such a choice is not motivated in theory, and for our tests, we used the following value for edge coefficients:

$$w_{i,j} = \delta \frac{\sqrt{V_i V_j}}{(h/\alpha)^2 + \|\mathbf{x}_j - \mathbf{x}_i\|^2} \tag{4.34}$$

The factor $\alpha \neq 0$ is added as to avoid divergence in the case where two nodes are very close to each other. Its value is (somewhat arbitrarily) set to 20, and is of secondary importance. On the other hand, the choice of an adequate value for δ is crucial to achieve the best possible level of convergence.

On figure 4.9, we compare the L^2 errors of the weak discretization with added stabilization on the diffusion test case. On the Cartesian arrangement, every gradient achieves at least first order convergence on the diffusion system. The DMLS gradient even achieves second order convergence. On the Halton arrangement however, the classical SPH gradient does not lead to a convergent discretization, whereas all the other gradients yield an oscillating convergence with rate close to one. And indeed, the standard SPH gradient is not even consistent of order zero, whereas all other gradients are consistent of order at least zero, ensuring the consistency of order zero of the weak Laplace operator. This test confirms that stable meshless discretizations that have order zero consistency are convergent of order one.

Achieving consistency of order one (*i.e.* passing the linear patch test) not only requires the gradient operator to be linear consistent, but also zeroth-order dual consistent. The next section addresses this topic.

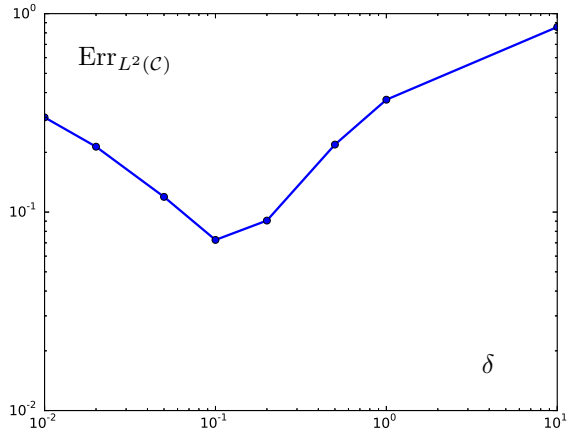


Figure 4.10 – *Relative L^2 errors on the diffusion test case using the non-corrected order one renormalized SPH gradient on a Halton distribution with 2500 nodes as a function of the non-dimensional stabilization factor δ. The evolution of the error is comprised of two steps: it first decreases from the unstabilized value $\delta = 0$ to the optimal value ($\delta \approx 1 \times 10^{-1}$ here), then the error steadily increases towards $\lessapprox 100\%$ as $\delta \to \infty$.*

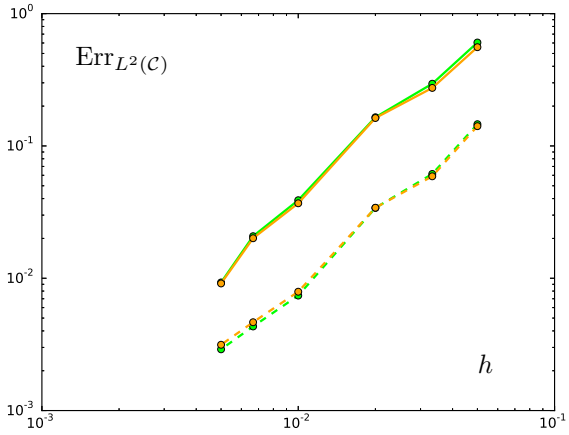


Figure 4.11 – *Relative L^2 errors on the diffusion test case as a function of the smoothing length. We used the corrected renormalized SPH gradient of order one and the linear DMLS gradient, with (dashed lines, $\delta = 5 \times 10^{-2}$) and without (plain lines) stabilization on Halton distributions.*

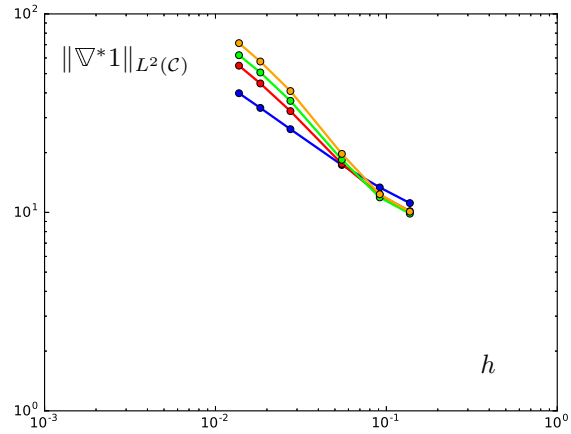


Figure 4.12 – *L^2 norm of the error on the dual of a unit field on Halton distributed point clouds with the standard SPH gradient, the renormalized SPH gradient of order zero, the renormalized SPH gradient of order one, and the linear DMLS gradient.*

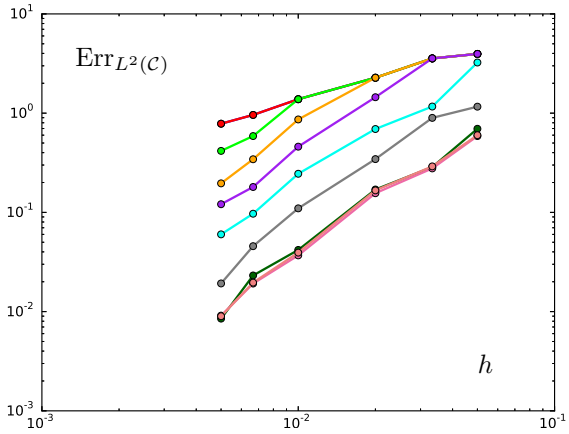


Figure 4.13 – *Relative L^2 errors on the diffusion test case using partially-corrected DMLS gradients.*
$\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 50$, $\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 20$,
$\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 10$, $\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 5$,
$\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 2$, $\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 1$,
$\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 0.5$, $\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 0.2$,
$\|\mathbb{\nabla}^* 1\|_{L^2(\mathcal{C})} \leq 0.1$

4.3.2 Is compatibility really beneficial?

On figure 4.11, we represent the L^2 errors of the weak discretization on the same test-case using two first order consistent compatible gradients generated with the correction procedure described in section 2.6.2. On this figure, we see that the convergence is greatly increased (the measured rate of convergence is $h^{1.8}$), and that the error level is always lower compared to non-compatible discretization. For an unknown reason, the compatible gradients did not exhibit the same loss of convergence due to unstable behavior as their non-compatible analogues: it seems that compatibility has a regularizing effect on the system. This effect is not total as the error level is still lowered by a multiplicative factor ranging from 3 to 5 with the addition of the stabilization term.

Compatibility, *i.e.* zeroth order consistence of the dual gradient is thus confirmed to be a highly desirable feature. However, most of the purely meshless operators described in the literature are not compatible. Even worse, the error on their dual gradients quickly increases with the fineness of the point cloud when the nodal distribution is not regular. On figure 4.12, we represent the L^2 norm of the error of the dual gradient on a unit field as a function of the smoothing length for several non-corrected gradient operators on the Halton distribution. For first order consistent operators, this quantity varies roughly as $\propto h^{0.9}$.

In order to solve the correction system and generate the corrected gradients, we explicitly form the covariance matrix of the constrained least squares problem and used an in-house deflated restarted Generalized Minimal Residuals (GMRES) linear solver. The iterative nature of this procedure makes it possible to stop the algorithm before convergence, and generate discrete operators with a lower but still non-vanishing Stokes error. This feature allows us to test the following hypothesis: is it really needed to require that the dual gradient of a uniform field exactly vanishes for the weak discretization to ensure convergence? Instead, it is possible to retain optimal convergence, only bounding the value of $\|\mathbb{\nabla}^*1\|_{L^2(\mathcal{C})}$ independently of the level of refinement of the point cloud (*i.e.* $\|\mathbb{\nabla}^*1\|_{L^2(\mathcal{C})} = \mathcal{O}(1)$ instead of $\mathcal{O}(h^\alpha)$ with α close to 1).

On figure 4.13, we represent the relative L^2 error on the diffusion test case for partially corrected DMLS gradients, *i.e.* corrected DMLS gradients whose correction coefficients are the result of the partially converged GMRES algorithm. Iterations are stopped when the integrated Stokes error $\|\mathbb{\nabla}^*1\|_{L^2(\mathcal{C})}$ decreases below a given threshold. As expected, the error on the diffusion system is a decreasing function of the threshold on the Stokes error. Interestingly enough, a quasi-second order rate of convergence is restored in all cases, and the error on the diffusion system reaches a plateau at $\|\mathbb{\nabla}^*1\|_{L^2(\mathcal{C})} \approx 0.5$. Lower values of Stokes errors does not have any beneficial effect on the error on the diffusion system.

Remark. In fact, compatibility is also to be credited for convergence in the case of non-corrected gradients on the Cartesian distribution (see figure 4.9 a)). Indeed, we observed that the symmetries in the nodal distribution helped the error on the primal gradient to remain low on the regular Cartesian cloud compared to that on the irregular Halton arrangement: the measured behavior of the Stokes error in the Cartesian case as a function of the smoothing length is only $\|\mathbb{\nabla}^*1\|_{L^2(\mathcal{C})} \propto h^{0.17}$, namely a much lower increase rate than for the more irregular Halton distribution.

This remark shows that the regularity and symmetry properties of the point cloud can greatly influence the properties of the discrete operators built on it. This is similar to mesh-based simulation, whose accuracy is in most cases heavily influenced by the quality of the mesh (skewness of cells, aspect ration, ...). We have already seen in section 3.6 that several classical stabilization procedures (*e.g.* XSPH, background pressure, Fick-based shifting ...) of the SPH method having a regularizing effect on the nodal distribution can actually be re-interpreted as gradient descent algorithms seeking to decrease the Stokes error. The

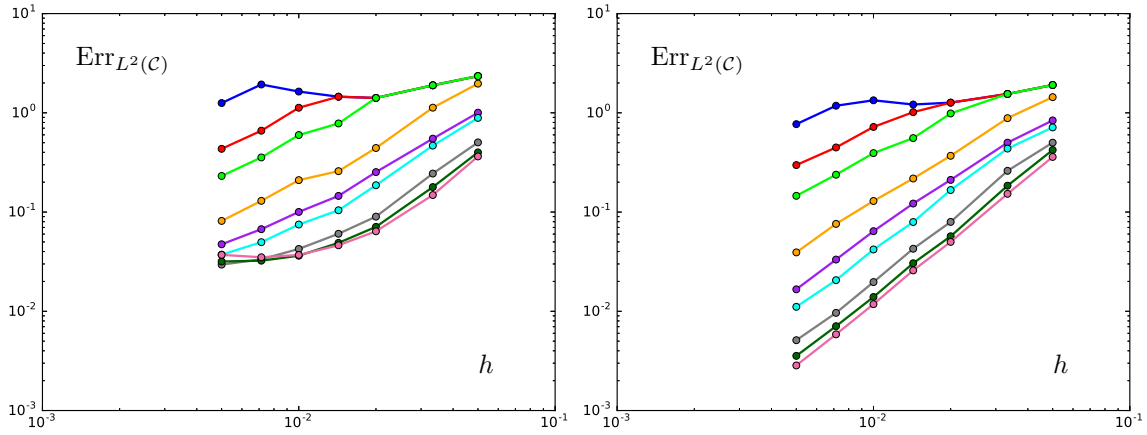


Figure 4.14 – *Relative error of the laplacian system as a function of discretization length using the volume-based SPH gradient (left) and the renormalized volume-based SPH gradient of order one (right) on point clouds with Stokes error kept below 100, 50, 30, 10, 5, 3, 1, 0.5 and 0.3.*

resulting point clouds exhibit little variation in distance to the closest neighbor, producing "harmonious" and "eye-pleasing" nodal distributions. The regularity of these arrangements (see figure 3.4) is clearly akin to that of Cartesian point clouds.

In particular, these intrinsically iterative algorithms can be stopped as soon as a desired level of convergence is achieved, producing point distributions on which the volume-based SPH operators (see section 3.5) achieve any level of compatibility error. This gives us another means of confirming that bounding the compatibility error as the point cloud is refined actually suffices to ensure second order convergence.

In order to run simulations on these point clouds, we impose the values of nodes in the motionless "protective" layer of nodes to the exact solution values, which is tantamount to considering these nodes as Dirichlet boundary nodes. The fact that these nodes do not lay on the boundary does not mean that we have lifted this constraint of the discretization. Rather, it is a numerical trick commonly referred to as "ghost nodes" or "ghost particles" (see [152, 165, 193, 202] for some examples of usage of ghost particles). We do not consider that using ghost particles is a satisfying method for the imposition of boundary conditions, but it is a good enough placeholder to prove our point.

The corresponding discrete L^2 error on the diffusion test case using the volume-based SPH gradient without stabilization term is plotted on the left of figure 4.14. The point clouds are generated starting from the Halton distribution and solving the system of ODEs defined in expression (3.71) until the Stokes error reaches the required level. On the finest point cloud, the error is ≈ 30 times lower on the optimized point cloud compared to the non-optimized ones, which we interpret as the consequence of increased compatibility. The plot shows a reduced convergence rate as the smoothing length increases. This behavior is expected since the volume-based SPH gradient is not first-order compatible.

During our tests, we were surprised to discover that the blue noise distributions generated by the optimization procedure were susceptible to give low values of the Stokes error, even for gradients for which it was not initially designed. For instance, we were able to decrease the Stokes error for the renormalized version of the volume-based gradient (which we could not itself re-interpret as a volume-based gradient) to values as low as $\|\nabla^{VSPH\ R1} * 1\| \leq 0.3$ for

all tested point cloud refinement levels. Even if convergence to a zero value is not eventually achieved, this represents a very significant improvement, with a huge impact on the results of the diffusion test case. The corresponding error is plotted on the right of figure 4.14. In particular, the three last curves (Stokes error below 1, 0.5 and 0.3 respectively) exhibit a clear second order convergence behavior.

The key take-away is the following: what is *really* necessary in order to ensure a correct convergence behavior is to be able to control the compatibility error of the discrete operators. In the author's opinion, this is the most promising approach to circumvent the meshless curse and devise efficient truly meshless discretization of PDEs.

In the next section, we assess the main issues and limitations of the previous formulations, especially concerning the enforcements of boundary conditions.

4.3.3 Critique of the nodal weak formulation

We have seen in sections 4.3.1 and 4.3.2 that the classical weak discretization of the diffusion equation of definition 4.3 can achieve very satisfying quasi-second-order convergence rates if used with compatible sets of first order consistent discrete operators.

However, in most cases of operator construction with built-in compatibility (for instance with the SFEM operators of sections 2.7.2 and 2.7.3), it is difficult to ensure that boundary nodes do lie on the boundary. Indeed, the formulation is very restrictive when it comes to the boundary cloud $\partial\mathcal{C}$. Specifically, boundary nodes *have to* be placed exactly on the boundary of the domain: $\forall\ i \in \mathcal{C},\ \mathbf{x}_i \in \partial\Omega$. This constraint appears because boundary condition fields (Dirichlet u_D and Neumann g field) are only defined on the boundary and our choice of reduction map entails that function evaluations only occur at nodes of the point cloud. Furthermore, this condition cannot be relaxed for Dirichlet nodes if we insist on treating them as *loci* for the essential boundary conditions, *i.e.* if we keep on imposing $u_i = u_D(\mathbf{x}_i)\ \forall\ i \in \partial\mathcal{C}_D$.

The alternative is to combine different discrete operators in a set without built-in compatibility (such as the renormalized SPH methods). In this case, we have seen that there is much to gain either from correcting the gradient operator (which is a CPU and memory intensive operation), or using harmonious distributions of points which typically have a lower Stokes error. Unfortunately, designing harmonious point distributions with the constraint that boundary nodes are exactly on $\partial\Omega$ can also prove very challenging.

A second objection to the weak formulation presented earlier concerns the role of nodes and edges of the graph of the point cloud. As we have seen in section 2.2.2, nodes in a point cloud are analogous to cells in a mesh (they hold a volume, ...), and edges are analogous to faces (they separate cells, they hold vector surface areas, ...). It is thus surprising that boundary conditions representing the behavior of the sought field on a surface should be imposed on discrete objects analogous to a volume. It would instead be more coherent to impose boundary conditions on edges of the graph rather than on nodes.

Let us note that these problems are specific to the classical weak formulations with essential treatment of the Dirichlet boundary conditions, and do not occur with the Nitsche symmetric formulation using either the operator-based discretization of section 4.2.1 or the Galerkin-type discretization of section 4.2.2.

In order to overcome these issues, we propose a new scheme for the discretization of the diffusion equation: in section 4.3.4, we augment the operator framework with a definition of boundary edge positions, allowing for the imposition of edge-based essential boundary conditions through the introduction of appropriate discrete Sobolev spaces.

4.3.4 Essential enforcement of Dirichlet boundary conditions: Designing discrete function spaces

In this section, we address the weaknesses detailed in section 4.3.3, and devise a novel meshless method with a different treatment of the boundary. This method is inspired by the work of Krongauz and Belytschko [131] concerning meshless Galerkin discretizations (see also the related "continuous blending" method by Huerta [95, 96, 119, 120]).

This new formulation still treats Dirichlet boundary conditions essentially, *i.e. via* a restriction of the trial space (the space of candidate solutions to the problem). For our diffusion test case, this means in particular that we need to find a discrete analogue of $H^1(\Omega)_{0,D}$. This is achieved with the definition of positions for boundary edges: in sharp contrast with the boundary nodes of the weak operator discretization detailed in section 4.3, boundary edges do not correspond to degrees of freedom in the final discrete Poisson system, they do not hold a volume integration weight V_i and the are not *loci* of the definition of the discrete gradient.

Let us initially ignore the issue of the boundary and consider a point cloud $(\mathcal{U}, \mathbf{X})$ of a space larger than Ω. This point cloud is called "the universe". We suppose built on the universe a discrete integration $\fint_{\mathcal{U}}$ and a first order consistent gradient operator $\mathbb{\nabla}^{\mathcal{U}}$. We show how the actual discretization of the computational domain Ω can be inherited from operators defined on $\mathcal{U}$, even though the actual process of generating the discrete operators might not go through this detour.

In a first step, it seems natural to only retain the nodes of the universe $\mathcal{U}$ that are located inside Ω. Thus, let us define the interior cloud $\mathring{\mathcal{C}}$ as:

$$\mathring{\mathcal{C}} = \{i \in \mathcal{U} \mid \mathbf{x}_i \in \Omega\} \tag{4.35}$$

On figure 4.15 a), the nodes of $\mathring{\mathcal{C}}$ are displayed in blue and the nodes of $\mathcal{U} \backslash \mathring{\mathcal{C}}$ in black. This restricted cloud is naturally equipped with the following volume quadrature method, inherited from the original cloud $\mathcal{U}$: $\forall\ f : \mathcal{U} \to \mathbb{R}$,

$$\fint_{\mathcal{C}} f \overset{\text{def}}{=} \fint_{\mathcal{U}} f \delta_{\mathring{\mathcal{C}}} = \sum_{i \in \mathring{\mathcal{C}}} V_i f_i \tag{4.36}$$

This approximation is obviously quite crude, but nevertheless points to the fact that $L^2(\mathcal{C}) \overset{\text{def}}{=} \mathring{\mathcal{C}} \to \mathbb{R}$, along with a positive volume quadrature method, is a good choice for a discrete equivalent of $L^2(\Omega)$.

Ultimately, the semi-norm of a function f in a discrete equivalent of $H^1(\Omega)$ should read:

$$|f|^2_{H^1(\mathcal{C})} = \fint_{\mathcal{C}} \|\mathbb{\nabla} f\|^2 = \sum_{i \in \mathring{\mathcal{C}}} V_i \|\mathbb{\nabla}_i f\|^2 \tag{4.37}$$

If the gradient operator $\mathbb{\nabla}$ used in this definition is chosen to be the restriction of the gradient operator defined on $\mathcal{U}$, then the right hand side of equation (4.37) is ill defined. Indeed, there are nodes in $\mathring{\mathcal{C}}$ whose neighborhood is not included in $\mathring{\mathcal{C}}$: they have at least one neighbor in $\mathcal{U} \backslash \mathring{\mathcal{C}}$, where the f_i values are not defined. In other words, edges that join nodes inside Ω to outside nodes prevent equation (4.37) from being well defined. We define $\partial\mathcal{C}$ to be the set of those edges:

$$\partial\mathcal{C} \overset{\text{def}}{=} \{(i, o) \in \mathring{\mathcal{C}} \times \mathcal{U} \backslash \mathring{\mathcal{C}} \mid o \in \mathcal{N}(i)\} \tag{4.38}$$

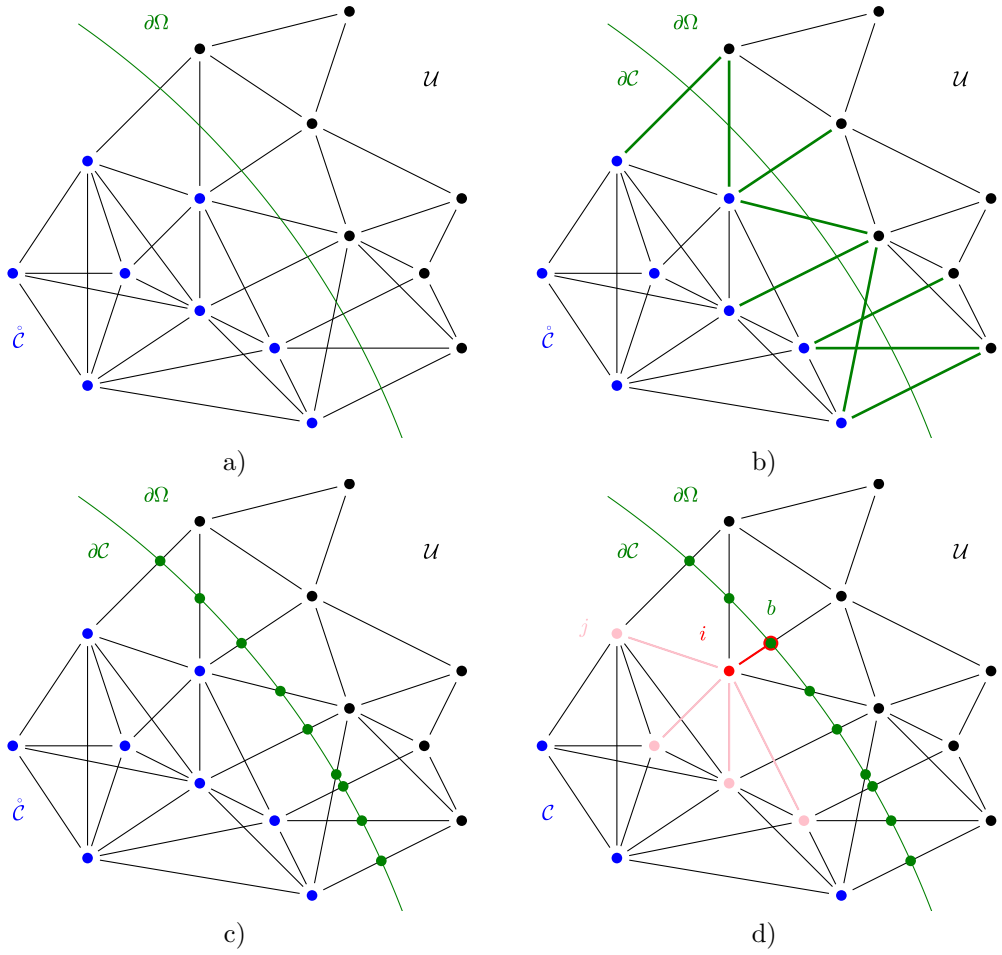


Figure 4.15 – *Restriction of the "universe" i.e. a covering point cloud $\mathcal{U}$ to the computational domain gives a way to capture the boundary.*
a) The continuous boundary $\partial\Omega$ separates the interior nodes $\mathring{\mathcal{C}}$ from the exterior nodes $\mathcal{U}\backslash\mathring{\mathcal{C}}$.
b) Boundary edges $\partial\mathcal{C}$ are edges that cross the physical boundary.
c) Boundary node positions are defined as the intersection of boundary edges and the physical boundary.
d) Structure of the point cloud at the vicinity of the boundary. A traversing edge is detected since it has a single interior node i. The geometrical intersection of the straight line representing this edge with the exact boundary $\partial\Omega$ gives the position associated to the boundary edge $\mathbf{x}_b$. The trace operator effectively maps a function defined on i and its interior neighbors j to a value at b.

At this point, let us point out that this choice of boundary edges is not the only possible one, but presents several desirable properties mentioned later and its actual construction can easily be automated.

This situation is represented in figure 4.15 b). The new point cloud $\mathcal{C}$ naturally inherits the graph structure defined on $\mathcal{U}$, and the $\partial\mathcal{C}$ is the set of edges that "go outside", *i.e.* that are not linked to a node of $\mathring{\mathcal{C}}$. It is convenient to attach a node to each of these edges, but the status of these "boundary nodes" is quite different from the "interior nodes". Indeed, they are linked only to a single interior node (we do take advantage of this feature when building the discrete trace operator), and they do not possess a volume.

Another (equivalent) justification for the definition of the boundary cloud $\partial\mathcal{C}$ is the following: the boundary $\partial\mathcal{C}$ is made up of the objects that separate $\mathring{\mathcal{C}}$ and $\mathcal{U}\backslash\mathring{\mathcal{C}}$. Indeed, any continuous path on the graph of $\mathcal{U}$ joining a node of $\mathring{\mathcal{C}}$ to a node of $\mathcal{U}\backslash\mathring{\mathcal{C}}$ necessarily contains at least one edge in $\partial\mathcal{C}$. The continuous equivalent of this remark is found when applying the intermediate value theorem to the signed distance function to Ω: each continuous path joining a point of $\mathring{\Omega}$ and a point of $\mathbb{R}^d\backslash\Omega$ crosses its zero level-set $\partial\Omega$ at least once. In particular for every edge $b = (i, o)$ of $\partial\mathcal{C}$, there exists at least one point of the segment $[\mathbf{x}_i, \mathbf{x}_o]$ lying on $\partial\Omega$ as pictured on figure 4.15 c). In coordinates, the corresponding system of equations reads:

$$\begin{cases} \mathbf{x}_b = (1-\alpha_b)\mathbf{x}_i + \alpha_b\mathbf{x}_o \\ \alpha_b \in [0,1] \\ \mathbf{x}_b \in \partial\Omega \end{cases} \tag{4.39}$$

Provided the discretization is fine enough and non-degenerate, there is only one solution to this problem, which we denote $\mathbf{x}_b$ and call position of the boundary node b.

Remark. This process can easily be adapted to the modeling of inner boundaries as shown on figure 4.16. Contrary to regular boundaries, inner boundaries cut edges joining two interior nodes, so that one boundary node needs to be introduced on either side of the boundary, for a total of two boundary nodes per cut edge. At the end of this process, the edge joining two interior nodes separated by an inner boundary is effectively severed, replaced by two edges each joining an interior node and a boundary node, with both boundary nodes sharing the same position.

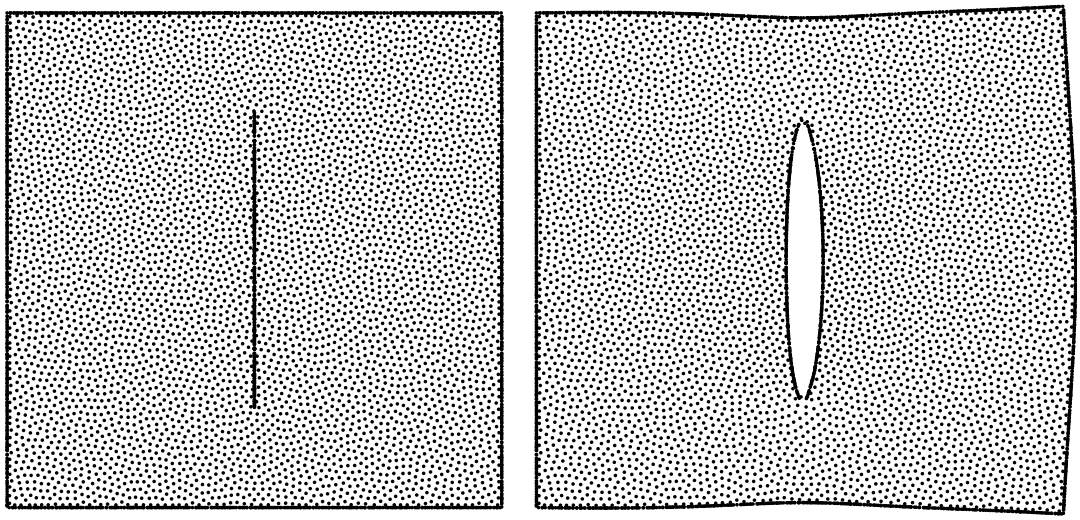
Similarly to definition 2.7, we define a linear integration operator for fields defined on the boundary: $\forall\, f : \partial\mathcal{C} \to \mathbb{R}$,

$$⨍_{\partial\mathcal{C}} f = \sum_{b\in\partial\mathcal{C}} \Gamma_b f_b \tag{4.40}$$

Where $\Gamma_i > 0$. We also denote $L^2(\partial\mathcal{C}) \overset{\text{def}}{=} \partial\mathcal{C} \to \mathbb{R}$ equipped with the scalar product $(f,g)_{L^2(\partial\mathcal{C})} = ⨍_{\partial\mathcal{C}} fg$.

Remark. Because of the dual status of the discrete boundary (its components are simultaneously edges and nodes), this definition does not contradict definition 2.7. Instead of using nodal values for the boundary integration – which somehow forced us to define a bi-linear boundary integration operator in order to define the dual gradient operator – we define a position for boundary edges and define edge values ... which makes boundary edges very comparable to nodes. For this reason, we do not try to distinguish the two notions further and define the whole point cloud as $\mathcal{C} \overset{\text{def}}{=} \mathring{\mathcal{C}} \cup \partial\mathcal{C}$.

The fact that we do not define volume weights for boundary nodes does not allow the definition of a dual gradient operator. Consequently, compatibility takes a different form than the simple $\mathbb{V}^*1 = 0$, but is no less essential as confirmed by our tests.



a) b)

Figure 4.16 – *Cloud of points with inner boundary.*
a) Discretization of the body at rest. Every edge of the initial point cloud intersecting the inner boundary is split into two edges, and two boundary nodes are introduced at the same position.
b) Deformed cloud with inner boundary in a traction test simulation. The two sides of the inner boundary are now clearly visible.

Finally, we can adapt the definition of the gradient operator. Equation (4.37) only requires the gradient of a function to be defined in the interior cloud $\mathring{\mathcal{C}}$. Thus the discrete gradient operator $\mathbb{\nabla} : (\mathcal{C} \to \mathbb{R}) \to (\mathring{\mathcal{C}} \to \mathbb{R}^d)$ naturally maps discrete (scalar) fields defined on the cloud with its boundary to discrete (vector) fields defined on the interior cloud: $\forall\ f : \mathcal{C} \to \mathbb{R}, \forall\ i \in \mathring{\mathcal{C}}$:

$$V_i \mathbb{\nabla}_i f \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \mathbf{A}_{i,j} f_j \tag{4.41}$$

At this point, the only missing ingredient is the definition of discrete Sobolev spaces. Although the trivial choice $H_0^1(\mathcal{C}) \overset{\text{def}}{=} \{f : \mathcal{C} \to \mathbb{R} \mid \forall\ b \in \partial\mathcal{C}\ ,\ u_b = 0\}$ is satisfying, the space $(\mathcal{C} \to \mathbb{R})$ should not be chosen as a discrete version of $H^1(\Omega)$: it is too big. Indeed, its dimension is the total (interior and boundary) number of nodes, which is higher that the dimension of $L^2(\mathcal{C}) = \mathring{\mathcal{C}} \to \mathbb{R}$. More precisely, we can argue that $(f, g) \mapsto \oint_{\mathcal{C}} fg$ is not a scalar product on $\mathcal{C} \to \mathbb{R}$ because of its rank deficiency on the boundary (boundary nodes do not hold a volume).

The ability to define a discrete equivalent of the Sobolev space $H^1(\Omega)$ is closely linked with the representation of the boundary. Indeed, $H^1(\Omega)$ is rich enough to define the "restriction" (or trace, see [88] for a rigorous introduction to the trace operator) of a function to the boundary $\partial\Omega$ in a continuous way whereas $L^2(\Omega)$ is not restrictive enough. In other words, the differentiability properties of functions in $H^1(\Omega)$ give enough structure to the space to be able to define boundary values. This remark suggests using our discrete differentiation operator to define a trace operator. The simplest way to do that is to perform a linear

reconstruction of boundary values: $\forall\ u : \mathcal{C} \to \mathbb{R}$, $\forall\ b = (i, o) \in \partial\mathcal{C}$:

$$\hat{u}_b = u_i + (\mathbf{x}_b - \mathbf{x}_i) \cdot \mathbb{\nabla}_i u \tag{4.42}$$

Setting interior values and requiring that every reconstructed value $\hat{u}_b$ should match the boundary value u_b constitutes a system of linear equations. There is exactly one equation and one unknown per boundary node, hence this system cannot be overdetermined. If it is underdetermined, we consider the solution with the least L^2 norm. This procedure implicitly defines a trace operator $\mathrm{Tr}^{\mathcal{C}} : (\mathring{\mathcal{C}} \to \mathbb{R}) \to (\partial\mathcal{C} \to \mathbb{R})$.

A careful examination of equation (4.42) reveals that the definition of the trace of a discrete function at a boundary node $b = (i, o)$ only requires the values of the neighbors $j \in \mathring{\mathcal{C}}$ of its sole interior neighbor $i \in \mathring{\mathcal{C}}$ as depicted on figure 4.15 d). Indeed, every boundary node has only one interior neighbor, so that there is no coupling between boundary nodes relative to different interior nodes. Consequently, writing boundary values as linear combinations of interior values (a technique known as *static condensation*) only requires the inversion of small and local linear systems. This desirable property is a direct consequence of the definition of boundary nodes as edges of $\mathcal{U}$ traversing $\partial\Omega$.

Our discrete version of $H^1(\Omega)$ is then made up of functions defined on the point cloud $\mathcal{C}$, whose values on the boundary agree with their linear approximations:

$$\begin{aligned} H^1(\mathcal{C}) &= \{u : \mathcal{C} \to \mathbb{R} \mid u_{|\partial\mathcal{C}} = \mathrm{Tr}^{\mathcal{C}}(u_{|\mathring{\mathcal{C}}})\} \\ &= \{u : \mathcal{C} \to \mathbb{R} \mid \forall\ b = (i, o) \in \partial\mathcal{C},\ u_b = u_i + (\mathbf{x}_b - \mathbf{x}_i) \cdot \mathbb{\nabla}_i u\} \end{aligned} \tag{4.43}$$

Finally, in order to define $H^1_{0,D}(\mathcal{C})$, we demand that Dirichlet boundary node values should vanish, and that the others should match the linearly reconstructed value:

$$\begin{aligned} H^1_{0,D}(\mathcal{C}) = \{u : \mathcal{C} \to \mathbb{R} \mid \forall\ b = (i, o) \in \partial\mathcal{C}\backslash\partial\mathcal{C}_D,\ u_b = u_i + (\mathbf{x}_b - \mathbf{x}_i) \cdot \mathbb{\nabla}_i u \\ \forall\ b \in \partial\mathcal{C}_D,\ u_b = 0 \qquad \} \end{aligned} \tag{4.44}$$

Since interior values suffice to define an element of $H^1_{0,D}(\mathcal{C})$, its dimension (the number of degrees of freedom of the problem) is the number of interior nodes (the cardinal of $\mathring{\mathcal{C}}$). In the end, the spaces $H^1(\mathcal{C})$, $H^1_{0,D}(\mathcal{C})$ and $H^1_0(\mathcal{C})$ all have the same dimension and are strictly distinct subspaces of $(\mathcal{C} \to \mathbb{R})$.

We now have all the tools we need to write a discretization of the diffusion problem on $\mathcal{C}$: Given discrete fields D, s, g and u_D, find $u : \mathcal{C} \to \mathbb{R}$ such that for all $v \in H^1_{0,D}(\mathcal{C})$, we have:

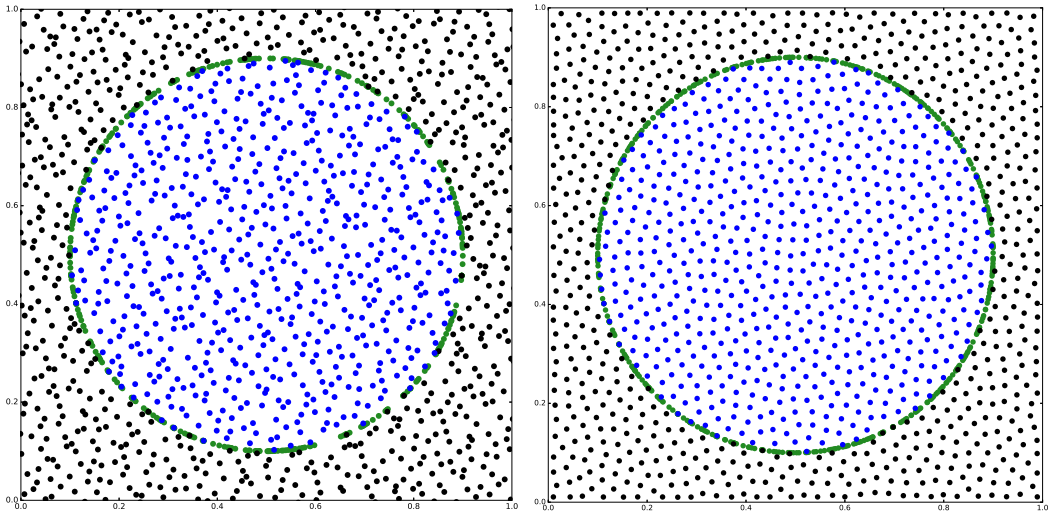
$$\begin{cases} \fint_{\mathcal{C}} \mathbb{\nabla} u \cdot \mathrm{D} \cdot \mathbb{\nabla} v = \fint_{\mathcal{C}} sv + \fint_{\partial\mathcal{C}_N} gv \\ \qquad u - u_D \in H^1_{0,D}(\mathcal{C}) \end{cases} \tag{4.45}$$

Once again, necessary conditions to pass the linear patch test are the first order consistency of $\mathbb{\nabla}$ as well as the following discrete version of Stokes's theorem:

$$\forall\ v : \mathcal{C} \to \mathbb{R},\quad \fint_{\mathcal{C}} \mathbb{\nabla} v = \fint_{\partial\mathcal{C}} v\mathbf{n} \tag{4.46}$$

Making use of the fact that boundary nodes are linked to a single interior node, let us re-write the above compatibility requirements in coordinates:

$$\begin{cases} \sum_{i \in \mathring{\mathcal{C}}} \mathbf{A}_{i,j} = \mathbf{0} & \forall\ j \in \mathring{\mathcal{C}} \\ \mathbf{A}_{i,b} = \Gamma_b \mathbf{n}_b & \forall\ b = (i, o) \in \partial\mathcal{C} \end{cases} \tag{4.47}$$



a) b)

Figure 4.17 – *The clipping procedure described in this section allow the representation of the boundary from a covering point cloud. Each boundary node corresponds to a traversing edge in the original point cloud.*
a) Clipped Halton distribution with 1277 interior nodes on $\Omega = [0,1]^2$*. The clipping procedure retains 644 interior nodes and creates 411 boundary nodes.*
b) Clipped blue noise distribution with 1277 interior nodes on $\Omega = [0,1]^2$ *. The clipping procedure retains 645 interior nodes and creates 457 boundary nodes.*

For boundary nodes, the above equation states that the gradient coefficients $\mathbf{A}_{i,b}$ should be normal to the boundary and have the magnitude of the boundary integration coefficient Γ_b. This strengthens the belief that graph edges are the meshless equivalent of mesh faces. For interior nodes, equation (4.47) states that the sum of incoming gradient coefficients to any cell should vanish. This condition might be interpreted as the closedness of the meshless "cell". Adaptation of the least-norm correction procedure for compatibility (see definition 2.22) to the above system of equations is almost straightforward.

Numerical experiments on the diffusion equation

We test the method described in the previous section on a similar test case of the diffusion equation. The exact solution is still given by expression (4.10). The computational domain is a ball $\Omega = B^2(\begin{pmatrix}0.5\\0.5\end{pmatrix}, 0.4)$. In order to build the point clouds for the test, we clip point clouds on $[0,1]^2$ using the procedure described in the previous section (see figure 4.17). Note that the blue noise point cloud are optimized for compatibility using the volume-based SPH gradient, and hence are not supposed to verify the compatibility equations for the tested gradients, *a fortiori* not at the boundary.

For all these tests, the volume and boundary integration operators are chosen as integrals of Shepard shape functions, with *ad-hoc* overkill discrete integration formulae on a

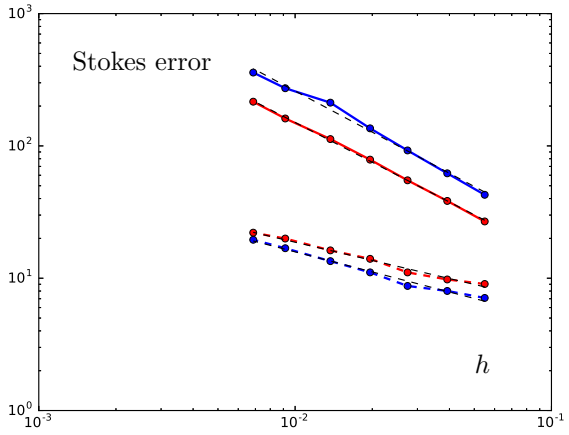


Figure 4.18 – *Evolution of the Stokes error as a function of the smoothing length for of the renormalized SPH gradient of order zero and the linear DMLS gradient on the Halton distribution (plain lines) and the blue noise distribution (dashed lines).*

background Cartesian grid:

$$V_i = \int_\Omega \frac{W_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{j \in \mathcal{C}} W_h(\mathbf{x} - \mathbf{x}_j)} \, \mathrm{d}V \tag{4.48}$$

$$\Gamma_b = \int_{\partial\Omega} \frac{W_h(\mathbf{x} - \mathbf{x}_b)}{\sum_{j \in \mathcal{C}} W_h(\mathbf{x} - \mathbf{x}_b)} \, \mathrm{d}S \tag{4.49}$$

These integration weights are then projected on the space of weights verifying the necessary conditions for compatibility (see proposition 2.3).

On figure 4.19, we compare the relative error for increasing discretization fineness for several point clouds and boundary conditions. The plots show a roughly first order convergence behavior on clouds following the blue noise distribution and no convergence at all (relative error $> 100\%$) on clouds following the Halton distribution, regardless of the boundary condition (full Dirichlet or Neumann) and of the consistency order of the gradient. Once again, this hints at the fact that the blue noise distribution has better properties than the Halton distribution.

Previous tests revealed that the Stokes error (before correction) seems to be affected by the choice of node distribution: the more "harmonious" and eye-pleasing, the lower the Stokes error. Even though we cannot define a dual gradient operator on $\mathcal{C}$ with the updated treatment of the boundary, we can still define the Stokes error as:

$$\sup_{\substack{v \in H^1_{0,D}(\mathcal{C}) \\ \|v\|_{L^2(\mathcal{C})}=1}} \left\| \fint_{\mathcal{C}} \nabla v - \fint_{\partial\mathcal{C}} v \right\| \tag{4.50}$$

This quantity is represented on figure 4.18. A least-square fit reveals that the Stokes error grows as $\propto h^{-1}$ on the Halton point cloud and only as $\propto h^{-\frac{1}{2}}$ on the blue noise distribution, confirming yet again the benefit of using more regular point distributions.

On figure 4.20, we compare the relative error on the same diffusion test-case with corrected gradients. For both boundary conditions and point clouds, a satisfying second order

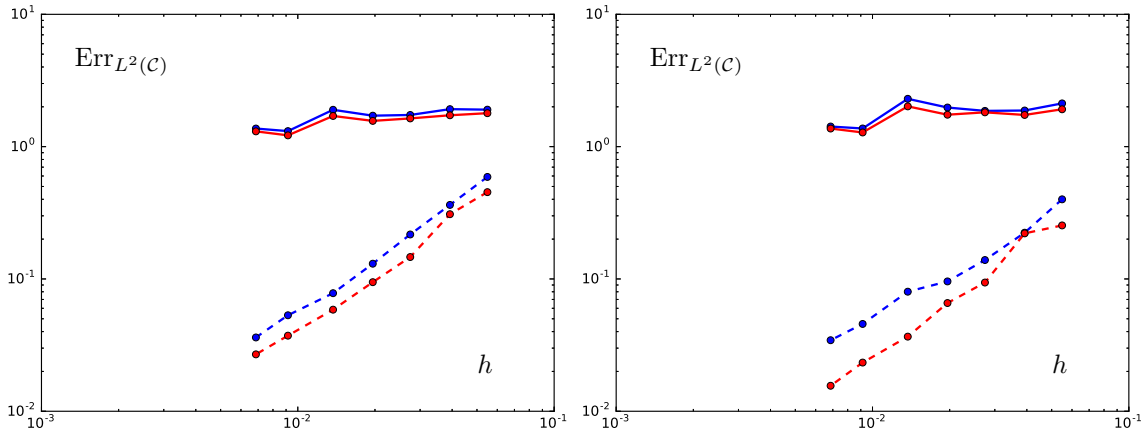


Figure 4.19 – *Relative L^2 errors on the diffusion test case as a function of the smoothing length of the non-stabilized weak formulation. We used the non-corrected renormalized SPH gradient of order zero and the linear DMLS gradient, on the Halton distribution (plain lines) and the blue noise distribution (dashed lines). Left: Dirichlet boundary conditions are imposed on the whole boundary. Right: Neumann conditions are imposed on the whole boundary except for a contiguous arc of* 20° *(i.e.* $\approx 5.56\%$ *of the boundary).*

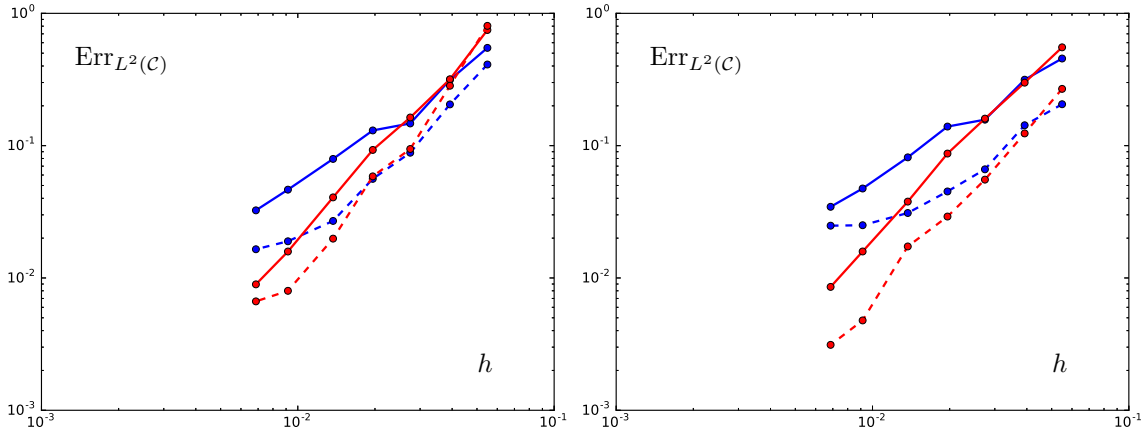


Figure 4.20 – *Relative L^2 errors on the diffusion test case as a function of the smoothing length of the non-stabilized weak formulation. We used the corrected renormalized SPH gradient of order zero and the linear DMLS gradient, on the Halton distribution (plain lines) and the blue noise distribution (dashed lines). Left: Dirichlet boundary conditions are imposed on the whole boundary. Right: Neumann conditions are imposed on the whole boundary except for a contiguous arc of* 20° *(i.e.* $\approx 5.56\%$ *of the boundary).*

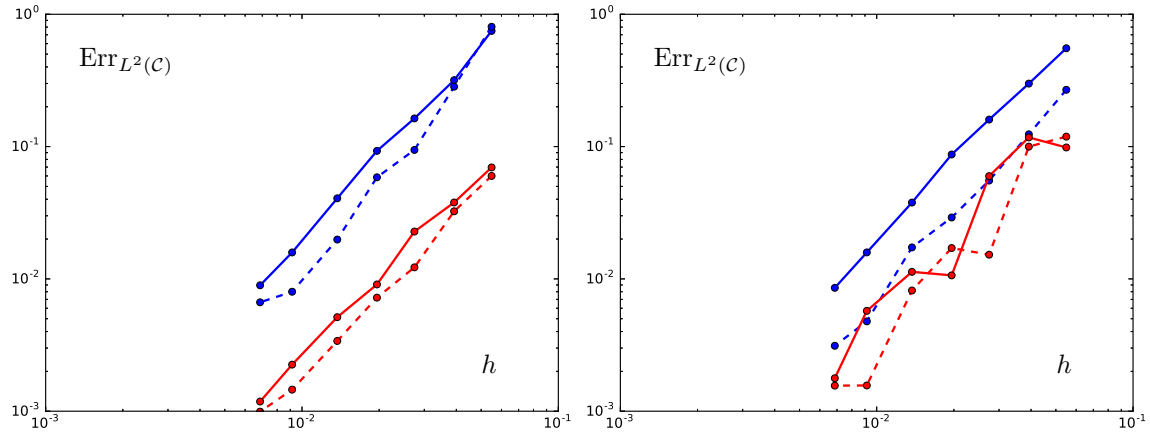


Figure 4.21 – *Relative L^2 errors on the diffusion test case as a function of the smoothing length of the non-stabilized and stabilized weak formulation. We used the corrected linear DMLS gradient, on the Halton distribution (plain lines) and the blue noise distribution (dashed lines). Left: Dirichlet boundary conditions are imposed on the whole boundary. Right: Neumann conditions are imposed on the whole boundary except for a contiguous arc of* $20°$ *(i.e.* $\approx 5.56\%$ *of the boundary).*

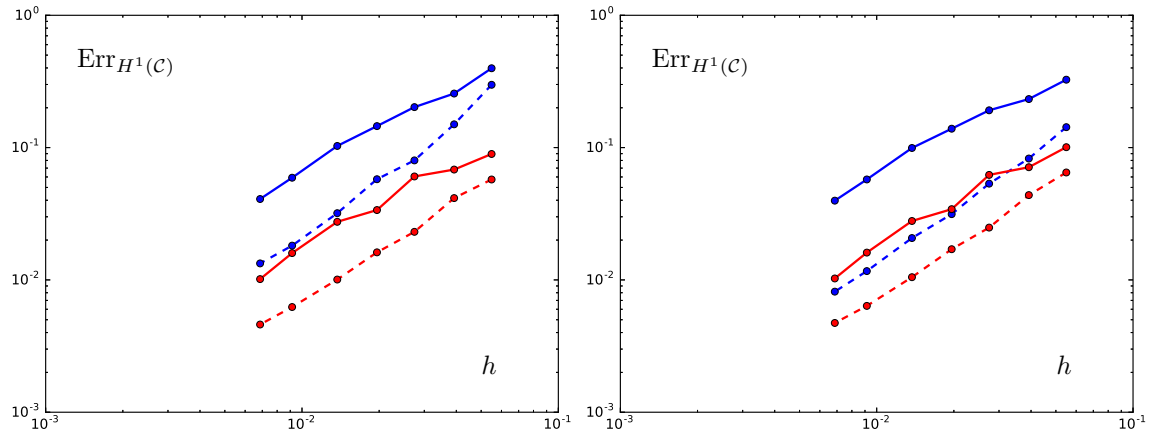


Figure 4.22 – *Relative H^1 errors on the diffusion test case as a function of the smoothing length of the non-stabilized and stabilized weak formulation. We used the corrected linear DMLS gradient, on the Halton distribution (plain lines) and the blue noise distribution (dashed lines). Left: Dirichlet boundary conditions are imposed on the whole boundary. Right: Neumann conditions are imposed on the whole boundary except for a contiguous arc of* $20°$ *(i.e.* $\approx 5.56\%$ *of the boundary).*

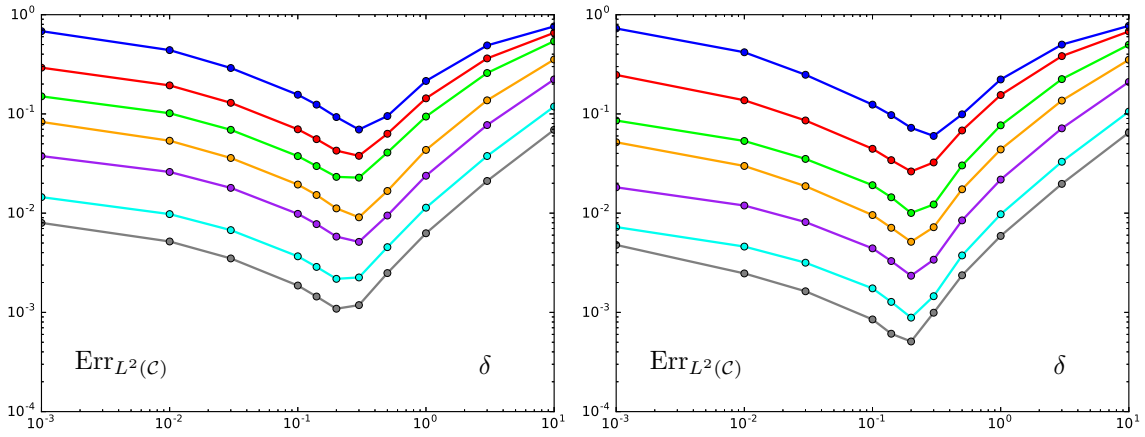


Figure 4.23 – *Relative L^2 errors on the diffusion test case as a function of the stabilization coefficient δ on the Halton (left plot) and the blue noise (right plot) distributions. The smoothing length is set to $h = 5.5 \times 10^{-2}$, 3.9×10^{-2}, 2.7×10^{-2}, 1.9×10^{-2}, 1.4×10^{-2}, 9.2×10^{-3} and 6.9×10^{-3}.*

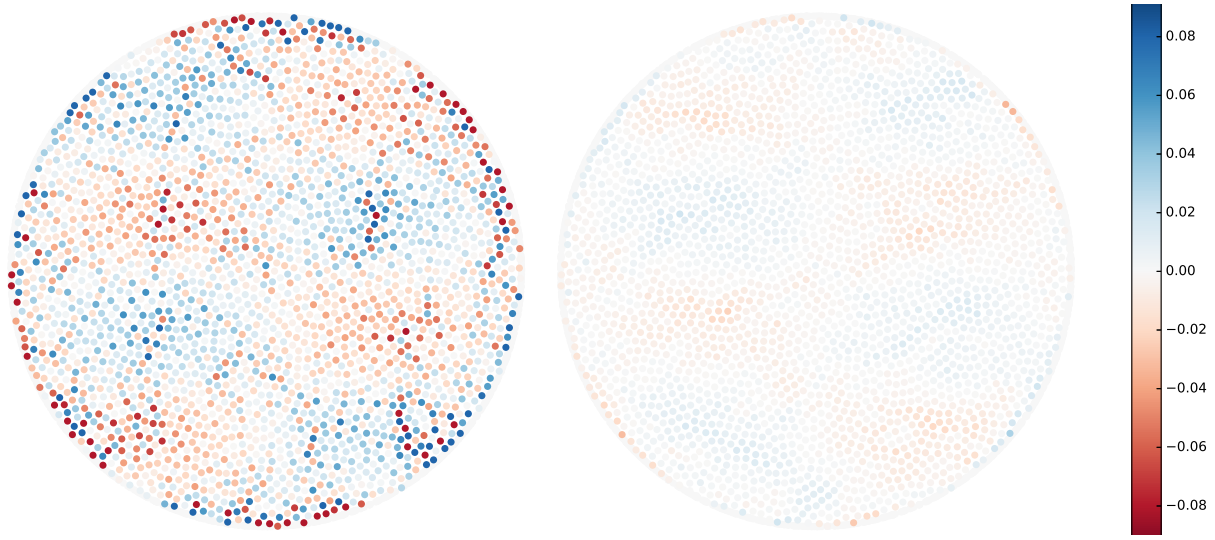


Figure 4.24 – *Node-wise signed error for the smallest point cloud using Dirichlet boundary conditions and the corrected linear DMLS gradient. Left: non-stabilized formulation. Notice that the error varies with a high space-frequency pattern (alternating blue-red colors for neighboring nodes) characteristic of a mild instability. Right: stabilized formulation with $\delta = 3 \times 10^{-1}$. The error is much lower in absolute value and the high-frequency pattern has disappeared.*

convergence is achieved when using the first order consistent DMLS gradient. In contrast, the zeroth order renormalized SPH displays what appears to be a pre-asymptotic phase where the error decreases as h decreases, but then appears to stagnate around a few percents. This behavior is very similar to the one observed on the case where compatibility is achieved *via* adaptation of the nodal positions with the volume-based SPH operators (see left plot of figure 4.14). While this behavior is still an improvement over the non-corrected gradient which systematically leads to a higher error, the only formulation that exhibits optimal convergence is the compatible first order consistent formulation, namely the formulation that passes the linear patch test.

Figure 4.24 represents the signed point-wise error of the first order consistent compatible discretization on the diffusion test-case. Its unphysical high frequency pattern (alternating blue and red dots) suggests the introduction of an additional smoothing stabilization term in the discrete formulation. We thus added the consistent stabilization term defined in expression (4.33) with $\alpha = 0.3$. Visually, its effect is to both smooth out and lower the absolute value of the error as seen on the right hand side of figure 4.24.

On figure 4.21, we compare the evolution of the relative L^2 errors of the stabilized and unstabilized formulations. These plots reveal that the stabilization term systematically has a beneficial effect on the results, reducing the error by a factor of eight to ten in the case of Dirichlet boundary conditions, and by a factor of two to four in the case of the Neumann boundary conditions. Actually, adding this stabilization term has an even more significant impact on the result than going from the Halton distribution to the blue noise distribution.

Very similar observations can be made concerning the approximation of the fluxes as pictured on figure 4.22 which compares the relative H^1 errors of the stabilized and unstabilized formulations for patch-test compliant operators. In all cases, the measured convergence behavior of the flux is at least $\propto h^1$, which is what we expect of a first order formulation.

On figure 4.23, we compare the error level for several values of the stabilization coefficient δ. This plot shows that a compromise needs to be found between under-stabilization (left side of the plot, the error tends to a finite value: the result of the unstabilized case), and over-stabilization (the solution is constrained to be linear, and the error is $\approx 100\%$). The optimal value is confirmed to be independent on the fineness of the point cloud (*i.e.* expression 4.34 is a good scaling) and on the distribution to a large extend.

4.3.5 Linear elasticity: minor theoretical tweaks

As developed in section 4.1.3, the equations of linear elasticity are vector analogs of scalar diffusion. As a consequence, their discretization can easily be obtained using the line of thought described earlier.

At a discrete level, we suppose the gradient operator to act coordinate-wise on vectors. Hence, the discrete gradient of a vector field reads: $\forall\ \mathbf{v} : \mathcal{C} \to \mathbb{R}^d, \forall\ i \in \mathring{\mathcal{C}}$:

$$V_i \mathbb{\nabla}_i \mathbf{v} \overset{\text{def}}{=} \sum_{j \in \mathcal{C}} \mathbf{A}_{i,j} \mathbf{v}_j^T \tag{4.51}$$

With this choice, first order consistency and compatibility still constitute sufficient conditions on the discrete operator to pass the linear patch test. Mixed boundary conditions make the construction of function spaces slightly more involved, but add no real conceptual complexity to the discretization process. Denoting $\mathrm{P}_b^{\perp} = \mathbf{n}_b \mathbf{n}_b^T$ (resp. $\mathrm{P}_b^{\shortparallel} = \mathrm{I}_d - \mathrm{P}_b^{\perp}$) the orthogonal projectors onto the normal (resp. tangent) direction to the boundary at $\mathbf{x}_b$, we define the discrete equivalent of $H^{1,d}_{0,D^{\perp},D^{\shortparallel}}(\Omega)$ as:

$$H^{1,d}_{0,D^\perp,D^{\shortparallel}}(\mathcal{C}) = \{\mathbf{u}:\mathcal{C}\to\mathbb{R}^d \mid \begin{aligned} &\forall\, b=(i,o)\in\partial\mathcal{C}\backslash\partial\mathcal{C}^\perp_D,\ \mathrm{P}^\perp_b\mathbf{u}_b = \mathrm{P}^\perp_b\mathbf{u}_i + (\mathbb{V}_i(\mathrm{P}^\perp_b\mathbf{u}))^T(\mathbf{x}_b-\mathbf{x}_i)\\ &\forall\, b\in\partial\mathcal{C}^\perp_D, \qquad\qquad \mathrm{P}^\perp_b\mathbf{u}_b = \mathbf{0} \qquad (4.52)\\ &\forall\, b=(i,o)\in\partial\mathcal{C}\backslash\partial\mathcal{C}^{\shortparallel}_D,\ \mathrm{P}^{\shortparallel}_b\mathbf{u}_b = \mathrm{P}^{\shortparallel}_b\mathbf{u}_i + (\mathbb{V}_i(\mathrm{P}^{\shortparallel}_b\mathbf{u}))^T(\mathbf{x}_b-\mathbf{x}_i)\\ &\forall\, b\in\partial\mathcal{C}^{\shortparallel}_D, \qquad\qquad \mathrm{P}^{\shortparallel}_b\mathbf{u}_b = \mathbf{0} \qquad \}\end{aligned}$$

Let us remark that once again, these equations cannot be overdetermined. Indeed, the coordinate-wise definition of the gradient operator implies a commutation with symmetric tensors, which in turn means that:

$$\mathrm{P}\mathbf{u}_i + (\mathbb{V}_i(\mathrm{P}\mathbf{u}))^T(\mathbf{x}_b-\mathbf{x}_i) = \mathrm{P}(\mathbf{u}_i + \mathbb{V}_i\mathbf{u}^T(\mathbf{x}_b-\mathbf{x}_i)) \tag{4.53}$$

Furthermore, comparing the range and nullspaces of both orthogonal projectors shows that each boundary node contributes d independent equations as well as d unknowns in equation (4.44).

Denoting $\mathfrak{E}_i\mathbf{v} \overset{\text{def}}{=} (\mathbb{V}_i\mathbf{v} + \mathbb{V}_i\mathbf{v}^T)/2$, the discrete weak formulation reads: $\forall\,\mathbf{v}\in H^{1,d}_{0,D^\perp,D^{\shortparallel}}(\mathcal{C})$,

$$\begin{cases} \cancel{\int\!\!\int}_{\mathcal{C}} \mathfrak{E}\mathbf{u} : \mathbb{H} : \mathfrak{E}\mathbf{v} = \cancel{\int\!\!\int}_{\mathcal{C}} \mathbf{s}\cdot\mathbf{v} + \int\!\!\!\!\square_{\partial\mathcal{C}^\perp_N} g^\perp\mathbf{n}\cdot\mathbf{v} + \int\!\!\!\!\square_{\partial\mathcal{C}^{\shortparallel}_N} \mathbf{g}^{\shortparallel}\cdot\mathbf{v} \\ \mathbf{u} - (u^\perp_D\mathbf{n} + \mathbf{u}^{\shortparallel}_D) \in H^{1,d}_{0,D^\perp,D^{\shortparallel}}(\mathcal{C}) \end{cases} \tag{4.54}$$

The analysis of the patch test is again identical to that presented in section 4.3.4 and first order consistency and compatibility still constitute sufficient conditions.

Numerical results on the elasticity equation

In this section, we consider the classical example of a finite square plate of half width $b = 1$ m pierced in its center with a circular hole of radius $a = 0.2$ m loaded in unidirectional tension normal to one side, as pictured on figure 4.25. The analysis is performed using an isotropic elasticity tensor:

$$\mathbb{H} = \lambda \mathrm{I}_d \otimes \mathrm{I}_d + 2\mu\mathbb{I}_d \tag{4.55}$$

There are no body forces and there are only Neumann boundary conditions in two dimensions, so that the theoretic stress profile does not depend on the actual values of λ and μ.

Using symmetry, we only discretize the problem on one fourth of the geometry, which reduces the size of the problem and allows our discrete displacement-based formulation to be well-posed even in the absence of Dirichlet boundary conditions in the initial model.

Figure 4.26 compares the stress profile as computed by our meshless method and the P-1 finite element method. We clearly recognize a tensile stress concentration (in dark blue) above the hole as well as a compression dominated area (in red) as predicted by the theory.

Our meshless model is generated using a covering point cloud following a blue noise distribution and the clipping process exposed earlier. The final point cloud is made of 1235 interior nodes (2470 degrees of freedom in the final formulation) and 645 boundary nodes. For the simulation, we use the corrected MLS gradient with stabilization ($\delta = 0.3$). The maximal tensile stress is computed to be ≈ 3.22 times the prescribed boundary stress, *i.e.* $\approx 2.3\%$ more than the theoretical value of 3.13. (which includes the finite width correction given in [185]).

The finite element method on a triangular mesh is made of 2111 cells and 1113 nodes for a total of 2226 degrees of freedom for displacements. The overall aspect of the stress

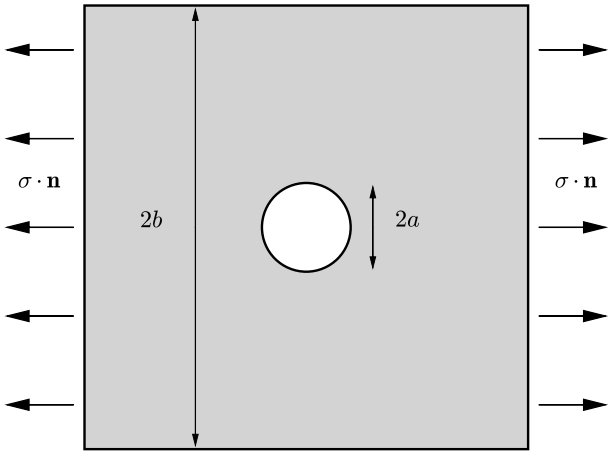


Figure 4.25 – *Geometry of the stress concentration approximation test-case.*

profile looks very similar to our computation. With this method, the maximal tensile stress is computed to be ≈ 3.38 times the prescribed boundary stress, *i.e.* an overestimation of $\approx 8.0\%$ compared to the theoretical value. This example illustrates well the similarities of approximation power between the two methods.

4.4 Comparison of the Nitsche and classical discretizations on a linear elasticity test case

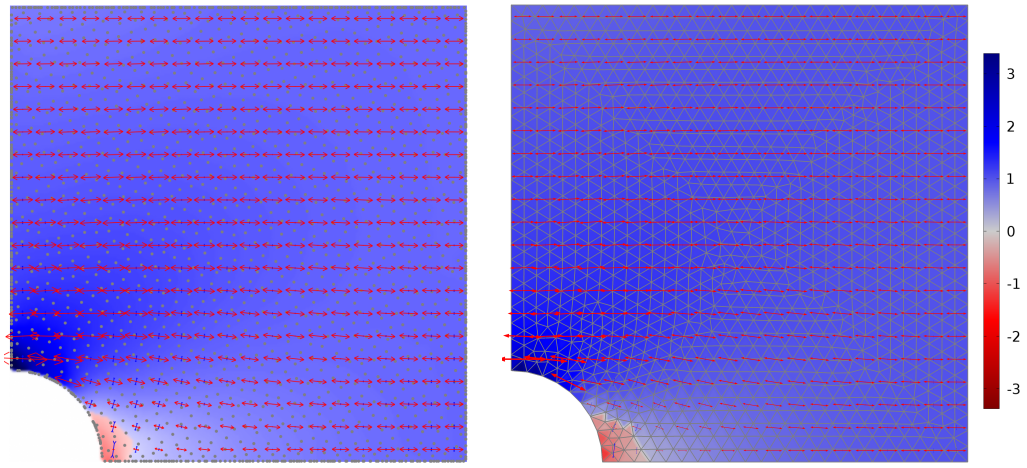
In this section, we compare our two propositions for the imposition of boundary conditions in a nodal integrated meshless framework on a common test case: the evaluation of the stress intensity factor of a two-dimensional finite square plate with centered crack under uniform uniaxial tensile stress treated in the linear elastic regime The exact geometry of the test case can be found in figure 4.27. The ratio of crack length to square side is chosen to be $\frac{a}{b} = 0.2$. The elasticity tensor is assumed isotropic, and the Lamé coefficients (λ, μ) (see expression (4.55) of the elasticity tensor) can be written in terms of the Young modulus Y and Poisson ratio ν as:

$$\lambda = \frac{Y\nu}{(1+\nu)(1-(d-1)\nu)} \qquad \mu = \frac{Y}{2(1+\nu)} \tag{4.56}$$

The difficulty in this test case lies with the non-regularity of the cracked geometry, which is conveyed to the solution. At the tip of the crack, the stress concentration on such a test case cannot be characterized by the ratio $\frac{\sigma_{\max}}{\sigma_\infty}$ as in the pierced plate test case of section 4.3.5. Indeed, the stress tensor of the exact linear elastic solution is typically unbounded in the vicinity of crack tips. Instead, the theory predicts the finiteness of the following limit, written in polar coordinates:

$$K \overset{\text{def}}{=} \lim_{\substack{r\to 0 \\ \theta=0}} \sqrt{2\pi r}\sigma \tag{4.57}$$

Consequently, accurately capturing the stress profile in the vicinity of the tip means being able to derive an accurate estimate of the so-called "stress intensity factor" K (see [21] for more information about stress intensity factors, their relation to energy release rates as well



a) b)

Figure 4.26 – *Simulated stress profile in a finite plate with circular hole in horizontal tension. Only a quarter of the physical domain is simulated and symmetry (mixed: Dirichlet in the normal direction and Neumann in the tangent direction) boundary conditions are applied on the left and bottom side of the simulation domain. Homogeneous Neumann boundary conditions are applied to the top side and the circular portion, and a normal traction is applied on the right side, as sketched on figure 4.25.*
The corresponding discretization structure is shown in gray. The background color scales with the stress eigenvalue of largest absolute value and the arrows represent the stress tensor in an orthonormal basis in which it is diagonal.
a) The proposed meshless method.
b) Linear finite element method of Comsol [1].

as fracture theory in general). If the plate is assumed infinite (*i.e.* $\frac{a}{b} = 0$), a classical result of bi-dimensional elasticity theory states that the stress intensity factor is exactly $K = \sigma\sqrt{\pi a}$. Taking finite effects into account, Rooke and Cartwright [197] give the following approximate correction:

$$K = \sigma\sqrt{\pi a}\frac{1 - \frac{a}{2b} + 0.326\left(\frac{a}{b}\right)^2}{\sqrt{1 - \frac{a}{b}}} \tag{4.58}$$

For $\frac{a}{b} = 0.2$, the relative magnitude of the correction is $\approx 2\%$.

Similarly, it is possible to relate the stress intensity factor and the normal displacements near the crack tip (see [113] for instance). In two dimensions, (or equivalently for a 3-dimensional plane stress approximation), this reads (in polar coordinates):

$$K = \lim_{\substack{r\to 0 \\ \theta=\pi}} \frac{Y}{4}\sqrt{\frac{2\pi}{r}}\mathbf{u}\cdot\mathbf{n} \tag{4.59}$$

There are two main classical methods to estimate the stress intensity factor. The first method is quite primitive: it consists in approximating one of the limits (4.57) or (4.59) directly. This second expression is easier to evaluate with the method developed in section 4.3.4 since the displacements are readily available at boundary "nodes", but not the stresses. These boundary nodes are apparent on the left plot of figure 4.28. Since we do not have

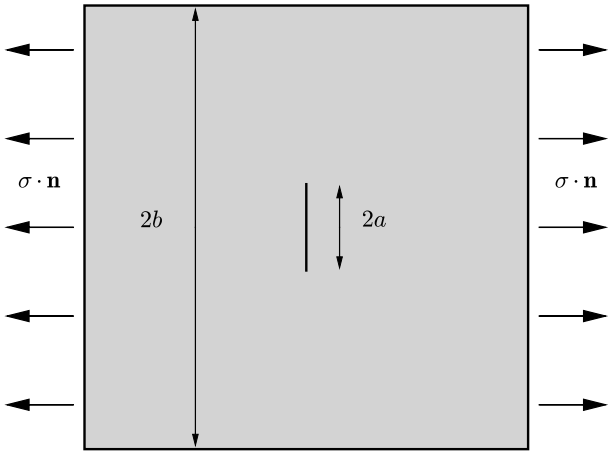


Figure 4.27 – *Geometry of the stress intensity factor approximation test-case.*

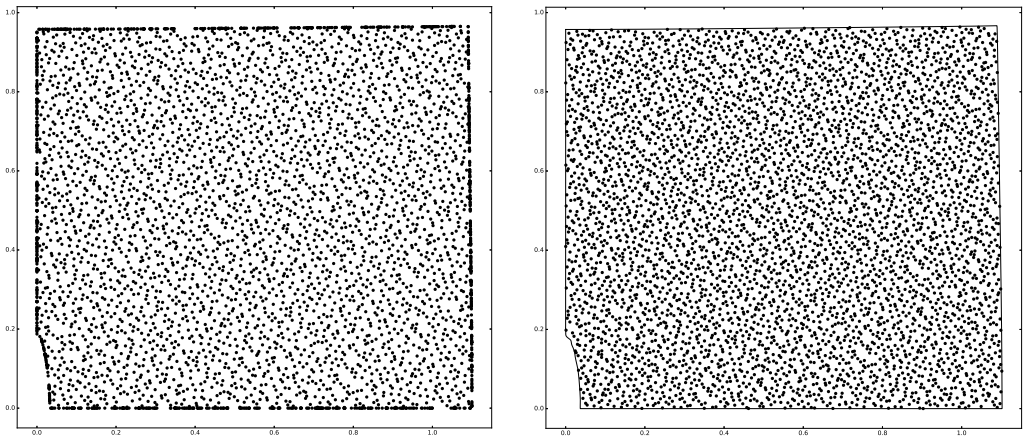
nodes on the boundary with the nodal Nitsche formulation of section 4.2.1, we need to use the reconstructed displacements. The right plot of figure 4.28 shows the displaced nodes and the MLS reconstruction of the boundary.

In order to numerically evaluate the limit (4.59), we extrapolate the RHS of expression (4.59) from the segment $[\varepsilon_1, \varepsilon_2]$ h away from the point $r = 0$ with $\theta = \pi$. The choice of the reconstruction method (especially that of the non-dimensional parameters r_1 and r_2) needs to accommodate the following concerns:

- By definition, $\varepsilon_1 < \varepsilon_2$.
- If the extrapolation segment is too close to the singularity (low values of ε_1), the errors in the displacements are greatly amplified by the factor $\sqrt{\frac{2\pi}{r}}$.
- If the extrapolation segment is too narrow ($\varepsilon_2 - \varepsilon_1$ small), the reconstruction is very sensitive to any error, and thus might not be very robust.
- If the extrapolation segment is too far from the singularity (high values of ε_2), the local behavior of the displacement field noticeably deviates from the theoretical local prediction $\mathbf{u} \propto \sqrt{r}\mathbf{e}_x$.
- The extrapolation method should be robust and precise enough. Here, we compare the LLS reconstructions (see section D.1) of order 0, 1 and 2.

The second method to compute the stress intensity factor is to exploit its link with the energy release rate during crack growth. For instance, via the evaluation of the J-integral, a path independent quantity first investigated by Rice et al. in [195], or the dual I-integral of Bui [49] (see [48] for a comparison of both integrals). In our test case, their common value is $\frac{1}{Y}K^2$.

This method found a new light with the work of Destuynder et al. (see [75, 76]), who proposed to consider the derivative of the potential energy of the system with respect to geometrical changes (opening of the crack). A very clear introduction to this so-called "theta method" and generalizations can be found in the thesis by Suo [219] (in French). In particular, its theorem 2.3 gives the derivative of the mechanical energy in the system $E = \frac{1}{2}\int_\Omega \mathcal{E}\mathbf{u} : \mathbb{H} : \mathcal{E}\mathbf{u}\ \mathrm{d}V - \int_{\partial\Omega} \mathbf{f}\cdot\mathbf{u}\ \mathrm{d}S$ with respect to the domain Ω contracted with a



a) b)

Figure 4.28 – *Simulated displacements on the centered crack test case. The two simulation are performed on Halton distributions with* 4900 *initial nodes. The symmetry of the problem is exploited so that only one fourth of the geometry needs to be discretized.*
a) Results using the method described in section 4.3.4 for the essential imposition of boundary conditions using the corrected DMLS gradient.
b) Results using the method described in section 4.2.1 for the weak Nitsche imposition of boundary conditions using volume-based mesh-integrated SFEM operators.

vector field $\boldsymbol{\theta}$ as:

$$G = -\frac{\partial E}{\partial \Omega} \cdot \boldsymbol{\theta} = \int_\Omega \left(\mathrm{Tr}((\mathbb{H} : \mathcal{E}\mathbf{u}) \nabla \mathbf{u} \nabla \boldsymbol{\theta}) - \frac{1}{2} \mathrm{Tr}(\mathcal{E}\mathbf{u} : \mathbb{H} : \mathcal{E}\mathbf{u}) \nabla \cdot \boldsymbol{\theta} \right) \mathrm{d}V = \frac{1}{Y} K^2 \quad (4.60)$$

Even though the structure theorem (see theorem 3.1) states that the above expression can be written as a boundary integral of the form $\int_{\partial\Omega} \psi(\mathbf{u}) \boldsymbol{\theta} \cdot \mathbf{n} \, \mathrm{d}S$ (such a formula is given in expression (2.35) of [219] for instance), the above expression is more convenient for numerical evaluation since the virtual displacement field $\boldsymbol{\theta}$ only appears through its gradient. Thus, $\boldsymbol{\theta}$ can be designed so that the non-vanishing contributions are not in the direct vicinity of the crack tip.

The above expression is quite convenient for a direct evaluation in our meshless context since discrete integration and differentiation operators are readily available, but the result does depend on the specific choice of virtual displacement field $\boldsymbol{\theta}$. In our tests, we compare several displacement fields representing the same infinitesimal boundary displacement: nothing moves, and the crack opens straight. Specifically, denoting $r = \|\mathbf{x} - \mathbf{x}_{\mathrm{tip}}\|$, we used:

$$\boldsymbol{\theta} = \begin{cases} \mathbf{e}_y & \text{if } r < r_1 \\ \left(1 - \left(\dfrac{r - r_1}{r_2 - r_1}\right)^2\right)^2 \mathbf{e}_y & \text{if } r_1 < r < r_2 \\ \mathbf{0} & \text{if } r_2 < r \end{cases} \quad (4.61)$$

With this definition, $\boldsymbol{\theta}$ is C^1 and has a compact support. Its gradient is non-vanishing only in the annulus $r_1 < r < r_2$, and only nodes in this region contribute to the computation

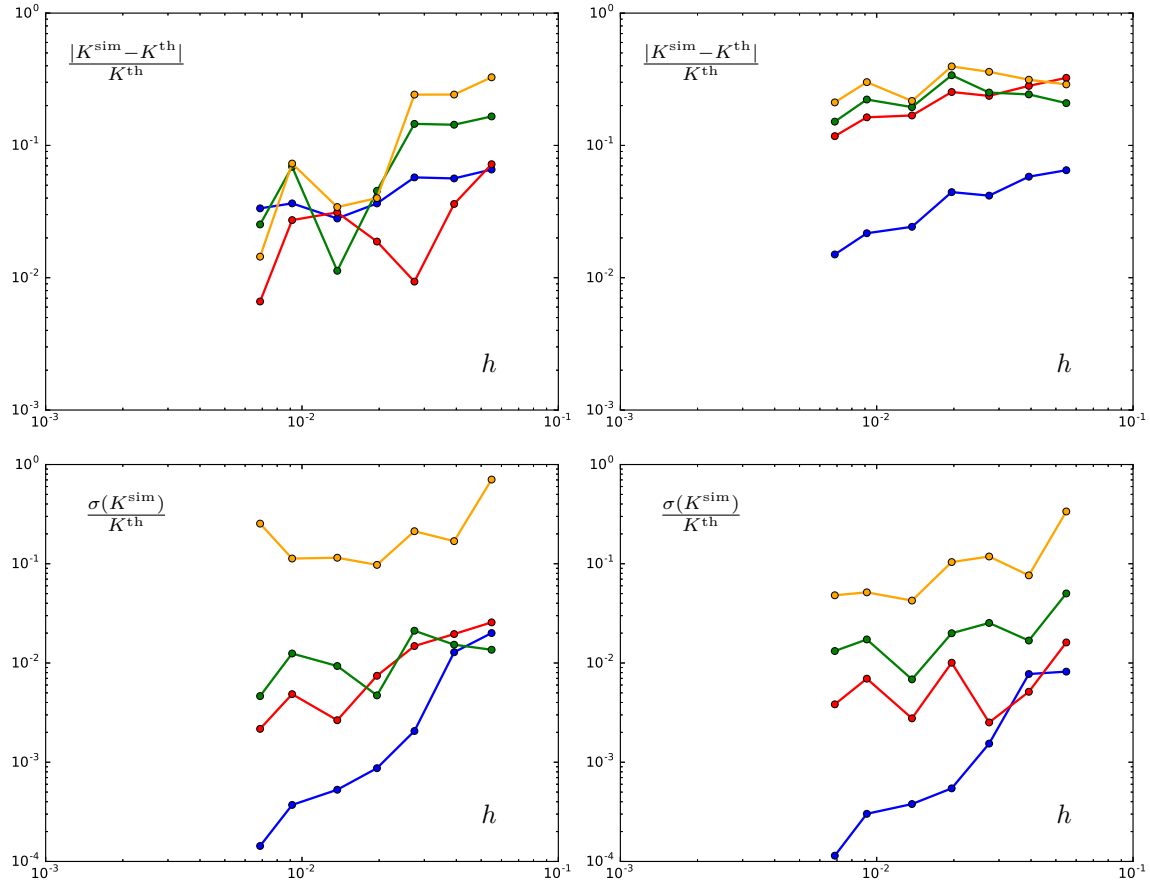


Figure 4.29 – *Numerical comparison of the different methods for the approximation of the stress intensity factors. The simulation was performed on regular blue noise point clouds.*
Top row: Relative error between the theoretical prediction and the meshless approximation of the stress intensity factor as a function of the smoothing length.
Bottom row: Relative standard deviation on the approximation of the stress intensity factor amongst all parameter values as a function of the smoothing length. Low values means that the estimation is not very sensitive to the parameters, and that fine-tuning is not vital.
Left column: Essential boundary conditions of section 4.3.4 using the corrected DMLS gradient.
Right column: Weak Nitsche boundary conditions of section 4.2.1 using volume-based mesh-integrated SFEM operators.
Blue curve: Average value of the stress intensity factor using the theta method with parameters $r_1 \in \{0.05, 0.06, 0.07, 0.08, 0.09, 0.1\}$ *and* $r_2 \in \{0.12, 0.13, 0.14, 0.15, 0.16, 0.17\}$.
Red, green and orange curves: Average value of the stress intensity factor using the displacement reconstruction method. The reconstruction method is the LLS of order zero, one and two, with parameters $\varepsilon_1 \in \{1, 1.2, 1.4, 1.6, 1.8, 2\}$ *and* $\varepsilon_2 \in \{2.4, 2.6, 2.8, 3.0, 3.2, 3.4\}$.

of the stress intensity factor. Similarly to the previous method, we want this region to be far enough from the crack tip so as no to suffer too much from this low accuracy region (r_1 not too small), to be big enough to average possible stress inaccuracies and avoid high gradients of $\boldsymbol{\theta}$ ($r_2 - r_1$ not too small), but not so large as to include irrelevant information (other singularity or other boundary conditions in particular).

On the top row figure 4.29, we compare the relative errors on the approximation of the stress intensity factor between the essential and the weak imposition of boundary conditions. Since these methods are expressed in terms of the arbitrary parameters ε_i and r_i, each curve is an average of the results obtained using several choices for those parameters.

For the simulation using the essential imposition of boundary conditions (left plot), the accuracy of the result is the best in the case of the LLS reconstruction of the displacements of order zero. The second best is the theta method, and then come the reconstruction-based approximations of order 1 and 2. The error levels are rather oscillatory from one simulation to another for all reconstruction-based methods contrary to the theta method, which does not present such a strong oscillatory profile.

For the simulation using the Nitsche-type boundary conditions (right plot), the error levels are far less oscillatory. The most precise method is clearly the theta method, which reaches levels of accuracy comparable to the simulation using essential boundary conditions. The reconstruction-based approximation gives a higher error than the theta method, and the error increases with the order of the reconstruction.

The bottom row of figure 4.29 shows the relative standard deviation of the approximation of the stress intensity factor on the set of parameters that we used. This is a way of measuring the sensitivity of the results with respect to parameters. For both methods of imposition of boundary conditions, the theta method is the least sensitive, then come the reconstruction-based approximations in increasing order. For this reason, and also considering its low error levels, the theta method appears to be a better way of estimating the stress intensity factor.

A possible explanation for the better result of the reconstruction-based method in the case where boundary conditions are imposed essentially is that the displacement information is readily available at the boundary of the domain (via the definition of boundary nodes). On the other hand, our method for imposing the Nitsche boundary conditions evaluates discrete integrals with the field value of nodes near the boundary, but not exactly at the boundary, so that a transition between the two is needed: the MLS reconstruction. The final approximation is an extrapolation of a reconstruction, possibly explaining the loss in accuracy.

A quick summary

This last chapter is dedicated to the development of numerical methods for the discretization of elliptic PDEs using the meshless operators developed in earlier sections. For each of the discretizations, we detail patch test conditions and give numerical evidence that passing the linear patch test, *i.e.* having order one consistency on the final system of discrete equations, is tightly linked with second order convergence. This is consistent with the hypothesis of the Bramble-Hilbert lemma. Note that all tests are performed on non-regular point clouds (and background integration meshes when relevant), *i.e.* on structures with non-periodic substructures such as low discrepancy sequences of points or blue noise distributions.

For the most part, we restrict ourselves to weak nodal integrated meshless methods. In this case, necessary conditions for the linear patch test are first order consistence of the primal gradient, and satisfaction of a discrete analogue of Stokes's theorem. This is the reason why chapters 2 and 3 mainly focus on the concept of compatibility. The case of

meshless Galerkin discretization with more general integration is quickly treated in section 4.2.2, and the conclusions are the same.

Regardless of the chosen method to enforce boundary conditions, we numerically demonstrate that it is indeed crucial to ensure low values of the compatibility error to observe second-order convergence of elliptic systems. More precisely, this error should be kept bounded as the point cloud is refined, contrary to its usual diverging behavior. Consequently, compatibility is not *stricto sensu* required for optimal convergence: this is the most important result of this chapter as it suggests a way out of the meshless curse, which only concerns *exact* compatibility.

Using the weak Nitsche formulation to impose Dirichlet boundary conditions is extremely straightforward in our nodal meshless framework: we simply replace every continuous operator with their discrete counterpart to form the discrete system. This formulation presupposes the availability of a first consistent (quasi-)compatible set of discrete operators, and is thus particularly well-suited to our classical and volume-based SFEM operators.

Here is how the two compare: the classical SFEM operators show signs of a mild instability, but are robust regarding under-integration, which can always be mitigated using an adequate stabilization. On the contrary, volume-based SFEM operators do not require any additional "bulk" stabilization (as opposed to boundary stabilization, which is compulsory in both cases) if the background integration structure is fine enough. However, the effects of under-integration are very sudden, and systematically leads to a dramatic failure of the discretization system, no matter the amount of added stabilization. Interestingly enough, a compatible integration of the Element-Free Galerkin reconciles the good behaviors of the two sets of SFEM discrete operators and avoid their shortcomings.

We also explore the possibility of imposing Dirichlet boundary conditions in an essential way. The most straightforward approach is quite inconvenient since it forces boundary nodes to be placed on the geometrical boundary of the domain. We propose an alteration of the meshless framework that completely alleviates this restriction thanks to the definition of boundary edge positions and a discrete trace operator. These boundary positions can easily be constructed as an intersection between an extended point cloud and the computational domain. Provided we can build compatible first order consistent operators on this new structure, we show that the optimal order of convergence is retrieved, and the error level is shown to be of the same order of magnitude as that of a mesh-based linear finite element method on a linear elasticity test case.

We compare the classical and the Nitsche formulations for the imposition of the boundary conditions on a Stress Intensity Factor (SIF) evaluation test case. The SIF is approximated using two methods, either exploiting the normal boundary displacements at the tip of the crack (reconstruction based method), or the stress distribution in a patch around the crack tip (theta method). In a nutshell, the essential formulation is more reliable for the evaluation of boundary displacements, and the weak Nitsche formulation performs slightly better for the evaluation of stresses. In both case, the theta method has the lowest variability, and is thus the recommended alternative.

Conclusions

The main focus of this thesis has been the meshless discretization of elliptic equations using nodal integrated weak symmetric formulations. We studied them within a novel general meshless nodal operator framework, and clearly showed that regardless of the chosen method to impose boundary conditions, we were able to achieve second order convergence on this type of problems as long as the linear patch test conditions are fulfilled. These conditions can be split into the following simple guidelines: the discrete operators need to be first order consistent and collectively satisfy a discrete analogue of Stokes's theorem, which we called compatibility.

We recognized the difficulty to build such discretizations in a purely meshless context, and materialized it in the following conjectured "meshless curse": effectively computing a set of first order consistent compatible operators requires either some use of a mesh or the solution of a globally coupled system of linear equations. And indeed, any set of such operators studied in this work – or in the literature – fall into either category. Or almost: it can be shown that the exact consistency and compatibility properties of both classical and volume-based SFEM operators can survive the discretization process as long as the approximation for the integrals itself satisfies first order consistency and compatibility. The mesh is simply the only currently known tool that breaks this vicious circle.

We have shown however that exact compatibility is not a strict requirement. In general, methods that succeed in constraining the error on the discrete version of Stokes's formula to a bounded behavior as the point cloud is refined still do allow second order convergence on elliptic problems. In the author's opinion, this is the most promising approach to circumvent the meshless curse and devise efficient and truly meshless discretization of PDEs.

This observation actually led to the reinterpretation of several classical stabilization procedures as actually decreasing the compatibility error, via the addition of either an ALE velocity source term or a uniform background pressure.

The common trait of such methods is that the gradient operator appears via differentiation of the integration volume weights with respect to node positions. In fact, we discovered that the functional dependency of the nodal volume weights with respect to geometrical parameters allows the definition of so-called "volume-based" meshless operators: gradient operator, boundary integration (and resulting dual gradient), and shape functions. Above all, we showed that the properties of volume-based operators could be fully characterized in terms of properties of the original volume weights.

In particular, patch test conditions are fulfilled by sets of meshless operators arising from translation invariant volume functions that are first order consistent for general node positions. This result sheds yet another light on patch test condition compliant operators: although translation invariance (the compatibility part) is fulfilled by virtually any sound volume function, first order consistency for general node position is more difficult to ensure without resorting to some kind of background mesh-based integration procedure. This comes in strong contrast with the usual constructions of meshless gradients, which easily ensure

first order consistency, but struggle with compatibility.

As an important consequence of this characterization, we proved an order limitation result concerning volume-based operators: if the volume function is local (and other technicalities), then it cannot yield a primal gradient consistent of order one. This forbids the existence of purely meshless method with high compatibility orders within this nodal volume-based approach. On the other hand, we proved with an example that volume-based meshless dual gradient can theoretically achieve arbitrary high consistency order.

Once all meshless operators are defined, the discretization of the Nitsche weak form is straightforward. As expected, we showed that SFEM operators are ideally suited to this formulation of elliptic equations, and confirmed experimentally that they achieve a very satisfying second order convergence behavior. We compared our two alternatives (classical and volume-based SFEM), and showed that classical SFEM operators are robust to under-integration, which can always be mitigated using proper stabilization. On the contrary, volume-based SFEM operators do not require any additional "bulk" stabilization, but is less robust to under-integration. Interestingly enough, a compatible integration of the Element-Free Galerkin reconciles the good behaviors of the two sets of SFEM discrete operators and avoid their shortcomings, for a reduced CPU cost and memory footprint. Its interpretation in terms of the definition of mixed operators with non-nodal integration could be an effective approach to defeat the meshless curse.

On these tests, the geometry of the problem and the background compatible integration procedure were provided by a mesh. The extension to a fully meshless procedure requires new structures to replace both these needs. We demonstrated that our meshless framework could be augmented to capture the geometry of the computational domain using traversing edges of a covering point cloud. The key conceptual steps of our method are to effectively sever the communication between inside and outside node, and define a discrete trace operator to map discrete values from the interior point cloud to the boundary, allowing the definition of discrete Sobolev spaces to be used in classical weak formulation. We showed that compatible operators could be built on this graph via the resolution of a global set of linear equations.

We numerically demonstrated that the resulting boundary displacements in the simulation of the opening of a crack led to a more precise estimation of the stress intensity factor than with the Nitsche weak form. Still, the theta method proved a more reliable alternative with both Nitsche and classical formulations, confirming the accuracy of the evaluation of the stresses in the vicinity of the crack tip.

Perspectives and future work

Several practical aspects of the meshless simulation have not been treated during our PhD, and are left to future investigations. In the short term, we would still like to assess the meshless capabilities for local and/or adaptive refinement, and give quantitative answers to the following questions:

- How sharp can the transition be between a dense and a less populated region of the point cloud?
- How to locally accommodate the background integration procedure to changes in the discretization fineness?
- What is the consequence of adding or removing a node from the point cloud in terms of computational effort and memory management?
- What precautions should we take when treating anisotropic distributions of nodes with a tensorial definition of the smoothing length?

In a more distant future, we would also like to study the possibility to rely at little as possible on the background integration mesh. For instance, we could use an adaptive locally Cartesian grids in the bulk of the domain, coupled with third-party representations of the boundary (using level-set methods for instance). In the vicinity of the boundary, the geometry of the cut cells could be effectively described with a cell-tuple structure. Even though the results of our work are extremely promising, we are still a long way off the fully meshless simulation of metal manufacturing processes involving ductile fracture (like riveting for instance), which will require the addition of several ingredients. Among them, let us cite:

- The transition to transient mechanics. This should not pose too many problems for the bulk equations, especially for volume-based formulations, as this discretization process is rather mature for fluid simulations (see appendix G).
- The treatment of boundary conditions for transient problems is a bit more subtle: the main missing ingredient is a dynamic representation of the boundary. A promising approach could be to exploit the flexibility and versatility of level-set functions, which have already shown their ability to represent complex changes in geometry and topology.
- An efficient contact (including self-contact) algorithm: we should avoid at all costs the "numerical fusion" of two parts brought close to each other.
- The choice of a plasticity model.

Several theoretical aspects of meshless methods also call for more work. Of course, the first example that comes in mind is a deeper understanding and a more precise formulation of the meshless curse, since it is the most critical theoretical hurdle to design efficient fully meshless simulation methods. We believe that the next step in the history of meshless methods can only be written if this conjecture is either proved or if a counter-example is found. Going even further, our work has also not been able to exhibit sets of meshless operators with high (*i.e.* first and more) order of compatibility in addition to consistency – that are not the result of the solution of a global linear system. We believe that such operators do exist even though we proved that they cannot be generated with locality properties within the volume method.

On a more modest theme, it would be desirable to design a Lagrangian-based meshless discretization of diffusion and elasticity. The main ambition is to be able to use or adapt the volume method to achieve discrete conservation of global energy as well as linear and angular momenta akin to those of the Lagrangian formulation of hydrodynamics. Moreover, this would readily yield a Noetherian theorem relative to the variation of the boundary, giving a fully discrete interpretation of the theta method and the design of energy-based fracture criteria.

Appendix A

Sobolev spaces and the Bramble-Hilbert lemma

In this appendix, we introduce a few notions from Sobolev space theory needed to show the convergence of the mesh-based integration formulae given in section 1.3.2.

Definition A.1 (Norms and semi-norms in Sobolev spaces)**.**
Let Ω be a bounded domain in $\mathbb{R}^n$ with boundary $\partial\Omega$, $k \in \mathbb{N}$ and $p \in \mathbb{N}^* \cup \infty$. We denote $W_p^k(\Omega, Y)$ the Sobolev space of all functions defined on Ω with values in a normed Banach space Y with weak derivatives $\frac{\partial}{\partial^\alpha x}$ of order $|\alpha| \leq k$ in $L^p(\Omega, Y)$. The classical Sobolev norms and semi-norms are defined as: $\forall\ u \in W_p^k(\Omega, Y)$,

$$\|u\|_{W_p^k(\Omega,Y)} \stackrel{\text{def}}{=} \left(\sum_{|\alpha|\leq k} \left\| \frac{\partial u}{\partial^\alpha x} \right\|^p_{L^p(\Omega,Y)} \right)^{\frac{1}{p}} \tag{A.1}$$

$$|u|_{W_p^k(\Omega,Y)} \stackrel{\text{def}}{=} \left(\sum_{|\alpha|=k} \left\| \frac{\partial u}{\partial^\alpha x} \right\|^p_{L^p(\Omega,Y)} \right)^{\frac{1}{p}} \tag{A.2}$$

The space of functions whose $W_p^k(\Omega, Y)$ semi-norm vanishes is denoted $P_{k-1}(\Omega)$. It is the space of polynomials of n variables on Ω whose total degree does not exceed $k-1$.

Definition A.2 (Strong cone property)**.**
A domain $\Omega \subset \mathbb{R}^n$ satisfies the strong cone property if there exists an open cover $(\Omega_k)_k$ of Ω and corresponding cones $(C_k)_k$ with vertices at the origin such that for each point $\mathbf{x}$ of $\Omega \cap \Omega_k$, the cone $\mathbf{x} + C_k$ is contained in Ω.

The strong cone property, much like its dual equivalent the Poincaré cone property (which is the cone property on $\mathbb{R}^n \backslash \Omega$) can be understood as a weak convexity requirement as discussed in [61]. The strong cone property is a weaker property than the strong Lipschitz property, and is almost equivalent in the case of bounded domain (in a sense made clear in [5] page 67).

Informally, it ensures that the boundary $\partial\Omega$ does not have any "thin pointy ends". See figure A.1 for a few two-dimensional examples.

Bramble-Hilbert lemma A.1.
Let $\Omega \subset \mathbb{R}^n$ be a bounded domain satisfying the strong cone property. Then there exists

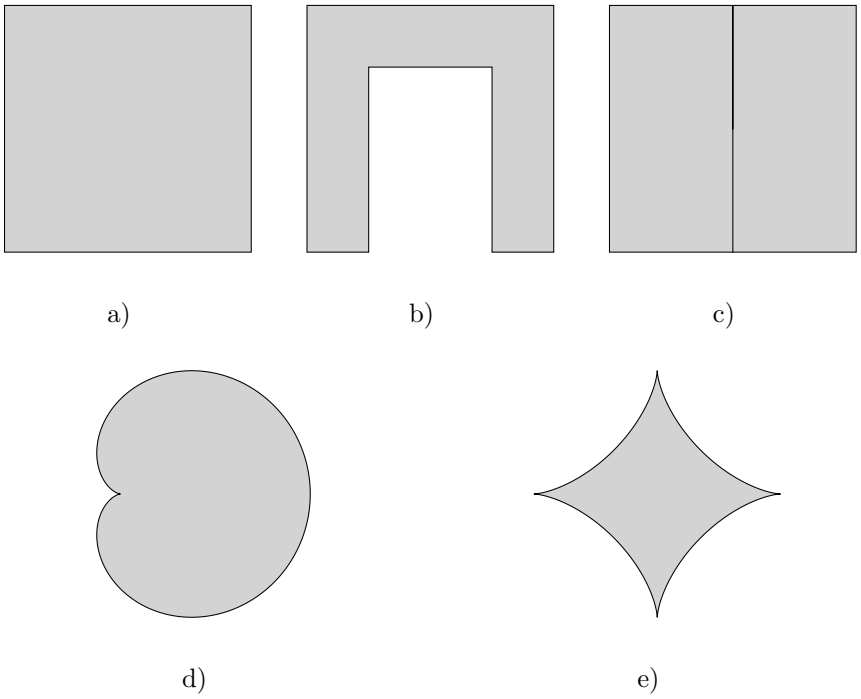


Figure A.1 – *The strong cone condition property means that Ω should not have any "thin pointy ends".*
a) The square is convex with non-empty interior, hence satisfies the strong cone property. In particular, sharp corners are allowed.
b) Non-convex polygons are still strong Lipschitz, hence satisfy the strong cone condition.
c) The slit square satisfies the strong cone condition but is not Lipschitz since it is not locally on one side of its boundary.
d) The cardioid has a thin pointy pit (or cavity), which does not impair the weak cone property, but does not satisfy the strong cone property.
e) The astroid has four thin pointy ends where the strong cone property is not satisfied.

a constant $C(\Omega, m)$ independent of p and u such that for any $u \in W_p^k(\Omega)$, there exists a polynomial $v \in P_{m-1}$ such that for all $k \in [\![0, m]\!]$,

$$|u - v|_{W_p^k(\Omega)} \leq C(\Omega, m)|u|_{W_p^m(\Omega)} \tag{A.3}$$

Proof. See [229] for instance. □

Remark. The dependence with respect to homogeneous scaling $\mathbf{x} \mapsto a\mathbf{x}$ can be made more explicit. Indeed for $a > 0$, let us consider the scaled domain $a\Omega = \{\mathbf{y} \in \mathbb{R}^n \mid \exists\ \mathbf{x} \in \Omega \mid \mathbf{y} = a\mathbf{x}\}$. There, we can define $u_a : a\Omega \to Y$ by the formula $u_a(\mathbf{y}) = u(\frac{\mathbf{y}}{a})$ for $\mathbf{y} \in \Omega$. The derivatives of u_a then satisfy $\frac{\partial u_a}{\partial^\alpha \mathbf{y}}(\mathbf{y}) = \frac{1}{a^{|\alpha|}} \frac{\partial u}{\partial^\alpha \mathbf{y}}(\frac{\mathbf{y}}{a})$, which entails $|u_a|_{W_p^m(a\Omega)} = a^{d-m}|u|_{W_p^m(a)}$. Thus, choosing the scaling factor as the inverse of the diameter of the domain $a = \frac{1}{D(\Omega)}$, we can write the following alternative version of expression (A.3):

$$|u - v|_{W_p^k(\Omega)} \leq C\left(\frac{1}{D(\Omega)}\Omega, m\right) D(\Omega)^{m-k}|u|_{W_p^m(\Omega)} \tag{A.4}$$

The constant $C\left(\frac{1}{D(\Omega)}\Omega, m\right)$ is obviously invariant with respect to homogeneous scaling.

The relevance of the Bramble-Hilbert lemma is best understood when stated under the following alternative form:

Corollary A.2 (Operator form of the Bramble-Hilbert lemma)**.**
Let $\Omega \subset \mathbb{R}^n$ be a domain satisfying the interior cone property and $\ell : W_p^k(\Omega) \to Y$ be a continuous linear operator. Suppose that ℓ vanishes on polynomials of maximal order $m-1$: $\ell(v) = 0 \ \forall \ v \in P_{m-1}$. Then there exists a constant C such that:

$$\|\ell(u)\|_Y \leq C\left(\frac{1}{D(\Omega)}\Omega, m\right) \ \|\ell\|_{\mathcal{L}(W_p^k(\Omega),Y)} \ D(\Omega)^{m-k}|u|_{W_p^m(\Omega)} \tag{A.5}$$

Proof. This is a straightforward combination of expression (A.4) and the continuity of ℓ. □

Under this form, the Bramble-Hilbert lemma is particularly useful to prove error bounds of numerical integration formulae. The sketch of such proofs is always the same: we want to use the Bramble-Hilbert lemma on the error functional ℓ. The hypothesis $\ell(v) = 0 \ \forall \ v \in P_{m-1}$ is a consistency requirement on the discrete operator. Convergence then follows from the regularity of Ω in a norm $|.|_{W_p^m(\Omega)}$for which ℓ is continuous. The exponent $m-k$ is called the order of accuracy of the method, since the transformation $\Omega \mapsto a\Omega$ changes the error bound by a factor a^{m-k}.

For sufficiently simple shapes, the dependency of the constant can be made more explicit. For this, we need the following definition:

Definition A.3 (Strongly star-shaped domains and chunkiness)**.**
A domain $\Omega \subset \mathbb{R}^n$ is *strongly star-shaped* if there exists a ball $B \subset \Omega$ such that the for every point $\mathbf{x}$ of Ω, the convex hull of $B \cup \{\mathbf{x}\}$ is a subset of Ω. In particular, a strongly star-shaped domain satisfies the strong cone condition.
Let us denote $\rho(\Omega)$ the supremum of the diameters of such balls. The following ratio is called the *chunkiness* of Ω.

$$\gamma(\Omega) \stackrel{\text{def}}{=} \frac{D(\Omega)}{\rho(\Omega)} \tag{A.6}$$

The following result relating chunkiness to the bound given in the Bramble-Hilbert lemma was first established by Dupont and Scott in [80].

Proposition A.3 (Dependency with respect to space shape in the Bramble-Hilbert lemma)**.**

With the notations of lemma A.1, if we further assume that Ω is strongly star-shaped, then the reduced constant of expression (A.4) may only depend on Ω through its chunkiness and the dimension of embedding space: $C\left(\frac{1}{D(\Omega)}\Omega, m\right) = C(\gamma(\Omega), m, n)$.

Proof. This result (albeit in a slightly different form) is stated as the Bramble-Hilbert lemma in (4.3.8) of [41]. A complete proof can be found in this book. □

This result can be extended to more general shapes, for instance where Ω is the union of finitely many strongly star-shaped domains (see [80] for more details). Uniform boundedness of the chunkiness of cells in a sequence of meshes is routinely assumed in FEM proofs of convergence in order to use the Bramble-Hilbert lemma with constants that are independent of mesh refinement.

The takeaway of this section is that polynomial consistence of discrete operators play a central role in their convergence proofs using the Bramble-Hilbert lemma. For this reason, consistency is always our prime concern when devising discrete operators in the rest of this work.

Appendix B

Quadratic optimization

In this appendix, we state a few useful results related to quadratic optimization needed for our proofs.

Definition B.1 (Least norm solution of quadratic optimization problem)**.**
Let $\mathrm{A} \in \mathrm{M}_{m,n}(\mathbb{R})$ and $\mathbf{b} \in \mathbb{R}^m$. Solving a set of linear equations means in general finding $\mathbf{x} \in \mathbb{R}^n$ such that the following holds:

$$\mathrm{A}\mathbf{x} = \mathbf{b} \tag{B.1}$$

Depending on the properties of A and $\mathbf{b}$, equation (B.1) may have either no solution, a unique solution or infinitely many solutions. In order to overcome thus indeterminacy issues, it is common to solve the above problem *in the least square sense*, *i.e.* to instead seek $\mathbf{x} \in \mathbb{R}^n$ such that:

$$\begin{cases} \|\mathbf{x}\|^2 & \text{is minimal, under the constraint that} \\ \|\mathrm{A}\mathbf{x} - \mathbf{b}\|^2 & \text{is minimal.} \end{cases} \tag{B.2}$$

Proposition B.1 (Characterization of solutions)**.**
Assuming there exists at least one solution to the system (B.1) (*i.e.* assuming $\mathbf{b} \in \mathrm{Im}(\mathrm{A})$) then, solutions are characterized by orthogonality with the kernel of A: $\mathbf{x}$ is the solution of equation (B.2) if and only if:

$$\begin{cases} \mathrm{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \cdot \mathbf{y} = 0 \qquad \forall\, \mathbf{y} \mid \mathrm{A}\mathbf{y} = \mathbf{0} \end{cases} \tag{B.3}$$

Proof. Let $\mathbf{x}$ be a solution of equation (B.2) and $\mathbf{y}$ such that $\mathrm{A}\mathbf{y} = \mathbf{0}$. Then for all $\varepsilon \in \mathbb{R}^{+*}$, we have $\mathrm{A}(\mathbf{x} + \varepsilon\mathbf{y}) = \mathbf{b}$. Thus by optimality of $\mathbf{x}$, we have:

$$\begin{aligned} 0 &\leq \|\mathbf{x} + \varepsilon\mathbf{y}\|^2 - \|\mathbf{x}\|^2 \\ &= \varepsilon(2\mathbf{x} \cdot \mathbf{y} + \varepsilon\|y\|^2) \end{aligned}$$

Simplifying the above expression by ε and having $\varepsilon \longrightarrow 0$, we get $0 \leq \mathbf{x} \cdot \mathbf{y}$. Repeating the process for $-\mathbf{y}$, we get $\mathbf{x} \cdot \mathbf{y} = 0$.

Reciprocally, suppose that $\mathbf{x}$ is a solution of equation (B.3). Then let $\mathbf{z}$ satisfy the constraint $\mathrm{A}\mathbf{z} = \mathbf{b}$. We have $\mathrm{A}(\mathbf{z} - \mathbf{x}) = \mathbf{0}$ and thus, $\mathbf{x} \cdot (\mathbf{z} - \mathbf{x}) = 0$. Incidentally, we have:

$$\begin{aligned} \|\mathbf{z}\|^2 &= \|\mathbf{x} + \mathbf{z} - \mathbf{x}\|^2 \\ &= \|\mathbf{x}\|^2 + 2\mathbf{x} \cdot (\mathbf{z} - \mathbf{x}) + \|\mathbf{z} - \mathbf{x}\|^2 \end{aligned}$$

$$
\begin{aligned}
&= \|\mathbf{x}\|^2 + \|\mathbf{z} - \mathbf{x}\|^2 \\
&\geq \|\mathbf{x}\|^2
\end{aligned}
$$

And thus, $\mathbf{x}$ is optimal. □

Proposition B.2 (Moore-Penrose pseudo-inverse)**.**
Let $m, n \in \mathbb{N}$ and $\mathrm{A} \in \mathrm{M}_{m,n}(\mathbb{R})$. There exists a unique matrix in $\mathrm{M}_{n,m}(\mathbb{R})$ called the *Moore-Penrose pseudo-inverse* of A and denoted A^+ satisfying the following conditions:

- $\mathrm{A}\mathrm{A}^+\mathrm{A} = \mathrm{A}$
- $\mathrm{A}^+\mathrm{A}\mathrm{A}^+ = \mathrm{A}^+$
- $\mathrm{A}\mathrm{A}^+ = \left(\mathrm{A}\mathrm{A}^+\right)^T$
- $\mathrm{A}^+\mathrm{A} = \left(\mathrm{A}^+\mathrm{A}\right)^T$

Moreover, the set of vectors $\mathbf{x}$ minimizing the cost expression $\|\mathrm{A}\mathbf{x} - \mathbf{b}\|^2$ can be written in terms of the Moore-Penrose pseudo-inverse as:

$$
\arg\min_{\mathbf{x}} \|\mathrm{A}\mathbf{x} - \mathbf{b}\|^2 = \{\mathrm{A}^+\mathbf{b} + (\mathrm{I}_n - \mathrm{A}^+\mathrm{A})\mathbf{z},\ \mathbf{z} \in \mathbb{R}^n\} \tag{B.4}
$$

And the least-norm solution is $\mathbf{x} = \mathrm{A}^+\mathbf{b}$. In the particular case where A has full rank r, the Moore-Penrose pseudo-inverse can be expressed using matrix inversion:

$$
\begin{aligned}
\mathrm{A}^+ &= \left(\mathrm{A}^T\mathrm{A}\right)^{-1}\mathrm{A}^T \qquad \text{if } m \geq n = r \\
\mathrm{A}^+ &= \mathrm{A}^T\left(\mathrm{A}\mathrm{A}^T\right)^{-1} \qquad \text{if } n \geq m = r
\end{aligned}
$$

Proof. See the excellent review paper by Barata and Hussein [17]. □

Proposition B.3 (Derivative of Moore-Penrose pseudo-inverse)**.**
Let $\mathrm{A}, \mathrm{B} \in \mathrm{M}_{m,n}(\mathbb{R})$ be two real matrices and $\mathrm{C} : \mathbb{R} \to \mathrm{M}_{m,n}(\mathbb{R})$ be a differential matrix valued real function such that $\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{C}(t)\big|_{t=0} = \mathrm{B}$. Then, the application $t \mapsto (\mathrm{A} + \mathrm{C}(t))^+$ is continuous at $t = 0$ if and only if $\mathrm{A} + \mathrm{C}(t)$ is of constant rank for all values of t in a neighborhood of the origin. In this case, it is furthermore differentiable, and its differential reads:

$$
\begin{aligned}
\frac{\partial \mathrm{A}^+}{\partial \mathrm{A}} : \mathrm{B} &\stackrel{\text{def}}{=} \frac{\mathrm{d}}{\mathrm{d}t}(\mathrm{A} + \mathrm{C}(t))^+\Big|_{t=0} \\
&= -\mathrm{A}^+\mathrm{B}\mathrm{A}^+ + (\mathrm{I}_n - \mathrm{A}^+\mathrm{A})\mathrm{B}^T\mathrm{A}^{+T}\mathrm{A}^+ + \mathrm{A}^+\mathrm{A}^{+T}\mathrm{B}^T(\mathrm{I}_m - \mathrm{A}\mathrm{A}^+)
\end{aligned}
$$

Proof. See [106] and [215] for the complete proof. □

This proposition generalizes the well-known formula for the derivative of the inverse of a matrix. The two additional terms contain the matrices $(\mathrm{I}_n - \mathrm{A}\mathrm{A}^+)$ and $(\mathrm{I}_m - \mathrm{A}^+\mathrm{A})$ which should be interpreted as orthogonal projectors onto the kernel of A and A^+ respectively (see proposition 3.3 of [17]). If A is invertible $\mathrm{A} \in \mathrm{GL}_n(\mathbb{R})$, these projectors vanish and the two formulae coincide.

Appendix C

Positivity of discrete integration

Positivity of the integration formula is a desirable property with heavy consequences on the stability of simulation schemes. In order to characterize the existence of positive nodal integration schemes with imposed exactness properties, we need the following "alternative result" characterizing the existence of systems of linear equations and inequations. This result is very related to Farkas's lemma [89], Gordan's theorem [107] and Stiemke's theorem [216]. It is a particular case of the general Motzkin's transposition theorem [167, 168]. A short yet detailed account of the topic of alternative theorems in linear algebra can be found in Ben-Israel [30]

Proposition C.1 (Strict alternative result)**.**
Let $\mathrm{A} \in \mathrm{M}_{m,n}(\mathbb{R})$ and $\mathbf{b} \in \mathbb{R}^m$. The following propositions are alternatives:

a. $\exists\ \mathbf{x} \in \mathbb{R}^n \mid \mathrm{A}\mathbf{x} = \mathbf{b}$ and $\mathbf{x} > \mathbf{0}$
b. $\exists\ \mathbf{y} \in \mathbb{R}^m \mid \mathrm{A}^T\mathbf{y} \geq \mathbf{0}$ and $\mathbf{b} \cdot \mathbf{y} \leq 0$ and $\left(\mathrm{A}^T\mathbf{y} \neq \mathbf{0} \text{ or } \mathbf{b} \cdot \mathbf{y} \neq 0\right)$

Equivalently, the following problem: find $\mathbf{x} \in \mathbb{R}^n$ such that:

$$\begin{cases} \mathrm{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} > 0 \end{cases} \tag{C.1}$$

has a solution if and only if we have: $\forall\ \mathbf{y} \in \mathbb{R}^m$

$$\left(\mathrm{A}^T\mathbf{y} \geq \mathbf{0} \text{ and } \mathbf{b} \cdot \mathbf{y} \leq 0\right) \text{ implies } \left(\mathrm{A}^T\mathbf{y} = \mathbf{0} \text{ and } \mathbf{b} \cdot \mathbf{y} = 0\right) \tag{C.2}$$

Proof. Substitute $\begin{pmatrix} \mathrm{A} & -\mathbf{b} \end{pmatrix}$ for A in theorem 6 of [30]. □

The following proposition fully characterizes the existence of positive nodal integration schemes with imposed exactness properties.

Proposition C.2 (Existence of positive discrete integration formulae)**.**
Let us suppose given a point cloud $(\mathcal{C}, \mathbf{X})$ and a continuous function space $\boldsymbol{\varphi} \subset (\Omega \to \mathbb{R})$ of finite dimension $n_\varphi \in \mathbb{N}$ with basis $(\varphi_k)_{k \in [\![1, n_\varphi]\!]}$. There exists a positive discrete nodal integration operator on $(\mathcal{C}, \mathbf{X})$ that is exact on $\boldsymbol{\varphi}$ if and only if all functions of $\boldsymbol{\varphi}$ that are non-negative on $\mathbf{X}$ and have non-positive integral actually vanish on $\mathbf{X}$ and have zero mean on Ω.

More precisely, denoting Φ the matrix of evaluation of basis functions $\Phi_{j,k} = \mathcal{R}(\varphi_k)_j = \varphi_k(\mathbf{x}_j)$, and $I(\varphi)_k$ the value of the integral of the basis function φ_k on the domain $I(\varphi)_k = \int_\Omega \varphi_k(\mathbf{x})\ \mathrm{d}V$, there exists $(V_i)_{i \in \mathcal{C}}$ such that:

• $V_i > 0$	$\forall\, i \in \mathcal{C}$	• $V > 0$
• $\oint_{\mathcal{C}} \varphi = \int_\Omega \varphi(\mathbf{x})\ \mathrm{d}V$	$\forall\, \varphi \in \boldsymbol{\varphi}$	• $\Phi^T V = I(\varphi)$

If and only if the following inequalities (with $\varphi = \sum_k = 1^{n_\varphi} a_i \varphi_k \in \boldsymbol{\varphi}$)

• $\varphi(\mathbf{x}_i) \geq 0$	$\forall\, i \in \mathcal{C}$	• $\Phi a \geq 0$
• $\int_\Omega \varphi(\mathbf{x})\ \mathrm{d}V \leq 0$		• $I(\varphi) \cdot a \leq 0$

imply the following equalities:

• $\varphi(\mathbf{x}_i) = 0$	$\forall\, i \in \mathcal{C}$	• $\Phi a = 0$
• $\int_\Omega \varphi(\mathbf{x})\ \mathrm{d}V = 0$		• $I(\varphi) \cdot a = 0$

Proof. Using the matrix-vector notations of the right hand column, this is a direct application of proposition C.1. □

Example. Let us illustrate proposition C.2 on a simple 1-D case. Suppose we want to characterize first order consistent positive one node integration formulae on segments. In other words, we consider the case $\Omega = [-1, 1] \subset \mathbb{R}$, $(\mathcal{C}, \mathbf{X}) = (\{1\}, \{x\})$ and $\boldsymbol{\varphi} = \mathbb{R}_1[X]$. We seek $V > 0$ such that $V \times (ax + b) = \int_{-1}^{1} ax + b\ \mathrm{d}x$ for all $a, b \in \mathbb{R}$.

Proposition C.2 states that such formulae exist if and only if $\Big(ax + b \geq 0$ and $\int_{-1}^{1} ax + b\ \mathrm{d}x \leq 0\Big)$ implies $\Big(ax + b = 0$ and $\int_{-1}^{1} ax + b\ \mathrm{d}x = 0\Big)$. Let us split the analysis into three distinct cases:

- $x > 0$: $b = -1$ and $a = \frac{1}{x} + 1$ show that no such formulae exists.
- $x < 0$: $b = -1$ and $a = -\frac{1}{x} + 1$ show that no such formulae exists.
- $x = 0$: $a \times 0 + b \geq 0$ and $b \leq 0$ indeed implies $a \times 0 + b = 0$ and $b = 0$. Hence, a discrete integration formula exist. The theorem is non-constructive: the solution is guaranteed to exist, but we still need to find it. In this simple case, we can easily verify that $V = 2$ is the only solution.

Appendix D

Meshless Least Square approximations

In this section, we recall the construction of several locally defined meshless operators: the Local Least Squares (LLS) and MLS reconstruction and the DMLS and Full Moving Least Squares (FMLS) gradients. These approximation techniques can achieve consistency of any order and are probably the most widespread in the meshless literature, and we heavily use them in the rest of this work. According to Wendland, the first published occurrence of the MLS can be traced back to Lancaster and Salkauskas [133] (with particular cases going back earlier *e.g.* [207]). More information can be found in the work of Wendland [232, 233], Fasshauer [90] and Mirzaei et al. [159].

All the gradients and reconstruction operators presented in this appendix can effectively be computed via the resolution of small linear problems, *i.e.* whose dimension does not depend on the total number of nodes in the point cloud. Instead – via the definition of compactly supported weight functions – their computation only involves a few neighboring nodes. The most important take-away of this section is that there exist efficient techniques to compute discrete gradient and reconstruction operators that are consistent of any order. These operators are used throughout this work.

D.1 Local Least Squares approximation

Local Least Squares (LLS) approximations aim at building a continuous local reconstruction $\mathbf{y} \mapsto < f >_{\mathbf{x}}^{LLS} (\mathbf{y})$ of a discrete function $f : \mathcal{C} \to \mathbb{R}$ in a neighborhood of the point $\mathbf{x} \in \Omega$, which best fits the nodal data in a least square sense. The approximation function space is typically the space of $d-$variate polynomials up to a given degree $p \in \mathbb{N}$, in which case the LLS approximation is termed "of order p". The function space may possibly be enriched with a few other functions exploiting *a priori* knowledge of the function to approximate (see [24, 98] for examples of the so-called extrinsic MLS enrichment).

Definition D.1 (Local Least Squares approximation)**.**
Let us suppose given an approximation function space $\varphi \subset \Omega \to \mathbb{R}$ of finite dimension $n_\varphi \in \mathbb{N}$ with basis $(\varphi_k)_{k \in [\![1, n_\varphi]\!]}$. The Local Least Squares (LLS) seeks an approximation of a discrete function around a given fixed point of space $\mathbf{x} \in \Omega$ as a linear combination of the basis functions. Expressed in terms of its coefficients $(c_k(f, \mathbf{x}))_{k \in [\![1, n_\varphi]\!]}$, it necessarily reads:

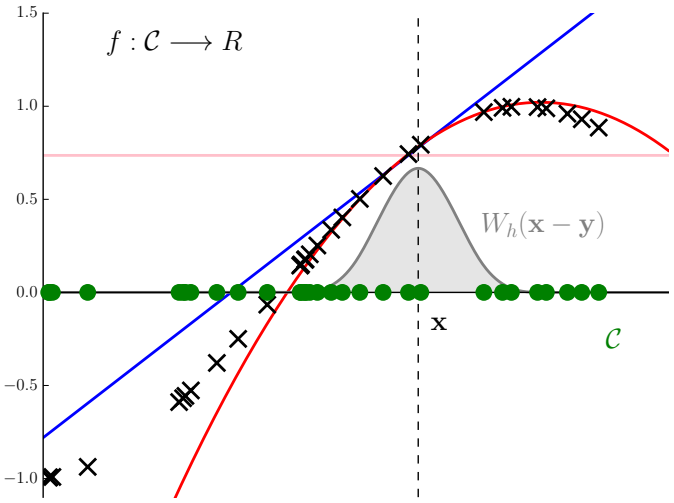


Figure D.1 – *The LLS procedure gives the closest approximation of the scattered data* $(\mathbf{x}_i, f_i)_{i\in\mathcal{C}}$ *in the chosen function space around the point* $\mathbf{x}$.
On the figure, green nodes make up the point cloud $(\mathcal{C}, \mathbf{X})$ *on which a discrete function* $f : \mathcal{C} \to \mathbb{R}$ *is defined by the black markers* $\times$. *The graph shows LLS approximation with weight function* $W_h(\mathbf{x}-\mathbf{y})$ *of order zero, order one and order two.*

$\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$< f >_{\mathbf{x}}^{LLS} (\mathbf{y}) \overset{\text{def}}{=} \sum_{k=1}^{n_\varphi} c_k(f,\mathbf{x})\varphi_k(\mathbf{y}) = \mathbf{c}(f,\mathbf{x}) \cdot \boldsymbol{\varphi}(\mathbf{y}) \tag{D.1}$$

Approximation quality is locally gauged at around $\mathbf{x}$ using a weight function $\mathbf{y} \mapsto W_h(\mathbf{x}-\mathbf{y})$. This weight function typically has compact support of size $h \in \mathbb{R}^+$ with rotational symmetry and is radially decreasing, but we do not impose constraints on the value of its integral contrary to the SPH kernels. The coefficients $(c_k(f,\mathbf{x}))_{k\in[\![1,n_\varphi]\!]}$ are chosen so that the LLS is the best approximation in the least-square sense, *i.e* they form the *minimum norm solution* to the minimization of the following positive quadratic expression:

$$\sum_{j\in\mathcal{C}} W_h(\mathbf{x}-\mathbf{x}_j)\left(\sum_{k=1}^{n_\varphi} z_k\varphi_k(\mathbf{x}_j) - f_j\right)^2 \tag{D.2}$$

Figure D.1 gives a 1-D example of LLS reconstruction.

Proposition D.1 (Closed form of the LLS approximation)**.**
Denoting the weight matrix as $\sqrt{W_{\mathbf{x}}}_{i,j} = \sqrt{W_h(\mathbf{x}-\mathbf{x}_j)}\delta_{i,j}$, and the matrix of nodal values of the basis functions as $\Phi_{j,k} = \mathcal{R}(\varphi_k)_j = \varphi_k(\mathbf{x}_j)$, the LLS approximation has the following closed form:

$$< f >_{\mathbf{x}}^{LLS} (\mathbf{y}) = f \cdot \sqrt{W_{\mathbf{x}}}\left(\Phi^T\sqrt{W_{\mathbf{x}}}\right)^+ \cdot \boldsymbol{\varphi}(\mathbf{y}) \tag{D.3}$$

Where A^+ denotes the Moore-Penrose inverse of the matrix A.

Proof. See proposition B.2. □

Remark. The matrix $\Phi W_{\mathbf{x}} \Phi^T$ has n_{φ} lines and $\#\{i \in \mathcal{C} \mid \|\mathbf{x} - \mathbf{x}_i\| < h\}$ non-vanishing columns. Hence, the LLS procedure is *local* in the sense that its cost does not depend on the size of the entire point cloud, but rather on the complexity of the reconstruction space $\boldsymbol{\varphi}$ and the number of cloud points in the neighborhood of $\mathbf{x}$.

Definition D.2 (LLS shape functions)**.**
As a consequence of proposition D.1, the application $f : (\mathcal{C} \to \mathbb{R}) \to \mathbb{R}; f \mapsto < f >_{\mathbf{x}}^{LLS} (\mathbf{y})$ is linear. Hence, for every $\mathbf{x} \in \Omega$, the LLS approximation defines a local reconstruction operator. The corresponding LLS shape functions $\phi_{\mathbf{x}}^{LLS}$ are the best approximation in $\boldsymbol{\varphi}$ of the discrete Kronecker delta functions around $\mathbf{x}$ in the least-square sense. They can be expressed as:

$$\phi_{\mathbf{x}}^{LLS}(\mathbf{y}) \stackrel{\text{def}}{=} \sqrt{W_{\mathbf{x}}} \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \cdot \boldsymbol{\varphi}(\mathbf{y}) \tag{D.4}$$

Definition D.3 (Unisolvent and h-unisolvent point cloud)**.**
A point cloud $(\mathcal{C}, \mathbf{X})$ is unisolvent on $\boldsymbol{\varphi}$ if the zero function is the only function in $\boldsymbol{\varphi}$ that vanishes on all points. In other words, $(\mathcal{C}, \mathbf{X})$ is unisolvent on $\boldsymbol{\varphi}$ if the intersection of $\boldsymbol{\varphi}$ with the null-space of the reduction operator is $\{0\}$.
Similarly, a point cloud $(\mathcal{C}, \mathbf{X})$ is h-unisolvent on $\boldsymbol{\varphi}$ with $h > 0$ if $\mathbf{X} \cap B^d(\mathbf{x}, h)$ is unisolvent on $\boldsymbol{\varphi}$ for all $\mathbf{x} \in \Omega$.

Example (Unisolvence for polynomials spaces of low degree)**.** In the particular case where the approximation space is the space of constant functions, a point cloud $(\mathcal{C}, \mathbf{X})$ is unisolvent if and only if it has at least one node. Similarly, if the approximation space is the space of affine functions, a point cloud $(\mathcal{C}, \mathbf{X})$ is unisolvent if and only if its nodes do not lie in a $(d-1)$-dimensional hyperspace.

Proposition D.2 (Reproducibility of LLS)**.**
If the point cloud $(\mathcal{C}, \mathbf{X})$ is h-unisolvent on $\boldsymbol{\varphi}$, then the LLS procedure reproduces functions in $\boldsymbol{\varphi}$.

Proof. The h-unisolvence property of the point cloud implies that for all $\mathbf{x} \in \Omega$, the matrix $\Phi^T \sqrt{W_{\mathbf{x}}}$ has full line rank. Then, using the expression for the pseudo inverse of proposition B.2, we have $\left(\Phi^T \sqrt{W_{\mathbf{x}}}\right)^+ = \sqrt{W_{\mathbf{x}}} \Phi \left(\Phi^T W_{\mathbf{x}} \Phi\right)^{-1}$ and we can write the LLS approximation of an element of the approximation space as: $\forall\ \psi \in \boldsymbol{\varphi}$, $\forall\ \mathbf{y} \in \Omega$,

$$\begin{aligned}
< \psi >_{\mathbf{x}}^{LLS} (\mathbf{y}) &= \mathcal{R}(\psi) \cdot \sqrt{W_{\mathbf{x}}} \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \cdot \boldsymbol{\varphi}(\mathbf{y}) \\
&= \mathcal{R}(\psi) \cdot W_{\mathbf{x}} \Phi \left(\Phi^T W_{\mathbf{x}} \Phi \right)^{-1} \cdot \boldsymbol{\varphi}(\mathbf{y}) \\
&= \underbrace{\left(\left(\Phi^T W_{\mathbf{x}} \Phi \right)^{-1} \cdot \left(\Phi^T W_{\mathbf{x}} \mathcal{R}(\psi) \right) \right)}_{\text{By definition, this is the vector of coefficients of } \psi \text{ on the basis } (\varphi_k)_{k \in [\![1, n_{\varphi}]\!]}} \cdot \boldsymbol{\varphi}(\mathbf{y}) \\
&= \psi(\mathbf{y})
\end{aligned}$$

Hence, ψ is reproduced by the LLS procedure. □

D.2 Moving Least Squares reconstruction

The LLS methodology described in the previous section gives a locally determined non-local reconstruction. The fit is the best on the data set – as measured by a local weighted L^2 norm – in the approximation space, for instance a polynomial space. However, it is built only around a single point $\mathbf{x}$. The Moving Least Square reconstruction defined in this section remedies this limitation by stitching together LLS reconstructions. The corresponding

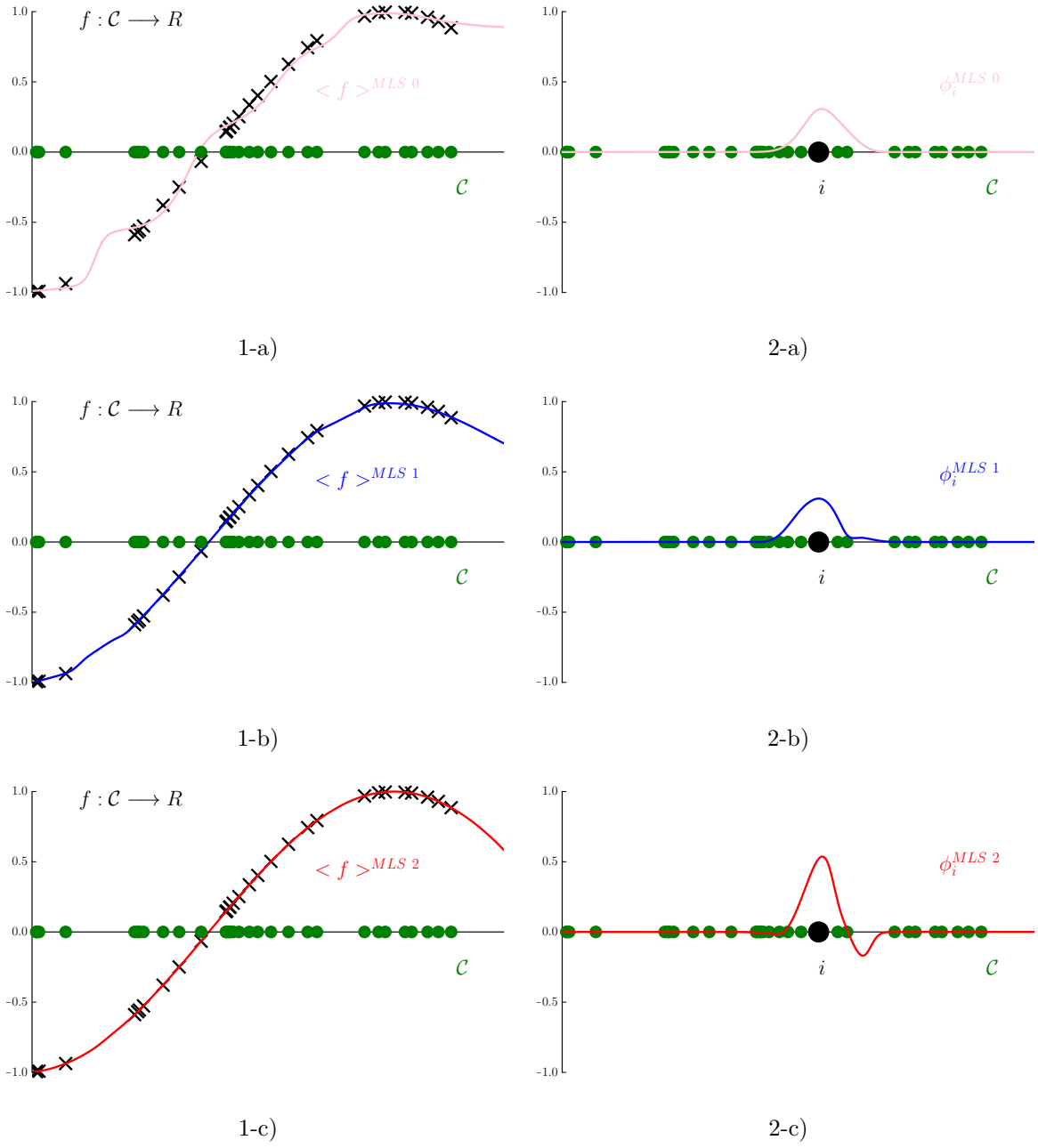


Figure D.2 – *The MLS procedure provides a way to globally approximate scattered meshless data with polynomial consistency via the definition of local shape functions.*
The point cloud $(\mathcal{C}, \mathbf{X})$*, the discrete function* f *and the weight function* W_h *are the same as those of figure D.1.*
The left column gives the MLS approximation of the data with order 0 accuracy in 1-a), order 1 accuracy in 1-b) and order 2 accuracy in 1-c).
The corresponding MLS shape functions are given for a specific node on the right column. Note that the MLS shape functions of order > 0 *are not necessarily positive, and indeed,* ϕ_i^{MLS2} *has a negative part.*

approximate does not lie in the approximation space, but is still able to reproduce functions in the approximation space. Moreover, it is local in the sense that the nodal values only influence the reconstruction of nearby nodes.

Definition D.4 (Moving Least Squares reconstruction)**.**
The *Moving Least Squares (MLS)* reconstruction at a point $\mathbf{x}$ is defined as the evaluation of the LLS reconstruction around $\mathbf{x}$ evaluated at $\mathbf{x}$: $\forall\ f : \mathcal{C} \to \mathbb{R}$,

$$< f >^{MLS} (\mathbf{x}) \stackrel{\text{def}}{=} < f >_{\mathbf{x}}^{LLS} (\mathbf{x}) \tag{D.5}$$

The corresponding shape functions read: $\forall\ i \in \mathcal{C}$, $\forall\ \mathbf{x} \in \Omega$,

$$\begin{aligned} \phi_i^{MLS}(\mathbf{x}) &\stackrel{\text{def}}{=} \phi_{\mathbf{x}}^{LLS}(\mathbf{x}) \\ &= \sqrt{W_{\mathbf{x}}} \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \cdot \boldsymbol{\varphi}(\mathbf{x}) \end{aligned} \tag{D.6}$$

In particular, the support of ϕ_i^{MLS} is included in $B^d(\mathbf{x}_i, h)$.

Figure D.2 gives several 1-D examples of MLS reconstruction.

Example (Shepard shape functions)**.** In the particular case where the approximation space is the space of constant functions, the MLS reconstruction reduces to the well-known Shepard approximation (see the original publication of Shepard [207]). The Shepard shape functions have the particularly simple following form: $\forall\ i \in \mathcal{C}$, $\forall\ \mathbf{x} \in B^d(\mathbf{x}_i, h)$,

$$\phi_i^{\text{Shepard}}(\mathbf{x}) = \frac{W_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{j \in \mathcal{C}} W_h(\mathbf{x} - \mathbf{x}_j)} \tag{D.7}$$

Under this form, it is obvious that the Shepard shape functions are positive. This feature is not true in general for other MLS shape functions. A two dimensional example of Shepard shape function is sketched on figure D.3.

Proposition D.3 (Backus-Gilbert formulation of Moving Least Squares)**.**
The MLS shape functions are themselves solution of the following optimization problem: Find $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_{n_n}(\mathbf{x}))$ such that the following expression is minimized, where $\forall\ z : \mathcal{C} \to \mathbb{R}$,

$$z \cdot W_{\mathbf{x}}^{-1} \cdot z = \sum_{i \in \mathcal{C}} \frac{1}{W(\mathbf{x} - \mathbf{x}_i)} z_i^2 \tag{D.8}$$

Under the constraint that $\phi(\mathbf{x})$ is a minimizer of the cost function defined as: $\forall\ z : \mathcal{C} \to \mathbb{R}$,

$$\sum_{k=1}^{n_\varphi} \left(\varphi_k(\mathbf{x}) - \sum_{i \in \mathcal{C}} \varphi_k(\mathbf{x}_i) z_i \right)^2 = \left\| \Phi^T z - \boldsymbol{\varphi}(\mathbf{x}) \right\|^2 \tag{D.9}$$

Remark. As the weight functions have local support, many of the nodes are such that $W(\mathbf{x} - \mathbf{x}_i)$ so that the weight matrix $W_{\mathbf{x}}^{-1}$ in equation (D.8) is ill-defined. Formally writing $\frac{1}{0} = \infty$ there, the cost function takes finite values only of the unknowns z_i corresponding to nodes i such that $W_h(\mathbf{x} - \mathbf{x}_i) = 0$ (equivalently $\|\mathbf{x} - \mathbf{x}_i\| >= h$) vanish. Hence, actually very few nodes participate to the optimization problem, which thus remains a local problem.

Proof of proposition D.3. Substituting y for $\sqrt{W_{\mathbf{x}}}^{-1} z$, the solution of the above nested optimization problem can be re-written as the minimum norm solution of the following quadratic optimization problem: Find y such that the expression $\left\| \Phi^T \sqrt{W_{\mathbf{x}}} t - \boldsymbol{\varphi}(\mathbf{x}) \right\|^2$ is minimal. As a consequence of proposition B.2, the solution of this problem reads $y = \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \cdot \boldsymbol{\varphi}(\mathbf{x})$, and thus the solution of the original problem is $\phi(\mathbf{x}) = \sqrt{W_{\mathbf{x}}} \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \cdot \boldsymbol{\varphi}(\mathbf{x})$, which is exactly expression (D.6) of the MLS shape functions. □

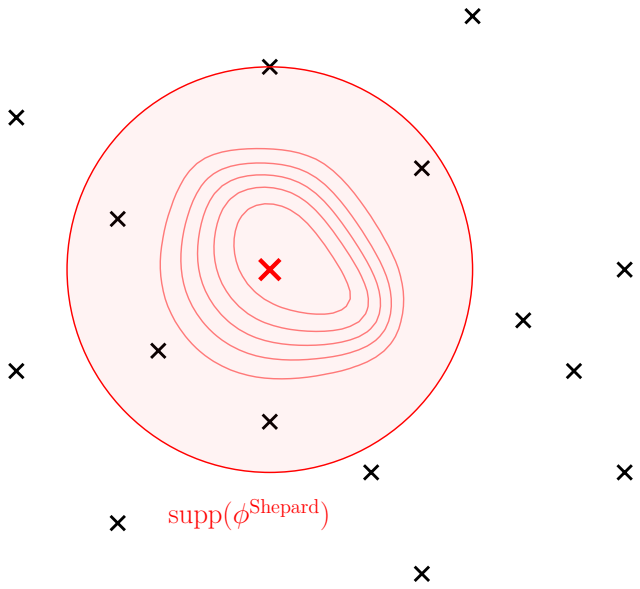


Figure D.3 – *Shepard shape functions are positive meshless shape functions that can reproduce constant fields. Like all MLS shape functions, their support is a ball of radius h centered on their associated node.*

This formulation was first given by Backus and Gilbert. Historically speaking, the Backus-Gilbert formulation is anterior to the standard MLS formulation (it was first published in 1968 [15], whereas the standard form of the MLS reconstruction in its full generality was first published in 1981 by Lancaster and Salkauskas [133]). The correspondence between the two formulations was realized by Abramovici [3] for the special case of the Shepard shape functions and by Bos and Salkauskas [40] in the general case.

Remark (Delta property and the Moving Least Square reconstruction)**.** A long standing criticism of the MLS reconstruction stems from the fact it does not satisfy the delta property in general. Nevertheless, it can be shown (see section 4 of [133]) that the delta property is verified if we use singular weight functions, *i.e.* weight functions satisfying $W_h(\mathbf{x}) \underset{\mathbf{x}\to\mathbf{0}}{\longrightarrow} +\infty$.

Proposition D.4 (Reproducibility and approximation order of MLS)**.**
If the point cloud $(\mathcal{C}, \mathbf{X})$ is h-unisolvent on $\mathcal{P}$, then the MLS shape functions reproduce functions in $\mathcal{P}$. Moreover, for multivariate polynomials, $\mathcal{P} = \mathbb{R}_p[X_1, \ldots, X_n]$ and assuming quasi-uniformity of the point cloud (see definition 2.2) and an additional regularity assumption on the computational domain Ω, the MLS approximation verifies:

$$\|f - < f >^{MLS\ p} \|_{W^0_\infty(\Omega)} \leq Ch^{p+1}|f|_{W^{p+1}_\infty} \tag{D.10}$$

The constant C in the above bound can be made explicit: see theorem 4.7 and corollary 4.8 of Wendland [233] for the exact result.

Proof. Reproducibility of the MLS shape functions immediately results from reproducibility of the LLS shape functions. Moreover, under an additional regularity condition on the computational domain and the point cloud similar to the strong cone condition of definition A.2, the approximation property stems from a discrete analogue of the Bramble-Hilbert lemma A.1 given in theorems 3.2 and 3.14 of [233]. A complete proof can be found in this book. □

D.3 Diffuse Moving Least Squares gradient

Just like the MLS reconstruction takes advantage of a different LLS approximation at each point $\mathbf{x} \in \Omega$, we now define the Diffuse Moving Least Square gradient operator, which exploits the local behavior of the LLS approximation around each node of the point cloud.

Definition D.5 (Diffuse Moving Least Squares gradient)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud and $f : \mathcal{C} \to \mathbb{R}$ be a discrete function. The *Diffuse Moving Least Squares (DMLS)* approximation of the gradient of f at a point $\mathbf{x} \in \Omega$ is the evaluation at $\mathbf{x}$ of the gradient of the LLS approximation of f around $\mathbf{x}$:

$$< \nabla f >^{DMLS}(\mathbf{x}) \stackrel{\text{def}}{=} \frac{\partial}{\partial \mathbf{y}} < f >_{\mathbf{x}}^{LLS}(\mathbf{y})\Big|_{\mathbf{y}=\mathbf{x}} \tag{D.11}$$

Evaluation of expression (D.11) at every node of the point cloud yields the following DMLS gradient operator:

$$\mathbb{\nabla}_i^{DMLS} f \stackrel{\text{def}}{=} < \nabla f >^{DMLS}(\mathbf{x}_i) \tag{D.12}$$

The DMLS discrete gradient of order inherits its exactness properties on an h-unisolvent point cloud from the reproducibility properties of the LLS shape functions (proposition D.2). In particular, it means that it is possible to build discrete gradient operators that are consistent of *any* order $p \in \mathcal{N}$.

The coordinates of the DMLS gradient read: $\forall\, i, j \in \mathcal{C}$,

$$\begin{aligned} \mathbf{A}_{i,j}^{DMLS} &= V_i \left(\nabla \phi_{\mathbf{x}_i}^{LLS}{}_j \right)(\mathbf{x}_i) \\ &= V_i \left(\sqrt{W_{\mathbf{x}_i}} \left(\Phi^T \sqrt{W_{\mathbf{x}_i}} \right)^+ \cdot \nabla \boldsymbol{\varphi}(\mathbf{x}_i) \right)_j \end{aligned} \tag{D.13}$$

In the next section, we prove that the DMLS gradient of order one can be rephrased as a volume-based primal gradient.

D.4 A new formula for first order Diffuse Moving Least Squares

In order to build a first order consistent volume primal gradient, the volume function needs to comply with the linear covariance property stated in expression (3.24). We now give an example of such a volume function, and then prove that the corresponding primal gradient coincides with the linear DMLS gradient. Given a point cloud $\mathcal{C}$ equipped with a graph structure and positive edge weights $w_{i,j} > 0$, we define:

$$V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}(\mathbf{x}_1, \dots, \mathbf{x}_n) \stackrel{\text{def}}{=} \left(\det \sum_{j \in \mathcal{N}(i)} w_{i,j} (\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T \right)^{1/2} \tag{D.14}$$

In this definition, node positions only appear through the expression $\mathbf{x}_j - \mathbf{x}_i$, which ensures invariance with respect to translation, hence zeroth order consistency of the associated gradient. The inner sum is composed of positive semi-definite matrices. Consequently, the volume weights are non-negative. Since only nodes j for which $w_{i,j}$ does not vanish appear in definition (D.14), we can say that the graph induced by this volume function is contained into the assumed initial graph defined alongside the edge weights. We can prove that under mildly restrictive conditions, the two graphs exactly coincide. In the special case where

$\mathcal{N}(i)$ is a d-dimensional simplex and weights are chosen as $w_{i,j} = \delta_{j\in\mathcal{N}(i)}$, expression (D.14) effectively computes $d!$ times the volume of this simplex. In any case, this expression has long been recognized in the literature as a possible definition of the volume of the position matrix (see [29] for more information).

For $\mathrm{B} \in \mathrm{M}_n(\mathbb{R})$, let us compute $V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}(\mathrm{B}\mathbf{x}_1, \dots, \mathrm{B}\mathbf{x}_n)$:

$$\begin{aligned} V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}(\mathrm{B}\mathbf{x}_1, \dots, \mathrm{B}\mathbf{x}_n) &= \left(\det \sum_{j\in\mathcal{N}(i)} w_{i,j} (\mathrm{B}\mathbf{x}_j - \mathrm{B}\mathbf{x}_i)(\mathrm{B}\mathbf{x}_j - \mathrm{B}\mathbf{x}_i)^T \right)^{1/2} \\ &= \left(\det \mathrm{B} \sum_{j\in\mathcal{N}(i)} w_{i,j} (\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T \mathrm{B}^T \right)^{1/2} \\ &= \sqrt{\det(\mathrm{B}\mathrm{B}^T)} V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}\ (\mathbf{x}_1, \dots, \mathbf{x}_n) \\ &= |\det(\mathrm{B})|\ V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}(\mathbf{x}_1, \dots, \mathbf{x}_n) \end{aligned}$$

Hence, from (3.24), we conclude that the $\mathbb{V}^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}$ is first order consistent.

Let us denote $\mathrm{M}_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}} = \sum_{j\in\mathcal{N}(i)} w_{i,j}(\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T$. Using the Jacobi formula for the derivative of a determinant, we can write for $i \neq j$,

$$\begin{aligned} \frac{\partial V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}}{\partial \mathbf{x}_j} &= \frac{1}{2V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}} \frac{\partial}{\partial \mathbf{x}_j} \det\left(\mathrm{M}_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}\right) \\ &= \frac{1}{2} V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}} \operatorname{Tr}\left((\mathrm{M}_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}})^{-1} \frac{\partial \mathrm{M}_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}}{\partial \mathbf{x}_j} \right) \\ &= V_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}} w_{i,j} \left(\mathrm{M}_i^{DMLS\{\mathbf{x}-\mathbf{x}_i\}}\right)^{-1} \cdot (\mathbf{x}_j - \mathbf{x}_i) \end{aligned}$$

This formula is exactly that of the so-called "diffuse" moving least squares approximation around node i (see expression D.13) using a linear basis and singular weights. A similar computation shows that the following volume function yields the standard diffuse MLS gradient with respect to the affine basis $\{1, \mathbf{x}\}$ and non-singular weights:

$$V_i^{DMLS\{1,\mathbf{x}\}}(\mathbf{x}_1, \dots, \mathbf{x}_n) = \left(\det \sum_{j\in\mathcal{N}(i)} w_{i,j} \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix}^T \right)^{1/2} \tag{D.15}$$

This expression of diffuse moving least squares approximation constitutes an original result. Formulae (D.14) and (D.15) were designed using the global result (3.24) as a basis, and their link with MLS approximations was unexpected. Let us remark that these formulae are of little practical value since edge weights are not allowed to vary and, in most situations, there is no known guideline to choose appropriate values. In particular, this means that the graph is static independently of nodes positions, which violates the locality principle. Hence, this does not contradict proposition 3.9 concerning linear primal consistency of volume-based gradient operators.

Remark. Using the fact that W_h is radial, we can prove that the renormalized SPH gradient operator of order one can be re-interpreted as a kind of linear consistent DMLS gradient with singular weights. Indeed, denoting W' the derivative of W with respect to its scalar argument, we can write the renormalized SPH gradient of order zero of a linear field as:

$$V_i \mathbb{V}_i^{SPH\ R0}\mathbf{x} = -V_i \sum_{j\in\mathcal{C}} \left(V_j \frac{1}{c_d(W) h \|\mathbf{x}_j - \mathbf{x}_i\|} W'(\|\mathbf{x}_j - \mathbf{x}_i\|) \right) (\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^T \tag{D.16}$$

Hence, denoting $w_{i,j} = V_j \frac{1}{c_d(W)h\|\mathbf{x}_j - \mathbf{x}_i\|} W'(\|\mathbf{x}_j - \mathbf{x}_i\|)$ for $i \neq j$, the expression of $\mathbb{\nabla}^{SPH\ R1}$ coincides with that of the DMLS gradient using a linear basis.

Proposition D.5 (Optimality of Diffuse Moving Least Squares gradient coordinates)**.**
The vector coefficients of the discrete DMLS gradient $\mathbb{\nabla}^{DMLS}$ constitute the minimum norm solution to the minimization of the following positive quadratic function: Find $\left(\mathbf{A}_{i,j}^{DMLS}\right)_{i,j\in\mathcal{C}}$ such that the following expression is minimized:

$$\sum_{i,j\in\mathcal{C}} \frac{1}{V_i W_h(\mathbf{x}_j - \mathbf{x}_i)} \mathbf{A}_{i,j}^2 \tag{D.17}$$

Under the constraint that $\mathbf{A}^{DLMS}$ is also a minimizer of the following cost expression:

$$\sum_{i\in\mathcal{C}} \sum_{k=1}^{n_\varphi} \left(\nabla\varphi_k(\mathbf{x}_i) - \frac{1}{V_i} \sum_{j\in\mathcal{C}} \mathbf{A}_{i,j}\varphi_k(\mathbf{x}_j) \right)^2 = \sum_{i\in\mathcal{C}} \left\| \frac{1}{V_i} \Phi^T \mathbf{A} - \nabla\boldsymbol{\varphi}(\mathbf{x}_i) \right\|^2 \tag{D.18}$$

Proof. The proof is essentially identical to that of proposition D.3. □

Remark. Even though proposition D.5 and especially expression (D.18) suggest that computing the DMLS gradient requires solving a *global* system of linear equations, expression (D.13) shows that the actual cost is that of n_n *local* problems.

D.5 Full Moving Least Squares gradient

Just like the DMLS gradient operator is defined as the derivative of the LLS reconstruction at nodes of the point cloud, we define the Full Moving Least Squares gradient as the derivative of the MLS approximation at each node:

Definition D.6 (Full Moving Least Squares gradient)**.**
Let $(\mathcal{C}, \mathbf{X})$ be a point cloud and $f : \mathcal{C} \to \mathbb{R}$ be a discrete function. Suppose that the MLS approximation of f is differentiable, then, the *Full Moving Least Squares (FMLS)* approximation of the gradient of f at a point $\mathbf{x} \in \Omega$ is defined as the evaluation at $\mathbf{x}$ of the gradient of the MLS approximation of f:

$$< \nabla f >^{FMLS} (\mathbf{x}) \overset{\text{def}}{=} \nabla < f >^{MLS} (\mathbf{x}) \tag{D.19}$$

Evaluation of expression (D.19) at every node of the point cloud yields the following FMLS gradient operator:

$$\mathbb{\nabla}_i^{FMLS} f \overset{\text{def}}{=} < \nabla f >^{FMLS} (\mathbf{x}_i) \tag{D.20}$$

The FMLS discrete gradient of order inherits its exactness properties on an h-unisolvent point cloud from the reproducibility properties of the MLS shape functions (proposition D.4).

Proposition D.6 (Differentiability of Moving Least Squares shape functions)**.**
Suppose that the weight function W_h and the approximation basis $(\varphi_k)_{k\in[\![1,n_\varphi]\!]}$ are differentiable. If the point cloud $(\mathcal{C}, \mathbf{X})$ is h-unisolvent on $\boldsymbol{\varphi}$, then the MLS reconstruction is differentiable. Moreover, in this case, the coordinates of the FMLS gradient read: $\forall\ i, j \in \mathcal{C}$,

$$\mathbf{A}_{i,j}^{FMLS} = V_i \left(\nabla\phi_j^{MLS} \right) (\mathbf{x}_i)$$

$$
\begin{aligned}
&= V_i \frac{\partial}{\partial \mathbf{x}} \sqrt{W_{\mathbf{x}}} \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \cdot \boldsymbol{\varphi}(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_i} \\
&= \underbrace{V_i \sqrt{W_{\mathbf{x}_i}} \left(\Phi^T \sqrt{W_{\mathbf{x}_i}} \right)^+ \cdot \nabla \boldsymbol{\varphi}(\mathbf{x}_i)}_{=\mathbf{A}_{i,j}^{DMLS}} + V_i \frac{\partial}{\partial \mathbf{x}} \left(\sqrt{W_{\mathbf{x}}} \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \right) \cdot \boldsymbol{\varphi}(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_i}
\end{aligned} \tag{D.21}
$$

In particular, this expression shows that $\mathbb{\nabla}^{FMLS}$ is the sum of the consistent term $\mathbb{\nabla}^{DMLS}$ and another term that does not affect consistency. Proposition B.3 can be used to derive the full expression of $\frac{\partial}{\partial \mathbf{x}} \left(\sqrt{W_{\mathbf{x}}} \Phi^T \sqrt{W_{\mathbf{x}}} \left(\Phi^T \sqrt{W_{\mathbf{x}}} \right)^+ \right)$ necessary for a practical implementation. However, we do not consider that this intricate expression is informative enough to be written here.

Proof. This is a direct consequence of the full rank of $\Phi^T \sqrt{W_{\mathbf{x}}}$ and proposition B.3. □

D.6 Comparison of DMLS and FMLS gradients

The related LLS and MLS procedures yield two different discrete gradients $\mathbb{\nabla}^{DMLS}$ and $\mathbb{\nabla}^{FMLS}$, which we now qualitatively compare.

On the one hand, proposition D.5 ensures that $\mathbb{\nabla}^{DMLS}$ is the *best* approximation of the gradient in the least-square sense. This feature was remarked by Mirzaei et al. in theorem 3.1 of [159], and is a strong argument in favor of DMLS gradients. The adjective "diffuse" might be a reason of the comparative unpopularity of DMLS gradients, but as Mirzaei et al. indicate, "there is nothing *diffuse* or *uncertain* about them". This term was coined in 1992 in [174] where they were used in a Galerkin-like formulation to form the first instance of Diffuse Element Method (DEM).

Supporters of $\mathbb{\nabla}^{FMLS}$ argue that it is the point-wise evaluation of a globally defined function. This property is especially useful when the MLS shape functions are used in a Galerkin formulation (see section 4.2.2), or for post-processing purposes. However, detractors of $\mathbb{\nabla}^{FMLS}$ point out the fact that its consistency in merely a consequence of the consistency of $\mathbb{\nabla}^{DMLS}$, and that the addition of the last term of expression (D.21) is a useless computational burden.

The fact that both DMLS and FMLS gradients lead to consistent approximation of any order sets them on equal theoretical footing. Moreover, none of these gradients are part of known sets of compatible operators, and are thus – as is – not completely satisfying choices for the simulation of PDEs.

Building compatible operators around the DMLS and FMLS concepts lead to widely different strategies, which correspond to the two cases of the meshless curse (see conjecture 2.4): In section 2.7.2 and 3.7.1 on the one hand, we propose two variation of the Smoothed Finite Element Method (SFEM) that exploit the fact that the MLS reconstruction is globally valid to form a discrete integration operator. The corresponding discrete gradient operators use the derivative of the MLS shape functions, and are thus in essence very close to the FMLS operator. In these methods, the details of the integration process are capital to ensure compatibility, and we propose mesh-based integration techniques that allow the practical computation of compatible operators.

On the other hand, we define in the section 2.6 a correction procedure to retrieve compatibility of a set of meshless operators with first order consistency. This procedure is particularly well suited to the DMLS gradient as illustrated in proposition 2.5, which states that this correction transfers the local optimality of DMLS to a global optimality.

Appendix E

Volume-based reproducing kernel operators

The theory of Reproducing Kernel Hilbert Space (RKHS) gives a framework for the study of functions spaces in which point evaluation (our favorite choice of reduction map) is a continuous functional. In particular, it develops a powerful approximation theory and provides several optimality results for the reconstruction of functions given scattered nodal data (see [84, 201, 233] and references therein for more information). This theory seems particularly well suited for the development of meshless methods, and has been successfully used in a meshless simulation context, with the work of Fasshauer [91] and Wendland [231] (also see [186, 238]).

Using the methodology described in section 3.7.1, we can exploit the RKHS reconstruction to build a volume function, transferring reproducibility of the shape functions into consistency of the discrete integration and dual gradient. The idea of designing a meshless nodal integration procedure using the RKHS principle was first studied by Sommariva and Vianello in [211, 212]. In particular, these authors perform an interesting error analysis of the resulting volume weights.

In this section, we first recall the definition of the RKHS reconstruction operator. Then, we quickly review the consistency properties of the resulting volume-based SFEM operators and prove an amusing result concerning the corresponding Sibson reconstruction. Finally, we investigate the formal similarities between this approach and the SPH methodology in section E.1.

Following definition 6.2 of [201], we introduce the main notion for RKHS reconstructions:

Definition E.1 (Conditionally positive (semi-)definite reproducing kernel)**.**
A *conditionally positive (semi-)definite reproducing kernel of order m* is a symmetric function:

$$\left\{ \begin{array}{c} K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \\ (\mathbf{x}, \mathbf{y}) \mapsto K(\mathbf{x}, \mathbf{y}) = K(\mathbf{y}, \mathbf{x}) \end{array} \right. \tag{E.1}$$

Such that for all finite point cloud $(\mathcal{C}, \mathbf{X})$ with *distinct* node positions, the symmetric matrix $\mathrm{K}_{\mathcal{C}} \in \mathrm{M}_{n_n}(\mathbb{R})$ with the following coefficients:

$$\mathrm{K}_{\mathcal{C}} \overset{\text{def}}{=} \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \dots & K(\mathbf{x}_1, \mathbf{x}_{n_n}) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_{n_n}, \mathbf{x}_1) & \dots & K(\mathbf{x}_{n_n}, \mathbf{x}_{n_n}) \end{pmatrix} \tag{E.2}$$

defines a symmetric positive (semi-)definite quadratic form on the subspace:

$$V_{m,\mathbf{X}} \overset{\text{def}}{=} \{\alpha : \mathcal{C} \to \mathbb{R} \mid \sum_{i \in \mathcal{C}} \alpha_i p(\mathbf{x}_i) = 0 \ \forall \ p \in P_{m-1}\} \tag{E.3}$$

In the above expression, P_{m-1} denotes the space of d-variate polynomials of maximum degree $m-1$.

Example. The following compactly supported "Wendland" kernel (see [230]) is C^2-continuous and conditionally positive definite of order $m = 0$ for space dimensions $d \leq 3$: $\forall \ \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$K(\mathbf{x}, \mathbf{y}) = \begin{cases} (1 - \|\mathbf{y} - \mathbf{x}\|)^4 (4\|\mathbf{y} - \mathbf{x}\| + 1) & \text{if } \|\mathbf{y} - \mathbf{x}\| \leq 1 \\ 0 & \text{else} \end{cases} \tag{E.4}$$

It is proved in [233] to be part of a larger family of compactly supported kernels with arbitrarily high differentiability order in arbitrarily high space dimensions.

Definition E.2 (Reproducing Kernel Hilbert Space reconstruction)**.**
The dimension of P_{m-1} is $n_p = \binom{m+d-1}{d}$. We denote $(p_1, \ldots, p_{n_p})$ one of its basis, and P the matrix of nodal evaluations of basis functions $\mathrm{P}_{j,q} = \mathcal{R}(p_q)_j = p_q(\mathbf{x}_j)$. If $(\mathcal{C}, \mathbf{X})$ is an unisolvent point cloud on P_{m-1} (see definition D.3), and K is a conditionally positive definite reproducing kernel, then the Reproducing Kernel Hilbert Space (RKHS) approximation of a discrete function $f : \mathcal{C} \to \mathbb{R}$ reads: $\forall \ \mathbf{x} \in \mathbb{R}^d$,

$$< f >^{RKHS} (\mathbf{x}) \overset{\text{def}}{=} \sum_{i \in \mathcal{C}} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{q=1}^{n_p} \beta_q p_q(\mathbf{x}) \tag{E.5}$$

In the above expression, $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ is the unique solution to the following system of linear equations:

$$\begin{pmatrix} \mathrm{K}_{\mathcal{C}} & \mathrm{P} \\ \mathrm{P}^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \overset{\text{def}}{=} \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{E.6}$$

The first n_n lines of the above system ensure that the reconstruction given in expression (E.5) satisfies the delta property (see definition 2.11). The last n_p lines ensure that the solution satisfies the vanishing moment conditions characteristic of $V_{m,\mathbf{X}}$ defined by expression E.3. Defining $k_i(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}_i)$, the RKHS reconstruction finally reads:

$$< f >^{RKHS} (\mathbf{x}) = \begin{pmatrix} k(\mathbf{x}) \\ p(\mathbf{x}) \end{pmatrix} \cdot \begin{pmatrix} \mathrm{K}_{\mathcal{C}} & \mathrm{P} \\ \mathrm{P}^T & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{E.7}$$

So that the RKHS basis functions are linear combinations of polynomials and kernel functions. They read:

$$\phi^{RKHS}(\mathbf{x}) = \left(\begin{pmatrix} \mathrm{K}_{\mathcal{C}} & \mathrm{P} \\ \mathrm{P}^T & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} k(\mathbf{x}) \\ p(\mathbf{x}) \end{pmatrix} \right)_{1:n_n} \tag{E.8}$$

The subscript $1 : n_n$ means that only the first n_n coefficients of the vector are retained. In particular, these functions do not have compact support in general, even if the reproducing kernel K is local.

Proposition E.1 (Consistency of RKHS reconstruction)**.**
Let K be a conditionally positive definite reproducing kernel of order m and $(\mathcal{C}, \mathbf{X})$ a P_{m-1}-unisolvent point cloud (see definition D.3). Then the RKHS reconstruction reproduces polynomials of degree up to $m-1$.

Proof. See theorem 8.21 of [233]. □

Definition E.3 (Reproducing Kernel Hilbert Space discrete integration operator)**.**
The RKHS volume discrete integration operator can be defined using the volume SFEM methodology of section 3.7.1:

$$\oint_{\mathcal{C}} f \overset{\text{def}}{=} \int_{\mathbf{x}\in\Omega} < f >^{RKHS} (\mathbf{x})\ \mathrm{d}V \tag{E.9}$$

Denoting $\begin{pmatrix} I_{\mathcal{C}} \\ J \end{pmatrix} \overset{\text{def}}{=} \int_{\mathbf{x}\in\Omega} \begin{pmatrix} k \\ p \end{pmatrix} (\mathbf{x})\ \mathrm{d}V$ the values of the integrals of the partial evaluations of the kernel at node positions and of the basis polynomials, we can write the RKHS volumes as:

$$\begin{pmatrix} V^{RKHS} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathrm{K}_{\mathcal{C}} & \mathrm{P} \\ \mathrm{P}^T & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} I_{\mathcal{C}} \\ J \end{pmatrix} \tag{E.10}$$

As pointed out earlier, the RKHS shape functions depend on all nodes of the point cloud in a non-local way, hence, the RKHS volume function is non-local.

Proposition E.2 (Consistency of RKHS volume-based operators)**.**
The RKHS integration operator inherits its exactness on polynomials of maximum total degree $m-1$ from the reproducibility properties of the RKHS reconstruction. In accordance with proposition 3.13, this in turn implies consistency of order $m-1$ of the dual volume-based gradient.

Consistency of the primal volume based gradient is a consequence of the covariance property of the volume function. Invariance with respect to isometric transformations is achieved if the kernel is a Radial Basis Function (RBF), namely if: $\forall\ \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$,

$$K(\mathbf{x}, \mathbf{y}) = W(\|\mathbf{y} - \mathbf{x}\|) \tag{E.11}$$

This is for instance the case of the Wendland kernel defined earlier.

Proposition E.3 (Sibson reconstruction of RKHS volumes is the RKHS reconstruction)**.**
The Sibson procedure of definition 3.8 applied to the RKHS volume function yields the RKHS reconstruction.

Proof. Let us denote $(\mathcal{C}', (\mathbf{x}, \mathbf{X}))$ the point cloud $(\mathcal{C}, \mathbf{X})$ augmented by the addition of a n_n+1th node. For convenience, the index of this node is $i = 0$, and we denote $\mathrm{D} = \begin{pmatrix} \mathrm{K}_{\mathcal{C}} & \mathrm{P} \\ \mathrm{P}^T & 0 \end{pmatrix}$ and $\psi(\mathbf{x}) = \begin{pmatrix} k(\mathbf{x}) \\ p(\mathbf{x}) \end{pmatrix}$. Hence, denoting $I_0 = \int_{\mathbf{y}\in\Omega} K(\mathbf{x}, \mathbf{y})\ \mathrm{d}V$, the RKHS volumes on $\mathcal{C}'$ read:

$$\begin{pmatrix} V^{RKHS}_{\mathcal{C}'} \\ 0 \end{pmatrix} = \begin{pmatrix} K(\mathbf{x}, \mathbf{x}) & \psi(\mathbf{x})^T \\ \psi(\mathbf{x}) & \mathrm{D} \end{pmatrix}^{-1} \cdot \begin{pmatrix} I_0 \\ I_{\mathcal{C}} \\ J \end{pmatrix} \tag{E.12}$$

Since the sub-matrix D is invertible, we can re-write the above matrix inverse using the Schur complement.

$$\begin{pmatrix} K(\mathbf{x}, \mathbf{x}) & \psi(\mathbf{x})^T \\ \psi(\mathbf{x}) & \mathrm{D} \end{pmatrix}^{-1} = \begin{pmatrix} 0 & 0 \\ 0 & \mathrm{D}^{-1} \end{pmatrix} + \frac{1}{K(\mathbf{x}, \mathbf{x}) - \psi(\mathbf{x})^T \mathrm{D}^{-1} \psi(\mathbf{x})} \begin{pmatrix} 1 & -\psi(\mathbf{x})^T \mathrm{D}^{-1} \\ -\mathrm{D}^{-1}\psi & \mathrm{D}^{-1}\psi(\mathbf{x})\psi(\mathbf{x})^T \mathrm{D}^{-1} \end{pmatrix}$$

And thus, the RKHS volumes on $\mathcal{C}$ can be expressed in terms of the RKHS volumes on $\mathcal{C}$:

$$\begin{cases} V^{RKHS}_{\mathcal{C}'\ 0} = \dfrac{I_0 - k(\mathbf{x})^T V^{RKHS}_{\mathcal{C}}}{K(\mathbf{x},\mathbf{x}) - \psi(\mathbf{x})^T \mathrm{D}^{-1}\psi(\mathbf{x})} \\ V^{RKHS}_{\mathcal{C}'\ 1:n_n} = V^{RKHS}_{\mathcal{C}} - \dfrac{I_0 - k(\mathbf{x})^T V^{RKHS}_{\mathcal{C}}}{K(\mathbf{x},\mathbf{x}) - \psi(\mathbf{x})^T \mathrm{D}^{-1}\psi(\mathbf{x})} \left(\mathrm{D}^{-1}\psi(\mathbf{x})\right)_{1:n_n} \end{cases}$$

Hence, the corresponding Sibson shape functions on $\mathcal{C}$ evaluated at $\mathbf{x}$ read:

$$\begin{aligned} \phi^{RKHS\ S}(\mathbf{x}) &= \frac{V^{RKHS}_{\mathcal{C}} - V^{RKHS}_{\mathcal{C}'\ 1:n_n}}{V^{RKHS}_{\mathcal{C}'\ 0}} \\ &= \left(\mathrm{D}^{-1}\psi(\mathbf{x})\right)_{1:n_n} \\ &= \left(\begin{pmatrix} \mathrm{K}_{\mathcal{C}} & \mathrm{P} \\ \mathrm{P}^T & 0 \end{pmatrix}^{-1} \cdot \begin{pmatrix} k(\mathbf{x}) \\ p(\mathbf{x}) \end{pmatrix} \right)_{1-n_n} \\ &= \phi^{RKHS}(\mathbf{x}) \end{aligned}$$

Which is the desired result. □

E.1 Similarities between RKHS volumes and SPH

Let us have a more in-depth look at the particular case $m = 0$, namely the case where no reproducibility property is sought. Suppose $\Omega = \mathbb{R}^d$ in order to avoid technicalities near the boundary. Using a compactly supported RBFs denoted W_h, scaled so that its integral over space is unity, we can re-write expression (E.10) of the RKHS volumes weights as: $\forall\ i \in \mathcal{C}$,

$$\sum_{j\in\mathcal{C}} V_j^{RKHS} W_h(\mathbf{x}_j - \mathbf{x}_i) = 1 \tag{E.13}$$

Formally, this is exactly equivalent to choosing volume weights in such a way that the SPH reconstruction (see expression (2.41)) of a unit field is unity when evaluated at node positions: $\forall\ i \in \mathcal{C}, < 1 >^{SPH}(\mathbf{x}_i) = 1$. This definition for nodal volumes is very similar to the standard definition of SPH volumes, which in the case of equal mass weights reads as: $\sum_{j\in\mathcal{C}} V_i^{SPH} W_h(\mathbf{x}_j - \mathbf{x}_i) = 1$. Of course, this subtle change makes the RKHS volumes fully implicit. The connection with SPH is strengthened with the following result:

Proposition E.4 (A SPH-like convergent splitting for equation (E.13))**.**
Let W_h be a conditionally positive definite of order $m = 0$, that is also a SPH kernel (in the sense of definition 2.19). Denoting $\mathrm{W}_{i,j} = W_h(\mathbf{x}_j - \mathbf{x}_i)$ and $\mathrm{D}_{i,j} = \delta_{i,j} \sum_{k\in\mathcal{C}} \mathrm{W}_{i,k}$, the linear system (E.13) can be re-written as:

$$V^{RKHS} = \mathrm{D}^{-1}1 + (\mathrm{I} - \mathrm{D}^{-1}\mathrm{W})V^{RKHS} \tag{E.14}$$

The above expression suggests the following iterative algorithm to solve system (E.13): $\forall\ i \in \mathcal{C}$, $\forall\ k \in \mathbb{N}$,

$$\begin{cases} V_i^{(0)} = V_0 \\ V_i^{(k+1)} = \dfrac{1 - \sum_{j\in\mathcal{N}(i)} (V_j^{(k)} - V_i^{(k)}) W_h(\mathbf{x}_j - \mathbf{x}_i)}{\sum_{j\in\mathcal{N}(i)} W_h(\mathbf{x}_j - \mathbf{x}_i)} \end{cases} \tag{E.15}$$

The above splitting is convergent if the graph of W is strongly connected (*i.e.* if for two nodes i, j of the point cloud there is a path $(k_1, \ldots, k_p)$ of nodes from i to j verifying

$\|\mathbf{x}_{k_{q+1}} - \mathbf{x}_{k_q}\| < h$). Its first iteration ($k = 1$) exactly gives the classical SPH volumes (see expression (3.66)).

Proof. Every coefficient of the matrix W is non-negative since we assume W_h to be a SPH kernel. The matrix $\mathrm{D}^{-1}\mathrm{W}$ is non-negative and satisfies $\mathrm{D}^{-1}\mathrm{W}1 = 1$. It is irreducible thanks to the connectivity assumption on the point cloud. The Perron-Frobenius theorem (see [196] for an introduction to Perron-Frobenius theory) thus applies and shows that its spectrum is included in $]0, 1]$. Consequently, the spectrum $\mathrm{I} - \mathrm{D}^{-1}\mathrm{W}$ is a subset of $[0, 1[$. This means that $\mathrm{I} - \mathrm{D}^{-1}\mathrm{W}$ is a convergent matrix and the iterative algorithm converges. □

Remark. The above proposition provides a link between the RKHS and the SPH volume functions. It is thus natural to seek similarities between the two methods. Differentiation of expression (E.13) with respect to node position gives an implicit relationship for the gradient coefficients, which can be re-arranged into the following expression for the nodal evaluation of the reconstruction of the volume-based RKHS primal gradient: $\forall\, i \in \mathcal{C}$,

$$< \mathbb{\nabla}^{RKHS} f > (\mathbf{x}_i) \;\; = - \sum_{j \in \mathcal{N}(i)} V_j^{RKHS} \nabla W_h(\mathbf{x}_j - \mathbf{x}_i)(f_j - f_i) \tag{E.16}$$

Similarly, we can prove the following implicit relationship involving the corresponding dual gradient: $\forall\, i \in \mathcal{C}$,

$$V_i \mathbb{\nabla}_i^{RKHS\,*}(\mathrm{W}f) = - \sum_{j \in \mathcal{N}(i)} (V_j^{RKHS} f_i + V_i^{RKHS} f_j) \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \tag{E.17}$$

In particular, replacing f with V^{RKHS} in the above expression yields the expression of the dual gradient of a unit field:

$$V_i \mathbb{\nabla}_i^{RKHS\,*} 1 = -2 \sum_{j \in \mathcal{N}(i)} V_i^{RKHS} V_j^{RKHS} \nabla W_h(\mathbf{x}_j - \mathbf{x}_i) \tag{E.18}$$

This expression is formally identical to that of the classical dual renormalized SPH gradient of order zero (see expression (2.50)).

Appendix F

Voronoi and power diagram-based volume function

One of the most successful ideas in the meshless community has arguably been to define the volume of a node as the volume of its associated Voronoi cell. To the author's knowledge, the first apparition of this idea comes from Mikhailova, Shashkov and coworkers (see [158] for instance, where first order dual consistency is not even pointed out). This construction has been used successfully several times since, and we mention the important contributions of Serrano and Español in [206] and the weighted version of de Goes *et al.* in [67] called power diagrams. Unfortunately, none of the previously stated contributions have provided an operator based treatment of the boundary, thus requiring *ad-hoc* tricks for the imposition of non-homogeneous boundary conditions, the most popular being the introduction of "ghost" nodes as in [193] with imposed values.

The Voronoi cell Vor_i corresponding to node i in the point cloud $\mathcal{C}$ is defined as:

$$\mathrm{Vor}_i \overset{\text{def}}{=} \{\mathbf{x} \in \Omega \mid \forall\, j \in \mathcal{C}, \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\|\} \tag{F.1}$$

The corresponding volume weight is the volume of the Voronoi cell:

$$V_i^{VOR} = \int_{\mathrm{Vor}_i} 1 \, \mathrm{d}V = \int_{\Omega} 1_{|\mathrm{Vor}_i} \, \mathrm{d}V \tag{F.2}$$

Computing the corresponding partial derivatives require the computation of mesh geometric quantities. Indeed, denoting $F_{i,j}$ the scalar area of the common mesh face between cells Vor_i and Vor_j, and $\mathbf{b}_{i,j}$ its centroid, it can be shown (see [67]) that:

$$\frac{\partial V_i^{VOR}}{\partial \mathbf{x}_j} = \frac{F_{i,j}}{\|\mathbf{x}_i - \mathbf{x}_j\|}(\mathbf{x}_j - \mathbf{b}_{i,j}) \tag{F.3}$$

As illustrated on figure F.1 , this means that two nodes are neighbors if their corresponding Voronoi cells share a face. The previously stated characterization of primal and dual consistency immediately gives the following results:

- Translation covariance of the Voronoi volumes translate to zeroth order consistency of the associated primal gradient (see equation 3.23). Moreover, covariance with respect to rotations readily entails the symmetry of $\nabla^{VOR}\mathbf{x}$.
- A tiling (meshing!) of space is effectively built. Thus, node volumes always add up to the total volume of the domain. As a consequence of proposition 3.13, this means that the dual gradient is zeroth order consistent.

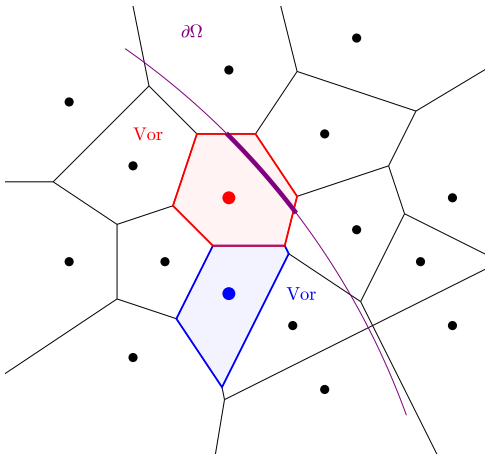


Figure F.1 – *In the case of Voronoi volumes, the value of a nodal volume weight locally depends on the position of another node if the corresponding Voronoi cells share a face (whose intersection with* Ω *has non-vanishing* $d-1$*-dimensional measure). Hence,* $\mathbf{A}^{VOR\,*}_{red,blue} \neq \mathbf{0}$*. Furthermore, the volume of a node locally depends on* Ω *if the corresponding Voronoi cell intersects the boundary* $\partial\Omega$*. In particular, the red cell is a left-sided boundary node while the blue node is not.*

- Amazingly enough, the dual gradient is moreover first order consistent. This highly non-trivial fact can be proved using the early result detailed in [208] in conjunction with proposition 3.19 of the present article. This superconsistency property is a long known fact and the main reason for the success of the Voronoi volumes.

In a nutshell, Voronoi volumes are first order dual consistent and zeroth order primal consistent, which make them very attractive options for simulation. Finally, let us point out that this method does not break the meshless curse (see conjecture 2.4) since a mesh still needs to be built.

Remark. Originally, the Sibson reconstruction (see [208]) was developed for Voronoi diagrams. Recently, the Sibson shape functions have been shown by by Hiyoshi, Sugihara and coworkers to be a part of a more general family of linear consistent shape functions. The corresponding construction (see [118] for instance) is extremely reminiscent of our own Sibson-SFEM cycle, but we have not managed to either prove or disprove that the two methods coincide.

Appendix G

Volume-based meshless discretization of hydrodynamics

The equations of hydrodynamics or gas dynamics are arguably the most popular set of equations addressed by meshless methods (see for instance [87, 158, 162, 206]), even giving its name to SPH. This appendix quickly states a few results concerning the discretization of hydrodynamics using the volume-based operators described in chapter 3. The discretization process is always more or less the same and is comprised of three distinct phases, which we quickly sum up.

Firstly, we need to define a frame in which the kinematics are described. Eulerian, Lagrangian or any in-between (ALE) is allowed, but this section focuses on Lagrangian descriptions, the simplest and most frequent choice for meshless methods. In practice, since the reduction map is assumed to match nodal evaluations, this means that the physical velocity is used for the advection of nodes. In other words, the kinematics of discretization nodes is governed by $\mathbf{v}_i = \frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}$, and no other velocity field is defined.

The second step is to postulate conservation of mass in any of the following *equivalent* discrete forms relating the scalar density field $\rho : \mathcal{C} \to \mathbb{R}$ and the velocity field. We insist on the fact that all of those expressions are formally identical to their continuous equivalent continuity equation.

- Existence of nodal mass weights m_i independent of time such that $m_i = \rho_i V_i$. This particular form of the conservation of mass stresses the fact that ρ is the Radon-Nikodym derivative of the (discrete) mass measure with respect to the volume measure.
- Global conservation for every "control" set of particles: $\forall\, \mathcal{U} \subset \mathcal{C}$,

$$\frac{\mathrm{d}}{\mathrm{d}t} \cancel{\int}_{\mathcal{U}} \rho \overset{\text{def}}{=} \frac{\mathrm{d}}{\mathrm{d}t} \sum_{i \in \mathcal{U}} V_i(\mathbf{x}_1, \ldots, \mathbf{x}_n)\rho_i = 0 \tag{G.1}$$

- Reynolds transport theorem for specific properties: $\forall\, f : \mathbb{R} \times \mathcal{C} \to \mathbb{R}$,

$$\frac{\mathrm{d}}{\mathrm{d}t} \cancel{\int}_{\mathcal{C}} \rho f = \cancel{\int}_{\mathcal{C}} \rho \frac{\mathrm{d}f}{\mathrm{d}t} \tag{G.2}$$

- Local continuity equation: $\forall\, i \in \mathcal{C}$,

$$\frac{\mathrm{d}\rho_i}{\mathrm{d}t} + \rho_i \nabla_i \cdot \mathbf{v} = 0 \tag{G.3}$$

Equivalence between all of those expressions is readily proved using expression (3.7). Let us note that since mass conservation is intimately linked with the geometrical transformation – the kinematics – underwent by the physical body, the relevant differentiation operator is naturally the primal gradient operator as highlighted on figure 3.1.

The next step is to write the discrete form of the conservation of momentum:

$$\rho_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\mathbb{\nabla}_i^* \cdot \sigma + \rho_i \mathbf{f}_i^{\text{ext}} \tag{G.4}$$

This equation (or similar ones in nature) has been extensively used for decades in the meshless community (this is the dominant practice in SPH simulations for example, see [161] for instance). The most important detail in equation (G.4) is that we define the interior forces as the dual divergence of the stress. Choosing two different differential operators (primal and dual) in the same system of equations should seem surprising, in particular since the SPH dual gradient $\mathbb{\nabla}^{SPH\ *}$ (see (3.69)) is not even first order consistent in general! The realistic results obtained in those cases can only be explained by two factors:

Firstly, more or less elaborate corrections are usually used (XSPH in [161], transport-velocity formulation in [4], ...), and their effect can be understood as ALE methods trying to remain close to nodal positions where the zeroth order dual consistency error remains low as developed in section 3.6.

Secondly, instead of aiming at a consistent approximation of the stresses, the SPH methodology focuses on conservation. Indeed, we can show that conservation of the total linear and angular momenta is a consequence of consistency properties of the differentiation operator dual to the one used in the momentum equation (G.4): in the absence of body and boundary forces, and if $\mathbb{\nabla}$ is zeroth order accurate,

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{P}}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t} \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} \rho \mathbf{x} \\
&= \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} \rho \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} && \text{(G.2) was used here} \\
&= \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} -\mathbb{\nabla}^* \cdot \sigma \\
&= \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} \sigma \cdot \mathbb{\nabla} 1 = \mathbf{0} && \text{(G.5)}
\end{aligned}$$

Similarly for the angular momentum,

$$\begin{aligned}
\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t} \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} \rho \mathbf{x} \times \mathbf{v} \\
&= \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} \rho \frac{\mathrm{d}\mathbf{x} \times \mathbf{v}}{\mathrm{d}t} && \text{(G.2) was used here} \\
&= \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} \rho \mathbf{v} \times \mathbf{v} + \mathbf{x} \times \rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} \\
&= \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} -\mathbf{x} \times \mathbb{\nabla}^* \cdot \sigma \\
&= \oint\!\!\!\!\!\!\diagup_{\mathcal{C}} \mathbb{\nabla} \times \mathbf{x} : \sigma = \mathbf{0} && \text{(G.6)}
\end{aligned}$$

Let us note that if $d \neq 3$ then $\mathbf{L}$ should be understood as a skew symmetric rank-2 tensor, and the result still holds. Hence, using $\mathbb{\nabla}^*$ in equation (G.4) entails that conservation of linear and angular momenta follow from $\mathbb{\nabla} 1 = \mathbf{0}$ and $\mathbb{\nabla} \times \mathbf{x} = \mathbf{0}$, which are consequences

of the covariance of the volume function with respect to isometric transformations (translations and rotations). This situation is analogous to the continuous one: linear and angular momenta are Noether invariants associated to isometric transformations of space.

This standard choice of dual gradient can further be justified in the specific case of Lagrangian systems. Let us postulate a discrete least action principle as stationarity of the following functional $S = \int L \, \mathrm{d}t$, and write the Lagrange functional as depending on particle positions *via* volume weights $L(V_1(\mathbf{x}_1, \dots), \dots, \mathbf{x}_1, \dots, \mathbf{v}_1, \dots)$. Then the corresponding Euler-Lagrange equations read: $\forall\ i \in \mathcal{C}$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \mathbf{v}_i} = \sum_{j\in\mathcal{C}} \frac{\partial L}{\partial V_j}\cdot\frac{\partial V_j}{\partial \mathbf{x}_i} + \frac{\partial L}{\partial \mathbf{x}_i} = -V_i \mathbb{\nabla}_j^* \frac{\partial L}{\partial V} + \frac{\partial L}{\partial \mathbf{x}_i} \tag{G.7}$$

Hence, regardless of the specific form of the Lagrangian, the relevant gradient operator to be used in the conservation of momentum is our dual gradient. A typical Lagrangian for hydrodynamics reads (see [162]) $L = \fint_{\mathcal{C}} \rho \left(\frac{1}{2}\mathbf{v}^2 - u(\rho) - \phi(\mathbf{x})\right)$, which becomes once ρ is replaced using conservation of mass:

$$L = \sum_{i\in\mathcal{C}} m_i \left(\frac{1}{2}\mathbf{v}_i^2 - u(m_i/V_i(\mathbf{x}_1, \dots, \mathbf{x}_n)) - \phi(\mathbf{x}_i)\right) \tag{G.8}$$

Defining the pressure field as $P_i = \rho_i^2 \frac{\partial u}{\partial \rho}(\rho_i)$ (this is the first law of thermodynamics) and writing $\mathbf{f}_i^{\mathrm{cons}} = -\frac{\partial \phi}{\partial \mathbf{x}}(\mathbf{x}_i)$, the conservation of momentum reads the same as (G.4) if we set $\sigma = -P\mathrm{I}_d$:

$$\rho_i \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\mathbb{\nabla}_i^* P + \rho_i \mathbf{f}_i^{\mathrm{cons}} \tag{G.9}$$

This variational derivation highlights the fact that the dual gradient effectively computes the infinitesimal work needed to displace a single node, relating volume and pressure as a conjugate pair of thermodynamic variables. This interpretation is consistent with our earlier interpretation of the dual gradient: forces are variations of energy with respect to displacements. The discrete variational principle from which equation (G.9) was derived gives it another desirable conservation property: the total energy H is conserved. This is a direct consequence of Noether's theorem, but let us still go through the steps of the computation: in the absence of boundary forces,

$$\begin{aligned}
\frac{\mathrm{d}H}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\fint_{\mathcal{C}} \rho\left(\frac{1}{2}\mathbf{v}^2 + u(\rho) + \phi(\mathbf{x})\right) \\
&= \fint_{\mathcal{C}} \rho \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{2}\mathbf{v}^2 + u(\rho) + \phi(\mathbf{x})\right) && \text{(G.2) was used here} \\
&= \fint_{\mathcal{C}} \rho\mathbf{v}\cdot\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} + \rho\frac{\partial u}{\partial\rho}\frac{\mathrm{d}\rho}{\mathrm{d}t} + \rho\frac{\partial\phi}{\partial\mathbf{x}}\cdot\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} \\
&= \fint_{\mathcal{C}} \mathbf{v}\cdot(-\mathbb{\nabla}^* P - \rho\mathbf{f}^{\mathrm{cons}} + \rho\mathbf{f}^{\mathrm{cons}}) - P\mathbb{\nabla}\cdot\mathbf{v} \\
&= -\fint_{\mathcal{C}} \mathbf{v}\cdot\mathbb{\nabla}^* P + P\mathbb{\nabla}\cdot\mathbf{v} \\
&= -\fint_{\partial\mathcal{C}} P\mathbf{v} = 0
\end{aligned}$$

At the discrete level, the power of internal forces is compensated by the variation of kinetic energy, and only boundary forces remain thanks to $\mathbb{\nabla}$ and $\mathbb{\nabla}^*$ being dual of each other.

Acronyms

ALE Arbitrary Lagrangian-Eulerian.
CDO Compatible Discrete Operator.
DEC Discrete Exterior Calculus.
DEM Diffuse Element Method.
DMLS Diffuse Moving Least Squares.
DOF Degree Of Freedom.
EFG Element Free Galerkin Method.
FD Finite Differences.
FEEC Finite Element Exterior Calculus.
FEM Finite Element Method.
FMLS Full Moving Least Squares.
FPM Finite Pointset Method.
FSI Fluid-Structure Interaction.
FTL Fortran Template Library.
FVM Finite Volume Method.
GMRES Generalized Minimal Residuals.
KDE Kernel Density Estimation.
LHS Left Hand Side.
LLS Local Least Squares.
MFD Mimetic Finite Differences.
MLPG Meshless Local Petrov-Galerkin.
MLS Moving Least Squares.
NS-FEM Node Smoothed Finite Element Method.
ODE Ordinary Differential Equation.
PDE Partial Differential Equation.
RBF Radial Basis Function.
RHS Right Hand Side.
RKHS Reproducing Kernel Hilbert Space.
RKPM Reproducing Kernel Particle Method.
RMD Renormalized Meshfree Derivatives.
SFEM Smoothed Finite Element Method.
SI International System of Units. Abbreviated from the French Système International d'unités.
SIF Stress Intensity Factor.
SPH Smooth Particle Hydrodynamics.
VEM Virtual Element Method.

Notations

$\mathrm{A}_n(\mathbb{A})$	Module of $n \times n$ skew-symmetric matrices with coefficients in the ring $\mathbb{A}$.
$B^p(\mathbf{x}, r)$	Open ball or cube of center $\mathbf{x}$ and radius r in $\mathbb{R}^p$.
B^p	Unit open ball or cube centered at the origin. Shorthand for $B^p(\mathbf{0}, 1)$.
∂_p	Boundary operator acting on p-chains.
$\#\mathcal{X}$	Cardinal of a set $\mathcal{X}$. If $\mathcal{X}$ is finite, $\#\mathcal{X}$ is the number of elements in $\mathcal{X}$.
$\not{C}^p$	A cell of intrinsic dimension p in an abstract cell complex or a mesh.
$\overline{\mathcal{S}}$	Closure of a subset $\mathcal{S}$ of a complete metrizable space.
$\mathcal{C}$	A point cloud.
$f(x, y) = \mathrm{cst}(x)$	An expression that is invariant with respect to y, but that might still vary with respect to x.
d	The intrinsic dimension of space, as opposed to the embedding dimension.
$D(\Omega)$	Diameter of a domain Ω.
$A \cdot B$	Contraction of two tensors on one pair of indices. In the case of two vectors, this is the dot product : $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T\mathbf{y}$. The symbol is sometimes omitted when there is no ambiguity as with the matrix-vector multiplication $A\mathbf{x} = A \cdot \mathbf{x}$
$\varnothing$	The empty set.
$\exists$	Existence quantifier. Reads "there exists".
$\exists\,!$	Uniqueness quantifier. Reads "there exists a unique".
d^p	Exterior derivative acting on p-forms (or p-cochains).
$\forall$	Universal quantifier. Reads "for all".
$(\mathcal{X} \to \mathcal{Y})$	Set of maps (or functions) from a set $\mathcal{X}$ (the domain) to a set $\mathcal{Y}$ (the co-domain).
$f : \mathcal{X} \to \mathcal{Y}$	A particular map from $\mathcal{X}$ to $\mathcal{Y}$. This is a shorthand notation for $f \in (\mathcal{X} \to \mathcal{Y})$.
$\mathrm{GL}_n(\mathbb{A})$	The general linear group : group of $n \times n$ invertible matrices with coefficients in the ring $\mathbb{A}$.
$\mathrm{Gr}_p(V)$	The Grassmannian : space of p-dimensionnal linear subspaces of a vector space V.
$L^p(\Omega, Y)$	Lebesgue space of functions with integrable p^{th} power with values in a normed Banach space Y. For $p = \infty$, this is understood as the space of essentially bounded Lebesgue measurable functions.
$\mathrm{Im}(f)$	Image of an homomorphism f.
I_n	Real identity matrix of order n.
ι	Inclusion map. If $\mathcal{X} \subset \mathcal{Y}$ then $\iota : \mathcal{X} \to \mathcal{Y}, \forall\, x \in \mathcal{X}, \iota(x) = x$.

$[\![p, q]\!]$	Inclusive integer interval bewteen p and q defined as $[\![p, q]\!] \stackrel{\text{def}}{=} \{n \in \mathbb{Z} \mid p \leq n \leq q\}$.
$\iota_{\mathbf{x}}$	Interior product with the vector field $\mathbf{x}$.
$\mathrm{Ker}(f)$	Kernel of an homomorphism f.
$\delta_{i,j}$	Kronecker symbol of i and j. Its value is 0 if $i \neq j$ and 1 if $i = j$.
$\mathbb{M}$	A manifold.
$\mathcal{M}$	A mesh.
$\mathrm{M}_n(\mathbb{A})$	Module of $n \times n$ matrices (i.e. n lines and n columns) with coefficients in the ring $\mathbb{A}$.
$\mathrm{M}_{n,m}(\mathbb{A})$	Module of $n \times m$ matrices (i.e. n lines and m columns) with coefficients in the ring $\mathbb{A}$.
n	The dimension of the affine embedding space.
$\mathrm{O}_n(\mathbb{R})$	Group of $n \times n$ real orthogonal matrices *i.e.* elements R of $\mathrm{M}_n(\mathbb{R})$ satisfying $\mathrm{RR}^T = \mathrm{R}^T\mathrm{R} = \mathrm{I}_n$.
$\mathbb{A}_p[X_1, \ldots, X_d]$	Space of d-variate polynomials with coefficients in the ring $\mathbb{A}$.
$\mathcal{P}(\mathcal{X})$	Powerset of a set $\mathcal{X}$, *i.e.* set of all subsets of $\mathcal{X}$.
$\mathrm{P}(V)$	Projective space of a vector space. Shorthand for $\mathrm{Gr}_1(V)$.
A^+	Moore-Penrose pseudo-inverse of a matrix. See annexe B for its definition and properties.
$\mathbb{R}$	The set of real numbers.
$\mathbb{R}^+$	The set of non-negative real numbers.
$\mathbb{R}^{+*}$	The set of positive real numbers.
$\mathbb{Z}$	The set of relative integers.
$\mathrm{SL}_n(\mathbb{A})$	The special linear group : group of $n \times n$ matrices with unit determinant and coefficients in the ring $\mathbb{A}$.
$\mathrm{S}_n(\mathbb{A})$	Module of $n \times n$ symmetric matrices with coefficients in the ring $\mathbb{A}$.
$\mathrm{S}_n^+(\mathbb{K})$	Cone of $n \times n$ symmetric positive semidefinite matrices with coefficients in the field $\mathbb{K}$.
$\mathrm{S}_n^{++}(\mathbb{K})$	Cone of $n \times n$ symmetric positive definite matrices with coefficients in the field $\mathbb{K}$.
$W_p^k(\Omega, Y)$	Sobolev space of functions u with weak derivatives $\frac{\partial u}{\partial^\alpha x}$ of order $\lvert\alpha\rvert$ up to k in $L^p(\Omega, Y)$
$S(\partial\Omega)$	Surface area of the boundary of a regular enough domain Ω.
$Sym(\mathcal{X})$	The symmetric group of $\mathcal{X}$, *i.e.* the set of bijective maps from $\mathcal{X}$ to itself.
$Sym(p)$	Shorthand for $Sym([\![1, p]\!])$.
$A \otimes B$	Tensor product of two tensors. In the particular case of vectors, we have $\mathbf{x} \otimes \mathbf{y} = \mathbf{x}\mathbf{y}^T$. When there is no possible ambiguity, we sometimes write $A \otimes B = AB$.
$V(\Omega)$	Volume of a measurable domain Ω.
$\alpha \wedge \beta$	Wedge product of two differential forms.

Bibliography

[1] COMSOL Inc. (2017). COMSOL Multiphysics®. Version 5.3.0.316.

[2] Abraham, R., Marsden, J., and Ratiu, T. (2012). *Manifolds, tensor analysis, and applications*, volume 75. Springer Science & Business Media.

[3] Abramovici, F. (1984). The Shepard interpolation as the best average of a set of data. Technical report, School of Math. Sci., Tel-Aviv University.

[4] Adami, S., Hu, X., and Adams, N. (2013). A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, 241:292–307.

[5] Adams, R. A. and Fournier, J. J. (2003). *Sobolev spaces*, volume 140. Academic press.

[6] Alayrangues, S., Damiand, G., Lienhardt, P., and Peltier, S. (2011). A Boundary Operator for Computing the Homology of Cellular Structures. Technical report, Université de Poitiers. 71 pages.

[7] Alayrangues, S., Daragon, X., Lachaud, J.-O., and Lienhardt, P. (2004). Equivalence between regular n-G-maps and n-surfaces. In *International Workshop on Combinatorial Image Analysis*, pages 122–136. Springer.

[8] Alayrangues, S., Daragon, X., Lachaud, J.-O., and Lienhardt, P. (2008). Equivalence between closed connected n-G-maps without multi-incidence and n-surfaces. *Journal of Mathematical Imaging and Vision*, 32(1):1–22.

[9] Aleksandrov, P. S. (1956). *Combinatorial topology*, volume 1. Courier Corporation.

[10] Arnold, D. N., Falk, R. S., and Winther, R. (2006). Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155.

[11] Atluri, S. and Zhu, T.-L. (2000). The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics. *Computational Mechanics*, 25(2-3):169–179.

[12] Babuška, I., Banerjee, U., Osborn, J., and Zhang, Q. (2009). Effect of numerical integration on meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 198(37):2886–2897.

[13] Babuška, I., Banerjee, U., and Osborn, J. E. (2003). Survey of meshless and generalized finite element methods: a unified approach. *Acta Numerica*, 12:1–125.

[14] Babuška, I., Banerjee, U., Osborn, J. E., and Li, Q. (2008). Quadrature for meshless methods. *International Journal for Numerical Methods in Engineering*, 76(9):1434–1470.

[15] Backus, G. and Gilbert, F. (1968). The resolving power of gross earth data. *Geophysical Journal International*, 16(2):169–205.

[16] Baldoni, V., Berline, N., De Loera, J., Köppe, M., and Vergne, M. (2011). How to integrate a polynomial over a simplex. *Mathematics of Computation*, 80(273):297–325.

[17] Barata, J. C. A. and Hussein, M. S. (2012). The Moore–Penrose pseudoinverse: A tutorial review of the theory. *Brazilian Journal of Physics*, 42(1-2):146–165.

[18] Barvinok, A. (1992). Exponential integrals and sums over convex polyhedra. *Funktsional. Anal. i Prilozhen*, 26(2):64–66.

[19] Baumgart, B. G. (1972). Winged edge polyhedron representation. Technical report, Stanford University Computer Science Department.

[20] Baumgart, B. G. (1975). A polyhedron representation for computer vision. In *Proceedings of the May 19-22, 1975, National Computer Conference and Exposition*, pages 589–596. ACM.

[21] Bažant, Z., Cedolin, L., Chen, W., and Lui, E. (2010). *Stability of structures: elastic, inelastic, fracture and damage theories*. World Scientific.

[22] Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L. D., and Russo, A. (2013). Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences*, 23(01):199–214.

[23] Beirão da Veiga, L., Brezzi, F., Marini, L. D., and Russo, A. (2014). The hitchhiker's guide to the virtual element method. *Mathematical Models and Methods in Applied Sciences*, 24(08):1541–1573.

[24] Belytschko, T. and Fleming, M. (1999). Smoothing, enrichment and contact in the element-free Galerkin method. *Computers & Structures*, 71(2):173–195.

[25] Belytschko, T., Guo, Y., Kam Liu, W., and Ping Xiao, S. (2000). A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering*, 48(9):1359–1400.

[26] Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., and Krysl, P. (1996). Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):3–47.

[27] Belytschko, T., Lu, Y., and Gu, L. (1994a). Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256.

[28] Belytschko, T., Lu, Y. Y., and Gu, L. (1994b). Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256.

[29] Ben-Israel, A. (1992). A volume associated with $m \times n$ matrices. *Linear Algebra and its Applications*, 167:87–111.

[30] Ben-Israel, A. (2002). Motzkin's transposition theorem, and the related theorems of Farkas, Gordan and Stiemke. *Encyclopaedia of Mathematics, Supplement III*.

[31] Ben Moussa, B. and Vila, J. (2000). Convergence of SPH method for scalar nonlinear conservation laws. *SIAM Journal on Numerical Analysis*, 37(3):863–887.

[32] Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Levine, J., Sharf, A., and Silva, C. (2014). State of the art in surface reconstruction from point clouds. In *EUROGRAPHICS star reports*, volume 1, pages 161–185.

[33] Bertrand, G. and Couprie, M. (1999). A model for digital topology. In *Proceedings of 8th Discrete Geometry for Computer Imagery (DGCI'99)*, pages 229–241. Springer, Berlin, Heidelberg.

[34] Bochev, P. and Hyman, J. (2006). Principles of mimetic discretizations of differential operators. *Compatible Spatial Discretizations*, pages 89–119.

[35] Bonelle, J. (2014). *Compatible Discrete Operator schemes on polyhedral meshes for elliptic and Stokes equations*. PhD thesis, Université Paris-Est.

[36] Bonet, J. and Kulasegaram, S. (2000). Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, 47(6):1189–1214.

[37] Bonet, J. and Kulasegaram, S. (2001). Remarks on tension instability of Eulerian and Lagrangian corrected smooth particle hydrodynamics (CSPH) methods. *International Journal for Numerical Methods in Engineering*, 52(11):1203–1220.

[38] Bonet, J. and Kulasegaram, S. (2002). A simplified approach to enhance the performance of smooth particle hydrodynamics methods. *Applied Mathematics and Computation*, 126(2-3):133–155.

[39] Bonet, J. and Lok, T. (1999). Variational and momentum preservation aspects of smooth particle hydrodynamic formulations. *Computer Methods in Applied Mechanics and Engineering*, 180(1):97 – 115.

[40] Bos, L. and Salkauskas, K. (1989). Moving least-squares are Backus-Gilbert optimal. *Journal of Approximation Theory*, 59(3):267–275.

[41] Brenner, S. and Scott, R. (2007). *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media.

[42] Brezzi, F., Lipnikov, K., and Shashkov, M. (2005a). Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM Journal on Numerical Analysis*, 43(5):1872–1896.

[43] Brezzi, F., Lipnikov, K., and Simoncini, V. (2005b). A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Mathematical Models and Methods in Applied Sciences*, 15(10):1533–1551.

[44] Brion, M. (1988). Points entiers dans les polyèdres convexes. *Ann. Sci. Ecole Norm. Sup*, 21(4):653–663.

[45] Brisson, E. (1989). Representing geometric structures in d dimensions: topology and order. In *Proceedings of the fifth annual symposium on Computational Geometry*, pages 218–227. ACM.

[46] Brisson, E. (1993). Representing geometric structures in d dimensions: Topology and order. *Discrete & Computational Geometry*, 9(1):387–426.

[47] Büeler, B., Enge, A., and Fukuda, K. (2000). Exact volume computation for polytopes: a practical study. In *Polytopes—Combinatorics and Computation*, pages 131–154. Springer.

[48] Bui, H. and Van, K. D. (1974). Trois problèmes non linéaires de l'étude mécanique de la rupture des matériaux. *Revue de Physique Appliquée*, 9(4):605–610.

[49] Bui, H. D. (1978). *Mécanique de la rupture fragile*. Masson.

[50] Cardoze, D. E., Miller, G. L., and Phillips, T. (2006). Representing topological structures using cell-chains. In *International Conference on Geometric Modeling and Processing*, pages 248–266. Springer.

[51] Carré, G., Del Pino, S., Després, B., and Labourasse, E. (2009). A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228(14):5160–5183.

[52] Castillo, J. E. and Miranda, G. F. (2013). *Mimetic discretization methods*. CRC Press.

[53] Chen, J., Wu, C., Yoon, S., and You, Y. (2001a). A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466.

[54] Chen, J.-S., Hillman, M., and Rüter, M. (2013). An arbitrary order variationally consistent integration for Galerkin meshfree methods. *International Journal for Numerical Methods in Engineering*, 95(5):387–418.

[55] Chen, J.-S., Wu, C.-T., Yoon, S., and You, Y. (2001b). A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50(2):435–466.

[56] Chen, J.-S., Yoon, S., and Wu, C.-T. (2002). Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 53(12):2587–2615.

[57] Chiu, E. K.-y. and Jameson, A. (2010). An edge-averaged semi-meshless framework for numerical solution of conservation laws. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1269.

[58] Chiu, E. K.-y., Wang, Q., Hu, R., and Jameson, A. (2012). A conservative mesh-free scheme and generalized framework for conservation laws. *SIAM Journal on Scientific Computing*, 34(6):A2896–A2916.

[59] Chiu, E. K.-y., Wang, Q., and Jameson, A. (2011). A conservative meshless scheme: general order formulation and application to Euler equations. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 651.

[60] Chiu, K. (2011). *A conservative meshless framework for conservation laws with applications in computational fluid dynamics*. Stanford University.

[61] Cholaquidis, A., Cuevas, A., Fraiman, R., et al. (2014). On Poincaré cone property. *The Annals of Statistics*, 42(1):255–284.

[62] Clair, G., Després, B., and Labourasse, E. (2013). A new method to introduce constraints in cell-centered Lagrangian schemes. *Computer Methods in Applied Mechanics and Engineering*, 261:56–65.

[63] Crane, K., de Goes, F., Desbrun, M., and Schröder, P. (2013). Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 courses*, SIGGRAPH '13, New York, NY, USA. ACM.

[64] Damelin, S. B. (2008). On bounds for diffusion, discrepancy and fill distance metrics. In Gorban, A. N., Kégl, B., Wunsch, D. C., and Zinovyev, A. Y., editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 261–270, Berlin, Heidelberg. Springer Berlin Heidelberg.

[65] Damiand, G. and Lienhardt, P. (2014). *Combinatorial maps: efficient data structures for computer graphics and image processing*. CRC Press.

[66] Danaraj, G. and Klee, V. (1978). A representation of 2-dimensional pseudomanifolds and its use in the design of a linear-time shelling algorithm. In *Annals of Discrete Mathematics*, volume 2, pages 53–63. Elsevier.

[67] de Goes, F., Wallez, C., Huang, J., Pavlov, D., and Desbrun, M. (2015). Power particles: an incompressible fluid solver based on power diagrams. *ACM Trans. Graph.*, 34(4):50–1.

[68] De Loera, J. A. (2010). Actually doing it: Polyhedral computation and its applications. *Manuscript in progress*.

[69] De Loera, J. A., Dutra, B., Köppe, M., Moreinis, S., Pinto, G., and Wu, J. (2013). Software for exact integration of polynomials over polyhedra. *Computational Geometry*, 46(3):232–252.

[70] Delfour, M. and Zolésio, J.-P. (2011). *Shapes and geometries: metrics, analysis, differential calculus, and optimization*. SIAM.

[71] Desbrun, M., Hirani, A. N., Leok, M., and Marsden, J. E. (2005). Discrete exterior calculus. *arXiv preprint math/0508341*.

[72] Desbrun, M., Kanso, E., and Tong, Y. (2008). Discrete differential forms for computational modeling. In *Discrete Differential Geometry*, pages 287–324. Springer.

[73] Després, B. (2010). Weak consistency of the cell-centered Lagrangian GLACE scheme on general meshes in any dimension. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2669–2679.

[74] Després, B. and Labourasse, E. (2015). Angular momentum preserving cell-centered Lagrangian and Eulerian schemes on arbitrary grids. *Journal of Computational Physics*, 290:28–54.

[75] Destuynder, P., Djaoua, M., and Lescure, S. (1983). Quelques remarques sur la mécanique de la rupture élastique. *Journal de mécanique théorique et appliquée*, 2(1):113–135.

[76] Destuynder, P., Djaoua, P. E. M., Chesnay, L., and Nedelec, J. (1981). Sur une interprétation mathématique de l'intégrale de Rice en théorie de la rupture fragile. *Mathematical Methods in the Applied Sciences*, 3(1):70–87.

[77] Dilts, G. A. (1999). Moving-least-squares-particle hydrodynamics—I. consistency and stability. *International Journal for Numerical Methods in Engineering*, 44(8):1115–1155.

[78] Dilts, G. A. (2000). Moving least-squares particle hydrodynamics II: conservation and boundaries. *International Journal for Numerical Methods in Engineering*, 48(10):1503–1524.

[79] Dorff, M. and Hall, L. (2003). Solids in Rn whose area is the derivative of the volume. *College Mathematics Journal*, 34(5):350–358.

[80] Dupont, T. and Scott, R. (1980). Polynomial approximation of functions in Sobolev spaces. *Mathematics of Computation*, 34(150):441–463.

[81] Dyer, M. E. and Frieze, A. M. (1988). On the complexity of computing the volume of a polyhedron. *SIAM Journal on Computing*, 17(5):967–974.

[82] Dyka, C. and Ingel, R. (1995). An approach for tension instability in smoothed particle hydrodynamics (SPH). *Computers & Structures*, 57(4):573–580.

[83] Dyka, C., Randles, P., and Ingel, R. (1997). Stress points for tension instability in SPH. *International Journal for Numerical Methods in Engineering*, 40(13):2325–2341.

[84] Dyn, N. (1987). Interpolation of scattered data by radial functions. In *Topics in Multivariate Approximation*, pages 47–61. Elsevier.

[85] E Oliveira, E. (1977). The patch test and the general convergence criteria of the finite element method. *International Journal of Solids and Structures*, 13(3):159–178.

[86] Eisenberg, M. A. and Malvern, L. E. (1973). On finite element integration in natural co-ordinates. *International Journal for Numerical Methods in Engineering*, 7(4):574–575.

[87] Ellero, M., Serrano, M., and Espanol, P. (2007). Incompressible smoothed particle hydrodynamics. *Journal of Computational Physics*, 226(2):1731–1752.

[88] Evans, L. (1996). *Partial Differential Equations*. American Mathematical Society.

[89] Farkas, J. (1902). Über die Theorie der einfachen Ungleichungen. *Journal für die Reine und Angewandte Mathematik*, 124:1–27.

[90] Fasshauer, G. (2003). Lecture notes in multivariate meshfree approximation. http://www.math.iit.edu/~fass/603_handouts.html.

[91] Fasshauer, G. E. (1996). Solving partial differential equations by collocation with radial basis functions. In *Proceedings of Chamonix*, volume 1997, pages 1–8. Vanderbilt University Press Nashville, TN.

[92] Fattal, R. (2011). Blue-noise point sampling using kernel density model. In *ACM Transactions on Graphics (TOG)*, volume 30, page 48. ACM.

[93] Faure, H., Kritzer, P., and Pillichshammer, F. (2015). From van der Corput to modern constructions of sequences for quasi-Monte Carlo rules. *Indagationes Mathematicae*, 26(5):760–822.

[94] Fernandes, R. L. (Spring 2014). University of Illinois course on differentiable manifolds. https://faculty.math.illinois.edu/~ruiloja/Math519/. Accessed February 27$^{\text{th}}$ 2018.

[95] Fernández-Méndez, S., Díez, P., and Huerta, A. (2003). Convergence of finite elements enriched with mesh-less methods. *Numerische Mathematik*, 96(1):43–59.

[96] Fernández-Méndez, S. and Huerta, A. (2003). Coupling finite elements and particles for adaptivity: An application to consistently stabilized convection-diffusion. In *Meshfree Methods for Partial Differential Equations*, pages 117–129. Springer.

[97] Fernández-Méndez, S. and Huerta, A. (2004). Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1257–1275.

[98] Fleming, M., Chu, Y., Moran, B., and Belytschko, T. (1997). Enriched element-free Galerkin methods for crack tip fields. *International Journal for Numerical Methods in Engineering*, 40(8):1483–1504.

[99] Floater, M. (2015). Generalized barycentric coordinates and applications. *Acta Numerica*, 24:161–214.

[100] Frankel, T. (2011). *The geometry of physics: an introduction*. Cambridge University Press.

[101] Garimella, R. V., Kim, J., and Berndt, M. (2014). Polyhedral mesh generation and optimization for non-manifold domains. In *Proceedings of the 22nd International Meshing Roundtable*, pages 313–330. Springer.

[102] Gawlik, E. S., Mullen, P., Pavlov, D., Marsden, J. E., and Desbrun, M. (2011). Geometric, variational discretization of continuum theories. *Physica D: Nonlinear Phenomena*, 240(21):1724–1760.

[103] Gilbarg, D. and Trudinger, N. (2015). *Elliptic partial differential equations of second order*. Springer.

[104] Gingold, R. and Monaghan, J. (1982). Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*, 46(3):429–453.

[105] Gingold, R. A. and Monaghan, J. (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181(3):375–389.

[106] Golub, G. H. and Pereyra, V. (1973). The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432.

[107] Gordan, P. (1873). Über die Auflösung linearer Gleichungen mit reellen Coefficienten. *Mathematische Annalen*, 6(1):23–28.

[108] Grasset-Simon, C., Damiand, G., and Lienhardt, P. (2005). nD generalized map pyramids: three equivalent representations. Technical report, Université de Poitiers.

[109] Gravin, N., Pasechnik, D. V., Shapiro, B., and Shapiro, M. (2012). On moments of a polytope. *arXiv preprint arXiv:1210.3193*.

[110] Griebel, M. and Schweitzer, M. A. (2003). A particle-partition of unity method part V: boundary conditions. In *Geometric Analysis and Nonlinear Partial Differential Equations*, pages 519–542. Springer.

[111] Griffiths, P. and Harris, J. (2014). *Principles of algebraic geometry*. John Wiley & Sons.

[112] Grünbaum, B. (2003). Are your polyhedra the same as my polyhedra? In *Discrete and Computational Geometry*, pages 461–488. Springer.

[113] Guinea, G., Planas, J., and Elices, M. (2000). K_I evaluation by the displacement extrapolation technique. *Engineering Fracture Mechanics*, 66(3):243–255.

[114] Günther, F. C. and Liu, W. K. (1998). Implementation of boundary conditions for meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 163(1-4):205–230.

[115] Hatcher, A. (2002). Algebraic topology. 2002. *Cambridge University Press*, 606(9).

[116] Hiptmair, R. (2001). Discrete Hodge operators. *Numerische Mathematik*, 90(2):265–289.

[117] Hirani, A. N. (2003). *Discrete exterior calculus*. PhD thesis, California Institute of Technology.

[118] Hiyoshi, H. and Sugihara, K. (2002). Improving continuity of Voronoi-based interpolation over Delaunay spheres. *Computational Geometry*, 22(1-3):167–183.

[119] Huerta, A. and Fernandez Mendez, S. (2000). Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48(11):1615–1636.

[120] Huerta, A., Fernández-Méndez, S., and Díez, P. (2002). Enrichissement des interpolations d'éléments finis en utilisant des méthodes sans maillage. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(6):1027–1042.

[121] Hyman, J. M. and Shashkov, M. (2001). Mimetic finite difference methods for Maxwell's equations and the equations of magnetic diffusion. *Progress in Electromagnetics Research*, 32:89–121.

[122] Iwaniec, T., Kauhanen, J., Kravetz, A., and Scott, C. (2004). The Hadamard-Schwarz inequality. *Journal of Function Spaces and Applications*, 2(2):191–215.

[123] Johnson, G. R. and Beissel, S. R. (1996). Normalized smoothing functions for SPH impact computations. *International Journal for Numerical Methods in Engineering*, 39(16):2725–2741.

[124] Kamoulakos, A. and Groenenboom, P. (1998). Moving from FE to SPH for space debris impact simulations – experience with PAM-SHOCK. *European Conference on Spacecraft Structures, Materials and Mechanical Testing*.

[125] Kamoulakos, A., Przybylowicz, M., and Groenenboom, P. (1997a). Smoothed particle hydrodynamics for space debris impact simulations: An approach with the PAM-SHOCK transient dynamics code. *International Workshop in New Models and Numerical Codes for Shock Wave Processes in Condensed Media*.

[126] Kamoulakos, A., Przybylowicz, M., and Lambert, M. (1997b). Virtual design for space debris impact (validation of the PAM-SHOCK transient dynamics code). *Second European Conference on Space Debris*, pages 455–458.

[127] Khachiyan, L. (1993). Complexity of polytope volume computation. In *New trends in Discrete and Computational Geometry*, pages 91–101. Springer.

[128] Khosravifard, M., Esmaeili, M., and Saidi, H. (2009). Extension of the Lasserre-Avrachenkov theorem on the integral of multilinear forms over simplices. *Applied Mathematics and Computation*, 212(1):94–99.

[129] Kitson, A., McLachlan, R. I., and Robidoux, N. (2003). Skew-adjoint finite difference methods on nonuniform grids. *New Zealand J. Math*, 32(2):139–159.

[130] Konerth, N. (2014). Exact multivariate integration on simplices: an explanation of the Lasserre-Avrachenkov theorem. Technical report, American University Washington, D.C.

[131] Krongauz, Y. and Belytschko, T. (1996). Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131(1-2):133–145.

[132] Krongauz, Y. and Belytschko, T. (1997). Consistent pseudo-derivatives in meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 146(3-4):371–386.

[133] Lancaster, P. and Salkauskas, K. (1981). Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158.

[134] Lanson, N. (2001). *Etude des méthodes particulaires renormalisées: applications aux problemes de dynamique rapide*. PhD thesis, Toulouse, INSA.

[135] Lanson, N. and Vila, J. (2008). Renormalized meshfree schemes I: consistency, stability, and hybrid methods for conservation laws. *SIAM Journal on Numerical Analysis*, 46(4):1912–1934.

[136] Lasserre, J. (1998). Integration on a convex polytope. *Proceedings of the American Mathematical Society*, 126(8):2433–2441.

[137] Lasserre, J. B. (1983). An analytical expression and an algorithm for the volume of a convex polyhedron in Rn. *Journal of optimization theory and applications*, 39(3):363–377.

[138] Lasserre, J. B. and Avrachenkov, K. E. (2001). The multi-dimensional version of $\int_a^b x^p dx$. *The American Mathematical Monthly*, 108(2):151–154.

[139] Lawrence, J. (1991). Polytope volume computation. *Mathematics of Computation*, 57(195):259–271.

[140] LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems*, volume 31. Cambridge University Press.

[141] Li, S. and Liu, W. K. (2007). *Meshfree particle methods*. Springer Science & Business Media.

[142] Lienhardt, P. (1989). Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *Proceedings of the fifth annual symposium on Computational Geometry*, pages 228–236. ACM.

[143] Lind, S., Xu, R., Stansby, P., and Rogers, B. D. (2012). Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves. *Journal of Computational Physics*, 231(4):1499–1523.

[144] Lipnikov, K., Manzini, G., and Shashkov, M. (2014). Mimetic finite difference method. *Journal of Computational Physics*, 257:1163–1227.

[145] Lipnikov, K., Shashkov, M., and Svyatskiy, D. (2006). The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes. *Journal of Computational Physics*, 211(2):473–491.

[146] Litvinov, S., Hu, X., and Adams, N. (2015). Towards consistence and convergence of conservative SPH approximations. *Journal of Computational Physics*, 301:394–401.

[147] Liu, G. (2010a). A G–space theory and a weakened weak (W2) form for a unified formulation of compatible and incompatible methods: Part I theory. *International Journal for Numerical Methods in Engineering*, 81(9):1093–1126.

[148] Liu, G. (2010b). A G–space theory and a weakened weak (W2) form for a unified formulation of compatible and incompatible methods: Part II applications to solid mechanics problems. *International Journal for Numerical Methods in Engineering*, 81(9):1127–1156.

[149] Liu, G., Nguyen-Thoi, T., Nguyen-Xuan, H., and Lam, K. (2009). A node-based smoothed finite element method (NS-FEM) for upper bound solutions to solid mechanics problems. *Computers & Structures*, 87(1):14–26.

[150] Liu, G.-R. and Trung, N. T. (2016). *Smoothed finite element methods*. CRC press.

[151] Liu, W. K., Jun, S., Li, S., Adee, J., and Belytschko, T. (1995). Reproducing kernel particle methods for structural dynamics. *International Journal for Numerical Methods in Engineering*, 38(10):1655–1679.

[152] Long, T., Hu, D., Wan, D., Zhuang, C., and Yang, G. (2017). An arbitrary boundary with ghost particles incorporated in coupled FEM–SPH model for FSI problems. *Journal of Computational Physics*, 350:166–183.

[153] Loubère, R., Shashkov, M., and Wendroff, B. (2008). Volume consistency in a staggered grid Lagrangian hydrodynamics scheme. *Journal of Computational Physics*, 227(8):3731–3737.

[154] Lucy, L. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024.

[155] Madhavan, P. (2010). Numerical integration in meshfree methods. Master's thesis, University of Oxford.

[156] Margolin, L. G., Shashkov, M., and Smolarkiewicz, P. K. (2000). A discrete operator calculus for finite difference approximations. *Computer Methods in Applied Mechanics and Engineering*, 187(3-4):365–383.

[157] Marshall, J. (2005). University of Iowa course of computer-aided engineering. http://user.engineering.uiowa.edu/~cae1/Presentations/. Accessed February 2nd 2018.

[158] Mikhailova, N., Tishkin, V., Tyurina, N., Favorskii, A., and Shashkov, M. (1986). Numerical modelling of two-dimensional gas-dynamic flows on a variable-structure mesh. *USSR Computational Mathematics and Mathematical Physics*, 26(5):74–84.

[159] Mirzaei, D., Schaback, R., and Dehghan, M. (2011). On generalized moving least squares and diffuse derivatives. *IMA Journal of Numerical Analysis*, 32(3):983–1000.

[160] Monaghan, J. (1989). On the problem of penetration in particle methods. *Journal of Computational Physics*, 82(1):1–15.

[161] Monaghan, J. (1992). Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574.

[162] Monaghan, J. (2005). Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703.

[163] Monaghan, J. and Price, D. (2001). Variational principles for relativistic smoothed particle hydrodynamics. *Monthly Notices of the Royal Astronomical Society*, 328(2):381–392.

[164] Monaghan, J. J. (2000). SPH without a tensile instability. *Journal of Computational Physics*, 159(2):290–311.

[165] Monaghan, J. J. and Kajtar, J. B. (2009). SPH particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180(10):1811–1820.

[166] Mondino, A., Spadaro, E., et al. (2017). On an isoperimetric-isodiametric inequality. *Analysis & PDE*, 10(1):95–126.

[167] Motzkin, T. S. (1936). *Beiträge zur Theorie der linearen Ungleichungen*. Azriel Press.

[168] Motzkin, T. S. (1951). Two consequences of the transposition theorem on linear inequalities. *Econometrica (pre-1986)*, 19(2):184.

[169] Moussa, B. B., Lanson, N., and Vila, J. P. (1999). Convergence of meshless methods for conservation laws applications to Euler equations. In *Hyperbolic Problems: Theory, Numerics, Applications*, pages 31–40. Springer.

[170] Mukherjee, Y. and Mukherjee, S. (1997). On boundary conditions in the element-free Galerkin method. *Computational Mechanics*, 19(4):264–270.

[171] Mullen, P., De Goes, F., Desbrun, M., Cohen-Steiner, D., and Alliez, P. (2010). Signing the unsigned: Robust surface reconstruction from raw pointsets. In *Computer Graphics Forum*, volume 29, pages 1733–1741. Wiley Online Library.

[172] Mullen, P., McKenzie, A., Pavlov, D., Durant, L., Tong, Y., Kanso, E., Marsden, J. E., and Desbrun, M. (2011a). Discrete Lie advection of differential forms. *Foundations of Computational Mathematics*, 11(2):131–149.

[173] Mullen, P., Memari, P., de Goes, F., and Desbrun, M. (2011b). HOT: Hodge-optimized triangulations. *ACM Transactions on Graphics (TOG)*, 30(4):103.

[174] Nayroles, B., Touzot, G., and Villon, P. (1992). Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5):307–318.

[175] Nitsche, J. (1971). Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer.

[176] Noll, W. (1965). Proof of the maximality of the orthogonal group in the unimodular group. *Archive for Rational Mechanics and Analysis*, 18(2):100–102.

[177] Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

[178] Pavlov, D., Mullen, P., Tong, Y., Kanso, E., Marsden, J. E., and Desbrun, M. (2011). Structure-preserving discretization of incompressible fluids. *Physica D: Nonlinear Phenomena*, 240(6):443–458.

[179] Perot, B. (2000). Conservation properties of unstructured staggered mesh schemes. *Journal of Computational Physics*, 159(1):58–89.

[180] Pierrot, G. (2008). Discrete Ostrogradsky property recovery: symmetrisation of MLS coefficients. Technical report, ESI Group. Lib-CFD specifications.

[181] Pierrot, G. (2010). A gentle introduction to ESI i-adjoint library.

[182] Pierrot, G. (2011). Meshless methods: a unified framework and analysis. Technical report, ESI Group.

[183] Pierrot, G. and Fougeron, G. (2015). Enforcing differentiation/integration compatibility in meshless methods with nodal integration. In *IV International Conference on Particle-Based Methods – Fundamentals and Applications*.

[184] Pierrot, G. and Fougeron, G. (2016). Discrete gradient theorem and element-based integration in meshless methods. Presentation at the ECCOMAS Congress 2016, Crete Island, Greece.

[185] Pilkey, W. (1994). *Formulas for stress, strain, and structural matrices*. J. Wiley New York.

[186] Power, H. and Barraco, V. (2002). A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations. *Computers & Mathematics with Applications*, 43(3-5):551–583.

[187] Price, D. (2012). Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794.

[188] Pukhlikov, A. V. and Khovanskii, A. G. (1992). The Riemann–Roch theorem for integrals and sums of quasipolynomials on virtual polytopes. *Algebra i analiz*, 4(4):188–216.

[189] Qian, J. (2009). Discrete gradient method in solid mechanics. *Ph. D. Thesis*.

[190] Qian, J., Li, J., and Wang, S. (2013). The generalized Riemann problems for hyperbolic balance laws: A unified formulation towards high order. *arXiv preprint arXiv:1303.2941*.

[191] Rabczuk, T., Xiao, S. P., and Sauer, M. (2006). Coupling of mesh-free methods with finite elements: basic concepts and test results. *Communications in Numerical Methods in Engineering*, 22(10):1031–1065.

[192] Randles, P. and Libersky, L. (1996a). Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):375–408.

[193] Randles, P. and Libersky, L. (1996b). Smoothed particle hydrodynamics: some recent improvements and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):375–408.

[194] Ranganathan, D. (2010). A gentle introduction to grassmannians.

[195] Rice, J. et al. (1968). A path independent integral and the approximate analysis of strain concentration by notches and cracks. In *ASME*.

[196] Rigo, M. (2016). Perron–Frobenius theory. *Advanced Graph Theory and Combinatorics*, pages 183–232.

[197] Rooke, D. P. and Cartwright, D. J. (1976). Compendium of stress intensity factors. *Procurement Executive, Ministry of Defence. H. M. S. O. 1976, 330 p(Book)*.

[198] Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.

[199] Samarskii, A., Tishkin, V., Favorskii, A., and Shashkov, M. (1981). Operational finite-difference schemes. *Differential Equations*, 17(7):854–862.

[200] Schaap, W. (2007). The Delaunay tessellation field estimator. *Ph. D. Thesis*.

[201] Schaback, R. and Wendland, H. (2006). Kernel techniques: from machine learning to meshless methods. *Acta Numerica*, 15:543–639.

[202] Schechter, H. and Bridson, R. (2012). Ghost SPH for animating water. *ACM Transactions on Graphics (TOG)*, 31(4):61.

[203] Schoenberg, I. J. (1988). Contributions to the problem of approximation of equidistant data by analytic functions. In *IJ Schoenberg Selected Papers*, pages 3–57. Springer.

[204] Serrano, M. and Español, P. (2000). Thermodynamically consistent mesoscopic fluid particle models for a Van der Waals fluid. *arXiv preprint cond-mat/0002026*.

[205] Serrano, M. and Espanol, P. (2001). Thermodynamically consistent mesoscopic fluid particle model. *Physical Review E*, 64(4):046115.

[206] Serrano, M., Español, P., and Zúñiga, I. (2005). Voronoi fluid particle model for Euler equations. *Journal of Statistical Physics*, 121(1-2):133–147.

[207] Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517–524. ACM.

[208] Sibson, R. (1980). A vector identity for the Dirichlet tessellation. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 87-1, pages 151–155. Cambridge University Press.

[209] Smith, G. (1970). On a fundamental error in two papers of C.-C. Wang "on representations for isotropic functions, parts I and II". *Archive for Rational Mechanics and Analysis*, 36(3):161–165.

[210] Solovev, A., Soloveva, E., Tishkin, V., Favorskii, A., and Shashkov, M. Y. (1986). Approximation of finite-difference operators on a mesh of Dirichlet cells. *Differential Equations*, 22(7):863–872.

[211] Sommariva, A. and Vianello, M. (2006a). Meshless cubature by Green's formula. *Applied Mathematics and Computation*, 183(2):1098–1107.

[212] Sommariva, A. and Vianello, M. (2006b). Numerical cubature on scattered data by radial basis functions. *Computing*, 76(3-4):295.

[213] Son, N. H. (2009). An enforced essential boundary condition by penalty method in the element-free Galerkin (EFG) methods. *Vietnam Journal of Mechanics*, 31(2):122–132.

[214] Sternberg, S. (1999). *Supersymmetry and Equivariant de Rham Theory*. Math. Past Present, Springer, Berlin.

[215] Stewart, G. W. (1977). On the perturbation of pseudo-inverses, projections and linear least squares problems. *SIAM review*, 19(4):634–662.

[216] Stiemke, E. (1915). Über positive Lösungen homogener linearer Gleichungen. *Mathematische Annalen*, 76(2-3):340–342.

[217] Strang, G. and Fix, G. (1973). *An analysis of the finite element method*, volume 212. Prentice-hall Englewood Cliffs, NJ.

[218] Stummel, F. (1980). The limitations of the patch test. *International Journal for Numerical Methods in Engineering*, 15(2):177–188.

[219] Suo, X.-Z. (1990). *Analyse mathématique et numérique de la propagation des fissures par le modèle de multi-couronnes*. PhD thesis, Ecole Nationale des Ponts et Chaussées.

[220] Swegle, J., Hicks, D., and Attaway, S. (1995). Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116(1):123–134.

[221] Swegle, J. W. (2000). Conservation of momentum and tensile instability in particle methods. Technical report, Sandia National Labs.

[222] Sylvie, A. and Jacques-Olivier, L. (2002). Equivalence between order and cell complex representations. In *Proceedings of the Computer Vision Winter Workshop*, pages 222–233.

[223] Syrakos, A., Varchanis, S., Dimakopoulos, Y., Goulas, A., and Tsamopoulos, J. (2017). On the order of accuracy of the divergence theorem (Green-Gauss) method for calculating the gradient in finite volume methods. *<hal-01532882v2>*.

[224] Taylor, R., Simo, J., Zienkiewicz, O., and Chan, A. (1986). The patch test—a condition for assessing FEM convergence. *International Journal for Numerical Methods in Engineering*, 22(1):39–62.

[225] Trevorgoodchild (2018). User Trevorgoodchild on Wikipedia. https://en.wikipedia.org/wiki/User:Trevorgoodchild. Accessed March 16th 2018.

[226] Versteeg, H. K. and Malalasekera, W. (2007). *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education.

[227] Vila, J. P. (2005). SPH renormalized hybrid methods for conservation laws: applications to free surface flows. In *Meshfree Methods for Partial Differential Equations II*, pages 207–229. Springer.

[228] Wang, C. C. (1969). On representations for isotropic functions. part I. isotropic functions of symmetric tensors and vectors. *Archive for Rational Mechanics and Analysis*, 33:249–267.

[229] Watkins, D. S. (1979). A generalization of the Bramble-Hilbert lemma and applications to multivariate interpolation. *Journal of Approximation Theory*, 26(3):219–231.

[230] Wendland, H. (1996). Konstruktion und Untersuchung radialer Basisfunktionen mit kompaktem Träger. *Göttingen, Georg-August-Universität zu Göttingen, Diss.*

[231] Wendland, H. (1999). Meshless Galerkin methods using radial basis functions. *Mathematics of Computation of the American Mathematical Society*, 68(228):1521–1531.

[232] Wendland, H. (2001). Local polynomial reproduction and moving least squares approximation. *IMA Journal of Numerical Analysis*, 21(1):285–300.

[233] Wendland, H. (2004). *Scattered data approximation*, volume 17. Cambridge University Press.

[234] Woźnicki, Z. I. (2009). *Solving Linear Systems: An Analysis of Matrix Prefactorization Iterative Methods.* Matrix Editions.

[235] Zeng, W. and Liu, G. (2018). Smoothed finite element methods (S-FEM): an overview and recent developments. *Archives of Computational Methods in Engineering*, 25(2):397–435.

[236] Zhang, Q. (2014). Quadrature for meshless Nitsche's method. *Numerical Methods for Partial Differential Equations*, 30(1):265–288.

[237] Zhang, X., Liu, X., Lu, M.-W., and Chen, Y. (2001). Imposition of essential boundary conditions by displacement constraint equations in meshless methods. *Communications in Numerical Methods in Engineering*, 17(3):165–178.

[238] Zhang, X., Song, K. Z., Lu, M. W., and Liu, X. (2000). Meshless methods based on collocation with radial basis functions. *Computational Mechanics*, 26(4):333–343.

[239] Zhu, T. and Atluri, S. (1998). A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. *Computational Mechanics*, 21(3):211–222.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre: Contribution à l'amélioration des méthodes sans maillage appliquées à la mécanique des milieux continus
Mots clefs: Sans-maillage; Simulation; Mécanique des milieux continus; SPH
Résumé: Cette thèse présente un cadre général pour l'étude de schémas de discrétisation nodaux sans maillage formulé en termes d'opérateurs discrets définis sur un nuage de points : intégration volumique et de bord, gradient et opérateur de reconstruction. Ces définitions dotent le nuage de points d'une structure plus faible que celle définie par un maillage, mais partageant avec elle certain concepts fondamentaux. Le plus important d'entre eux est la condition de compatibilité intégro-différentielle. Avec la consistance linéaire du gradient discret, cet analogue discret de la formule de Stokes constitue une condition nécessaire à la consistance linéaire des opérateurs elliptiques en formulation faible. Sa vérification, au moins de manière approchée, permet d'écrire des discrétisations dont le taux de convergence est optimal. La construction d'opérateurs discrets compatibles est si difficile que nous conjecturons – sans parvenir à le démontrer – qu'elle nécessite soit la résolution d'un système linéaire global, soit la construction d'un maillage : c'est "la malédiction sans-maillage". Trois grandes approches pour la construction d'opérateurs discrets compatibles sont étudiées. Premièrement, nous proposons une méthode de correction permettant de calculer l'opérateur gradient compatible le plus proche – au sens des moindres carrés – sans mettre à mal la consistance linéaire. Dans le cas particulier des gradients DMLS, nous montrons que le gradient corrigé est en réalité globalement optimal. Deuxièmement, nous adaptons l'approche SFEM au cadre opérateur et constatons qu'elle définit des opérateurs consistants à l'ordre un et compatibles. Nous proposons une méthode d'intégration discrète exploitant la relation topologique entre les cellules et les faces d'un maillage qui préserve ces caractéristiques. Troisièmement, nous montrons qu'il est possible de générer tous les opérateurs sans maillage rien qu'avec la donnée d'une formule d'intégration volumique nodale en exploitant la dépendance fonctionnelle des poids volumiques nodaux par rapport à la position des nœuds du nuage, l'espace continu sous-jacent et le nombre de nœuds. Les notions de consistance des différents opérateurs sont caractérisées en termes des poids volumiques initiaux, formant un jeu de recommandation pour la mise au point de bonnes formules d'intégration. Dans ce cadre, nous réinterprétons les méthodes classiques de stabilisation de la communauté SPH comme cherchant à annuler l'erreur sur la formule de Stokes discrète. L'exemple des opérateurs SFEM trouve un équivalent en formulation volume, ainsi que la méthode d'intégration discrète s'appuyant sur un maillage. Son écriture nécessite toutefois une description très précise de la géométrie des cellules du maillage, en particulier dans le cas où les faces ne sont pas planes. Nous menons donc à bien une caractérisation complète de la forme de telles cellules uniquement en fonction de la position des nœuds du maillage et des relations topologiques entre les cellules, permettant une définition sans ambigüité de leur volume et centre de gravité. Enfin, nous décrivons des schémas de discrétisation d'équations elliptiques utilisant les opérateurs sans-maillage et proposons plusieurs possibilités pour traiter les conditions au bord tout en imposant le moins de contraintes sur la position des nœuds du nuage de points. Nous donnons des résultats numériques confirmant l'importance capitale de vérifier les conditions de compatibilité, au moins de manière approchée. Cette simple recommandation permet dans tous les cas d'obtenir des discrétisations dont le taux de convergence est optimal.

Title: Contribution to the improvement of meshless methods applied to continuum mechanics
Keywords: Meshless; Simulation; Continuum Mechanics; SPH
Abstracts: This thesis introduces a general framework for the study of nodal meshless discretization schemes. Its fundamental objects are the discrete operators defined on a point cloud : volume and boundary integration, discrete gradient and reconstruction operator. These definitions endow the point cloud with a weaker structure than that defined by a mesh, but share several fundamental concepts with it, the most important of them being integration-differentiation compatibility. Along with linear consistency of the discrete gradient, this discrete analogue of Stokes's formula is a necessary condition to the linear consistency of weakly discretized elliptic operators. Its satisfaction, at least in an approximate fashion, yields optimally convergent discretizations. However, building compatible discrete operators is so difficult that we conjecture – without managing to prove it – that it either requires to solve a global linear system, or to build a mesh. We dub this conjecture the "meshless curse". Three main approaches for the construction of discrete meshless operators are studied. Firstly, we propose a correction method seeking the closest compatible gradient – in the least squares sense – that does not hurt linear consistency. In the special case of MLS gradients, we show that the corrected gradient is globally optimal. Secondly, we adapt the SFEM approach to our meshless framework and notice that it defines first order consistent compatible operators. We propose a discrete integration method exploiting the topological relation between cells and faces of a mesh preserving these characteristics. Thirdly, we show that it is possible to generate each of the meshless operators from a nodal discrete volume integration formula. This is made possible with the exploitation of the functional dependency of nodal volume weights with respect to node positions, the continuous underlying space and the total number of nodes. Consistency of the operators is characterized in terms of the initial volume weights, effectively constituting guidelines for the design of proper integration formulae. In this framework, we re-interpret the classical stabilization methods of the SPH community as actually seeking to cancel the error on the discrete version of Stokes's formula. The example of SFEM operators has a volume-based equivalent, and so does its discrete mesh-based integration. Actually computing it requires a very precise description of the geometry of cells of the mesh, in particular in the case where its faces are not planar. We thus fully characterize the shape of such cells, only as a function of nodes of the mesh and topological relations between cells, allowing unambiguous definition of their volumes and centroids. Finally, we describe meshless discretization schemes of elliptic partial differential equations. We propose several alternatives for the treatment of boundary conditions with the concern of imposing as few constraints on nodes of the point cloud as possible. We give numerical results confirming the crucial importance of verifying the compatibility conditions, at least in an approximate fashion. This simple guideline systematically allows the recovery of optimal convergence rates of the studied discretizations.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France