Three-dimensional modeling and experiment-driven numerical simulation of zebrafish escape swimming for biological applications

Guillaume Ravel

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE

MATHÉMATIQUES APPLIQUÉES ET CALCUL SCIENTIFIQUE

Par **Guillaume RAVEL**

Three-dimensional modeling and experiment-driven numerical simulation of zebrafish escape swimming for biological applications

---

Under the supervision of:
Afaf BOUHARGUANE, Associate professor, University of Bordeaux
Patrick J. BABIN, Professor, University of Bordeaux

Defended on December 17, 2020

Committee members:

Mr. Angelo IOLLO	Professor	University of Bordeaux	President
Mr. Johan L VAN LEEUWEN	Professor	Wageningen University	Referee
Mr. Rajat MITTAL	Professor	Johns Hopkins University	Referee
Ms. Astrid DECOENE	Associate professor	Paris-Sud University	Examiner
Ms. Afaf BOUHARGUANE	Associate professor	University of Bordeaux	Supervisor
Mr. Patrick J. BABIN	Professor	University of Bordeaux	Supervisor
Mr. Michel BERGMANN	Research scientist	INRIA	Guest
Ms. Anja KNOLL-GELLIDA	Associate professor	University of Bordeaux	Guest

Three-dimensional modeling and experiment-driven numerical simulation of zebrafish escape swimming for biological applications

Abstract: Biologists use zebrafish as an animal model to study the effects of genetic or environmental factors related to human locomotor diseases in order to develop pharmacological treatments. The general objectives of the project were 1) to develop a numerical model based on real-world data capable of accurately simulating the escape swimming of the zebrafish eleuthero-embryo and 2) to provide, in addition to swimming kinematic parameters, a fine estimate of the energetic performance of locomotor behavior to enrich experimental studies on locomotion. Furthermore, experiment-based numerical modeling would enhance the understanding of locomotor behavior. For this purpose, a computational fluid dynamics code describing the fluid flow around a moving and deforming immersed body was used to reproduce *in silico* the experimental escape response of a five-day post-fertilization eleuthero-embryo. The solution of the mechanistic model, governed by the incompressible Navier-Stokes equations and Newton's laws was approximated on a Cartesian mesh while the solid body represented by a level-set function, was described implicitly by a penalization method. As for the deformation kinematics, it was estimated directly from experimental locomotion videos by a *Procrustes analysis*. A first approach based on optimal transportation has been considered to extract the deformation velocity in two dimensions. To be faithful to three-dimensional physics, the morphology of the zebrafish eleuthero-embryo and the experimental escape kinematics were reconstructed in three dimensions, by tracking Lagrangian markers on the surface of the zebrafish body. Thus, a new approach has been developed to estimate the deformation velocity from experimental real data obtained by ultra-high-speed imaging after electric field pulse stimulation. Zebrafish eleuthero-embryo exhibits a highly stereotyped and complex escape behavior consisting of three swimming modules: C-bend, counter-bend, and fast-swimming cyclic phase. The developed approach enables high-performance and realistic numerical simulations of real locomotion. After performing a numerical validation of the model based on each component, a study was conducted on the energetic performance of the zebrafish's escape response, challenged by a change in fluid viscosity. A linear response of the cost of transport, associated with constant energy expenditure, regardless of the fluid environment, was thus demonstrated. This energy study can be extended to any immersed, moving, and deformable body, and in particular, to any biological experiment such as exposure to a neuro-toxicant, which would alter the locomotor behavior of the eleuthero-embryo. Thus, numerical simulations may enrich the quantitative assessments of biological conditions and pharmacological treatments which lead to disturbing or recovering the locomotor behavior.

Keywords: 3D modeling, computational fluid dynamics, imaging, bio-locomotion, zebrafish, swimming performance.

Modélisation tridimensionnelle et simulation numérique basée sur une approche expérimentale de la nage de fuite du poisson-zèbre pour des applications biologiques

Résumé : Les biologistes utilisent le poisson-zèbre comme modèle animal pour étudier les effets des facteurs génétiques ou environnementaux liés aux maladies locomotrices humaines afin de développer des traitements pharmacologiques. Les objectifs généraux du projet étaient 1) de développer un modèle numérique basé sur des données réelles capable de simuler avec précision la nage de fuite de l'eleuthéro-embryon de poisson-zèbre et 2) de fournir, en plus des paramètres cinématiques de nage, une estimation fine des performances énergétiques du comportement locomoteur pour enrichir les études expérimentales sur la locomotion. En outre, une modélisation numérique basée sur l'expérience pourrait améliorer la compréhension du comportement locomoteur. Pour cela, un code de calcul de dynamique des fluides décrivant l'écoulement des fluides autour d'un corps immergé, mobile et déformable a été utilisé pour reproduire *in silico* la réponse de fuite expérimentale d'un éleuthéro-embryon de cinq jours post-fécondation. La solution du modèle mécaniste, régie par les équations de Navier-Stokes incompressible et les lois de Newton, a été approchée sur un maillage cartésien tandis que le corps solide, représenté par une fonction *level-set*, a été décrit implicitement par une méthode de pénalisation. Quant à la cinématique de déformation, elle a été estimée directement à partir de vidéos expérimentales de locomotion par une *Procrustes analysis*. Une première approche a été envisagée pour en extraire la vitesse de déformation, en deux dimensions, basée sur le transport optimal. Afin d'être fidèle à la physique tridimensionnelle, la morphologie de l'eleuthéro-embryon de poisson-zèbre et la cinématique de fuite expérimentale ont été reconstruites en trois dimensions, par le suivi de marqueurs lagrangiens à la surface du corps du poisson-zèbre. Ainsi, une nouvelle approche a été développée pour estimer la vitesse de déformation à partir de données réelles expérimentales obtenues par imagerie ultra-rapide après stimulation par impulsion de champ électrique. L'eleuthéro-embryon de poisson-zèbre présente un comportement de fuite très stéréotypé et complexe, composé de trois modules de nage : courbure en C, contre-courbure et phase de nage cyclique rapide. L'approche développée permet de réaliser des simulations numériques haute-performance et réalistes de la locomotion réelle. Après avoir effectué une validation numérique du modèle qui repose sur chacune de ses composantes, une étude a été menée sur la performance énergétique de la réponse de fuite du poisson-zèbre, altérée par un changement de la viscosité du fluide. Une réponse linéaire du coût du transport, associée à une dépense d'énergie constante, indépendamment du milieu, a ainsi été montrée. Cette étude énergétique peut être étendue à tout corps immergé, en mouvement et déformable, et en particulier à toute expérience biologique comme l'exposition à un neuro-toxique, qui altérerait le comportement locomoteur de l'éleuthéro-embryon. Ainsi, la simulation numérique peut enrichir l'évaluation quantitative des conditions biologiques et des traitements pharmacologiques qui conduisent à perturber ou à restaurer le comportement locomoteur.

Mots-clés : Modélisation 3D, CFD, imagerie, bio-locomotion, poisson-zèbre, performance de nage.

Institut de Mathématiques de Bordeaux
UMR 5251, University of Bordeaux, F-33400 Talence, France.
&
Laboratoire Maladies Rares : Génétique et Métabolisme
INSERM U1211, University of Bordeaux, F-33615 Pessac, France.

« Nature is the source of all true knowledge. She has her own logic, her own laws, she has no effect without cause nor invention without necessity. »

– Leonardo Da Vinci

À Ulysse,

et À PJH,

Remerciements - Acknowledgements

En premier lieu, j'aimerais adresser de profonds remerciements à mes deux directeurs de thèse qui ont co-supervisé ce travail : Afaf Bouharguane pour l'approche numérique et Patrick Babin pour la direction expérimentale et sa vision pour la biologie. Je vous remercie notamment de m'avoir fait confiance pour ce projet autant d'un point de vue scientifique que sur le plan humain où j'ai appris à mener cette collaboration interdisciplinaire, avec vous. En particulier, je retiendrai l'exigence scientifique de Patrick (ces longues discussions n'ont sûrement pas été vaines !), et l'écoute de Afaf - notamment pour ta présence et tes encouragements, si importants face à mes doutes et mes questionnements ! Je peux dire que je n'aurais pas accompli ce travail de longue haleine sans vous. J'ai été ravi de ces trois années avec vous, tant sur le plan scientifique qu'humain. J'ai vraiment apprécié la patience dont vous avez fait preuve à mon égard, et cette complémentarité nécessaire pour diriger cette thèse sur les deux fronts scientifiques. Dans ce travail, j'ai essayé de faire honneur à la fois au champ de la biologie qui fût nouveau pour moi, et au foisonnement des mathématiques appliquées qui m'est si cher. J'espère avoir été à la hauteur de vos attentes, malgré le manque de valorisation scientifique notamment dû à cette dernière année rocambolesque.

Ma seconde pensée va tout droit vers les trois autres personnes qui ont accompagné ce projet pendant ces trois années, soit Anja Knoll-Gellida, Michel Bergmann et Angelo Iollo. Au gré des pérégrinations du projet, vous avez de près ou de loin, grandement contribué à cet aboutissement.

Anja, tu m'as guidé dans la partie expérimentale. Je te remercie de m'avoir rappelé que les statistiques sont parfois utiles en biologie et que l'expérimentation animale est tout un art, que dis-je, toute une science ! A tes côtés, j'ai appris à travailler avec des poisson-zèbres, avec assurance et patience.

Je souhaite ensuite exprimer toute ma reconnaissance à Michel, qui a toujours laissé la porte de son bureau ouverte. Il faut savoir que ce travail de thèse a reposé en grande partie sur ton travail , ton code de calcul *NaSCar* et ton expertise. Développer un code de calcul est peut-être un travail de l'ombre, mais je crois que nos échanges ont été très enrichissants, tant scientifiquement qu'humainement. J'ai beaucoup apprécié de travailler avec toi et je t'en remercie, sincèrement.

Finalement, je tiens exprimer ma profonde gratitude à Angelo, pour avoir gardé un regard - même éloigné, sur ce projet. Ce recul a permis d'éclairer certains points sous un autre angle, à des moments clés de cette thèse. Je te remercie notamment pour avoir crû à mon travail, et surtout pour avoir pris du temps pour le soutenir lorsque nécessaire.

En outre, merci à tous les trois d'avoir fait partie de mon jury de thèse et particulièrement à Angelo, pour avoir accepté de présider ce jury de thèse.

To that point, I would like to express my special thanks to Pr Rajat Mittal and Pr Johan L van Leeuwen. I was very honored to have you for reviewing my Ph.D. thesis and participating in my thesis defense committee. Your feedback was really important to me, especially from a scientific point of view. Similarly, I would like to offer my thanks to Ms. Astrid Decoene for accepting to be in my thesis defense committee.

Besides, I would like to express special thanks to the *NeuroData* group which provided access to the high-resolution massive database located at http://neurodata.io/data/hildebrand16 and http://zebrafish.link/hildebrand16, on which we the 3D reconstruction of the zebrafish morphology. Note the data have been generated by David Grant Colburn Hildebrand (davidh@fas.harvard.edu) [52].

All the numerical simulations which have been performed during these Ph.D. years and the numerical results presented in this Ph.D. thesis, were carried out using two computer clusters: the PlaFRIM experimental parallel testbed, supported by INRIA, LABRI, IMB, University of Bordeaux, Bordeaux INP, Conseil Régional d'Aquitaine, CNRS (see https://www.plafrim.fr) and the MCIA experimental testbed, CURTA, supported by the Pôle de Recherche et Enseignement Supérieur (PRES) of the University of Bordeaux and the University of Pau et des Pays de l'Adour (UPPA) (see https://calcul.math.cnrs.fr).

Tout au long de ces trois années de thèse, de nombreuses personnes m'ont accompagnées. En premier lieu, je souhaite remercier chaleureusement Nicolas Papadakis et Daniel Cattaert pour avoir fait parti de mon comité de thèse, pour votre suivi et vos précieux conseils. Par ailleurs, mille mercis à Anne-Laure Gautier, l'assistante d'équipe INRIA MEMPHIS, Sylvia Branchu et Karine Lecommandoux les gestionnaires de l'équipe INSERM MRGM, ainsi qu'au personnel administratif et informatique de l'IMB, pour votre support chaleureux et votre disponibilité. Également, je tiens aussi à remercier l'Ecole Doctorale Mathématiques et Informatique et sa gestionnaire Sylvaine Granier, pour son efficacité et son appui aux doctorants lors des missions.
Ensuite, je tiens à remercier tous mes collègues qui m'ont entourés pendant ces années de doctorat. J'ai eu la chance de côtoyer deux équipes de recherche, et même un peu plus. Certes, cela m'a demandé de partager mon temps sur deux sites si proches, si loin, et si différents, en termes de vie de laboratoire par exemple. Mais finalement, cela n'est qu'une question d'organisation pour allier rythme de travail adapté et présence à vos côtés. Je crois profondément qu'une présence sera toujours nécessaire pour entretenir une vie de laboratoire et d'équipe saine et dynamique, si importante à tout le monde. Cela se remarque d'autant plus lorsque nous en sommes privé.
D'abord, je souhaite remercier l'ensemble de l'équipe du MRGM et plus précisément

l'ensemble des membres du bâtiment B2. En particulier, je pense à Laure, Magalie (que deviendraient les poissons sans vous ?) et Leslie, bien sûr (ma co-bureau de la première heure) - qui a constitué un sacré trio depuis mon arrivée - mais aussi à Sara, Théo (bonne chance pour la suite) et Sandra. Merci les filles pour tous ces petits moments et votre énergie, vous avez définitivement été là pour la vie de cette équipe, et scientifiquement pour m'apprendre le poisson-zèbre ! Malgré la sensation de ne parfois pas parler le même langage, j'ai bien fini par "maniper" à vos côtés.
Ensuite, j'aimerais remercier l'ensemble de l'équipe Inria MEMPHIS pour cette aventure. Il se trouve que le relationnel, et nos moments d'équipe ont eux aussi dû se mettre au virtuel lors de cette dernière année, mais vous avez été là. En particulier, Sébastien et Luis avec qui j'ai partagé ces trois ans de thèse, de plus ou moins près. Également, je remercie tous les membres du Calcul scientifique du 3 ème étage de l'IMB, que j'ai pu côtoyer pendant ces trois années. Je tiens à adresser un remerciement particulier à Annabelle Collin, pour m'avoir suivi depuis mes années à l'ENSEIRB-MATMECA, m'avoir inspiré et encouragé à continuer vers le doctorat. Et puis à tous ceux passés par "MonPhis" (ou "Sik a koko"), ce groupe essentiellement franco-italien regroupant doctorants, post-doctorants, voire stagiaires, start-up et ingénieurs. Je ne vais pas pouvoir citer tout le monde d'une manière exhaustive, mais je retiendrai les fidèles du Haut-Carré (vous avez vu l'heure ?), et ceux des pauses cafés conviviales ainsi que ceux en vadrouille dans les bureaux ou dans le couloir (je ne citerai pas de nom ici-même), sans oublier nos sorties culinaires et culturelles (enfin, ça c'était avant cette dernière année). Bref, vous vous reconnaitrez; ce groupe est définitivement plein de ressources et cela a été chouette de partager nos moments de joies et galères du quotidien. Tout de même, petite dédicace à tous mes co-bureaux au fil des rotations : Stefano, Costi, Christèle, Giuliano, et bien sûr à Antoine, pour tous ces instants privilégiés à échanger de science et de tout, sans regarder l'heure. Merci à tous pour tous ces moments de vie, si enrichissants en période de thèse ! J'espère sincèrement que l'on pourra célébrer cette thèse d'une manière ou d'une autre pour pallier l'absence de pot de soutenance.

Enfin, je souhaite à tout le monde de côtoyer la colocation juste pour les rencontres improbables qu'elle permet. De la colocation aux amitiés naissantes. Mille mercis à vous tous. Fan - bien sûr, ainsi que les colocs de la première heure : Jerem, Coco, Pierre, Popo, Maïna, Aude et Natasha (super année de coloc' mine de rien), sans oublier Gabriel, Maxime, Alexandre, Marine, Kamil, Katherine, Marion, Anna, Léo, Michael. Et enfin à Emma (merci à toi, pour la soupe et pour ces échanges passionnants), Amélie (merci pour ton impro et ton thé) et Quentin (merci pour ta présence), à vous deux pour ce super "confinement" et pour avoir supporté cette répétition de soutenance. De ces amitiés, aux réseaux de colocs étendus : Alex, Chacha, Anto, Chiara, et tout ce super groupe. Éclectique. Puisse le virtuel ne pas prendre le pas aux moments fugaces

partagés.
Une dernière pensée à tous ceux qui m'ont suivi et soutenu pendant mes études, du lycée à l'Ecole d'Ingénieur, sans oublier les trois ans de Prépa mémorables, inénarrables. Petite dédicace à Monsieur Pierre-Jean Hormière, mon guide dans les mathématiques et un exemple de transmission.

Évidemment, je ne peux finir ces remerciements sans citer ma famille qui malgré la distance, je le sais, a toujours été là. Merci à mes parents et mes deux sœurettes donc, pour leur soutien inconditionnel et sans failles. Ce travail de thèse représente certes un aboutissement d'un long chemin de plusieurs années d'études. Mais soyez sûrs d'une chose, je n'oublierai ni d'où je viens, ni cette chance d'avoir pu aller là où je suis allé. Fort heureusement, j'aurai aussi une forte pensée pour l'ensemble de mes cousins et cousines et notamment, à Ulysse.

Contents

Résumé détaillé

Modélisation tridimensionnelle et simulation numérique basée sur une approche expérimentale de la nage de fuite du poisson-zèbre pour des applications biologiques

Contexte de la thèse

Cette thèse s'inscrit dans un cadre largement interdisciplinaire, étant le fruit d'une étroite collaboration entre une équipe de biologistes et une équipe de modélisation multi-physiques. L'objet de ce travail de thèse concerne la modélisation d'un phénomène biologique : la nage de fuite d'un éleuthero-embryo de poisson-zèbre. Le poisson-zèbre est étudié par les biologistes en tant que modèle animal pour un grand nombre de raisons. D'abord, il s'agit d'un animal qui possède un patrimoine génétique très proche de celui de l'homme. Cela permet d'amener les conclusions des expérimentations animales à l'homme. Ensuite, le poisson-zèbre est un animal d'une très haute fertilité qui possède un développement externe. Cela permet d'expérimenter dès les premiers stades de développement sur un très grand nombre d'animaux simultanément. De plus, le poisson-zèbre est très transparent ce qui est important pour visualiser certains processus biologiques à l'aide de marqueurs fluorescents par exemple. Par ailleurs, comme il s'agit d'un animal aquatique avec une très petite taille de quelques millimètres dans les premiers stades de développement, les conditions d'élevage sont particulièrement favorables à l'expérimentation animale. Enfin, dans le cadre de l'étude de la locomotion, le poisson-zèbre montre un mouvement de fuite qui est hautement stéréotypé, constitué d'une courbure en C (*C-bend*), d'un mouvement de contre-courbure (*counter-bend*) et de cycles de nage rapide. Le caractère reproductible de cette réaction de fuite est très important pour analyser sur un grand nombre d'animaux, l'impact de molécules neurotoxiques sur les processus biologiques impliqués dans la locomotion. Par la suite, une quantification des performances de nages après un démarrage en C (*C-start*) est nécessaire, afin de déterminer toute altération de cette réaction de fuite.

Pour l'étude de la locomotion du poisson-zèbre, le suivi des trajectoires est généralement utilisé pour calculer la distance parcourue voire éventuellement la

vitesse moyenne ou bien l'amplitude de la courbure du corps au cours de la nage. Les algorithmes de suivi (*tracking*) des poissons-zèbres les plus élaborés permettent de suivre automatiquement le déplacement de l'ensemble de la géométrie en deux ou trois dimensions, grâce notamment à une modélisation de la morphologie basée sur des splines ou ellipses par exemple. Quant à l'écoulement du fluide généré dans le sillage du poisson, il peut être analysé *via* la vélocimétrie par imagerie de particules (*PIV*). Outre les quantités basées sur la cinématique de nage, des études se sont employées à retrouver des quantités comme la quantité de mouvement, la puissance moyenne *via* le calcul de moments d'inertie et de rotation du corps et des méthodes de dynamique inverse. Remonter jusqu'aux données énergétiques est particulièrement intéressant pour les biologistes pour comprendre l'ensemble de la dynamique de la nage. C'est pourquoi les méthodes de calcul de dynamique des fluides (*CFD*) sont particulièrement attrayantes pour calculer directement à partir d'équations et donc entièrement *in silico*, l'écoulement fluide, les forces hydrodynamiques exercées sur le poisson-zèbre et ainsi, la puissance instantanée, l'énergie dépensée par le poisson-zèbre pour se mouvoir et l'efficacité de propulsion. Alors, les méthodes CFD permettent de modéliser l'ensemble de la nage et de fournir des données jusqu'alors inaccessibles sur la dynamique et l'énergétique de la nage. Dans la littérature, on peut trouver de nombreuses méthodes dédiées à la modélisation et la simulation de locomotions aquatiques, notamment développées pour calculer les performances de nage et alors, optimiser la géométrie du nageur ou bien sa loi de nage, notamment pour la robotique sous-marine ou bien l'ingénierie. Cependant, très peu d'études se sont appuyées directement sur des géométries ou lois de nages réelles afin de présenter des simulations numériques parfaitement réalistes. Ce travail de thèse avait pour objectif de développer une approche numérique basée sur du calcul CFD capable de reproduire *in silico* des locomotions expérimentales qui permettrait de fournir aux biologistes des informations notamment d'un point de vue énergétique sur les performances de nage d'un poisson-zèbre après cinq jours de développement, soit le stade eleuthero-embryo. Ces données obtenues par combinaison d'imagerie expérimentales et simulations numériques ont pour objectif d'étayer et préciser les effets de certaines molécules toxicologiques ou pharmacologique sur la locomotion, étudiées grâce à l'expérimentation massive sur le modèle poisson-zèbre. En définitive, on se propose dans ce travail, de développer une approche numérique CFD directement basée sur l'expérience, destinée à enrichir les expérimentations animales. Les challenges, nombreux, concernent d'une part, le développement numérique intégrant à la fois une morphologie réaliste du poisson-zèbre et les déformations réelles observées par imagerie expérimentale. D'autre part, cet outil numérique doit permettre d'aider à l'évaluation d'effets biologiques dus à une altération du milieu soit par la viscosité du fluide soit par la présence de particules neurotoxiques.

Cadre des simulations numériques

La simulation de nageurs auto-propulsés n'est pas simple à résoudre car cela engage une forte interaction entre le fluide et un objet solide possédant une géométrie complexe, mobile et très déformable. Les vitesses fluide et solide sont gouvernées par leurs propres équations : Navier-Stokes incompressible et Newton, respectivement. Cependant, la condition de bord, à l'interface, doit être imposée à chaque pas de temps. Des méthodes dites "body-fitted", qui maillent la surface du solide, peuvent s'avérer très coûteuses pour mettre à jour le maillage. Au contraire, les méthodes sur maillage Cartésien sont particulièrement adaptées pour suivre les mouvements du poisson sans avoir à remailler le domaine de calcul au cours du temps, et sont facilement parallélisables. C'est pourquoi, nous avons utilisé le code *NaSCar* développé au sein de l'équipe Inria MEMPHIS, combinant principalement maillage Cartésien et différences finies, représentation implicite du solide *via* une fonction *Level-set* et une méthode de frontière immergée par pénalisation. L'écoulement du fluide est alors modélisé par les équations de Navier-Stokes incompressible pénalisées :

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\Delta \mathbf{u} + \chi_s \lambda(\mathbf{u_b} - \mathbf{u}) \text{ in } \Omega \tag{2}$$

Dans ce système régissant la vitesse du fluide $\mathbf{u}$ et la pression p dans le domaine de calcul Ω, μ et ρ désignent la viscosité dynamique et la densité du fluide tandis que $\mathbf{u_b}$, χ_s, λ représentent la vitesse du solide, la fonction caractéristique associée au solide issue du calcul de la *Level-set* soit en chaque point la distance signée à l'interface et le facteur de pénalisation associé au terme de pénalisation $\chi_s\lambda(\mathbf{u_b}-\mathbf{u})$. Pour λ suffisamment grand, on impose alors une condition de non-glissement à l'interface. De plus, les vitesses proches de l'interface ont aussi été corrigées *via* la méthode IPC afin d'obtenir un ordre deux de précision en espace. Ces équations sont résolues par une méthode à pas fractionnaire de Chorin et Temam impliquant des étapes de prédiction, projection et correction. Quant à la vitesse du poisson $\mathbf{u_b}$, on l'a modélisée comme la somme d'une partie rigide $\mathbf{u_{rigid}}$ et d'une contribution déformable $\mathbf{u}_\epsilon$:

$$\mathbf{u_b} = \mathbf{u_{rigid}} + \mathbf{u}_\epsilon = \bar{\mathbf{u}} + \mathbf{u}^\theta + \mathbf{u}_\epsilon \tag{3}$$

La vitesse de translation $\bar{\mathbf{u}}$ et la vitesse associée à la rotation $\mathbf{u}^\theta$ peuvent alors être calculées par les lois de la mécanique classique :

$$\rho_s V_s \frac{\mathrm{d}\bar{\mathbf{u}}}{\mathrm{d}t} = \mathcal{F}^{\mathbf{hydro}} \tag{4}$$

$$\frac{\mathrm{d}J\omega}{\mathrm{d}t} = \mathcal{M}^{\mathbf{hydro}} \tag{5}$$

grâce au tenseur hydrodynamique $\mathcal{F}^{\mathbf{hydro}}$,$\mathcal{M}^{\mathbf{hydro}}$, J, ρ_s, V_s désignant respectivement le moment d'inertie, la densité de masse et le volume du solide. Le terme restant de déformation $\mathbf{u}_\epsilon$ est clé pour la modélisation de la nage. Dans la littérature, ce terme est généralement imposé par une formulation mathématique, comme une onde sinusoïdale par exemple. L'objet de ce travail est d'extraire ces informations manquantes cruciales que sont la déformation du poisson $\mathbf{u}_\epsilon$ et sa géométrie χ_s grâce aux données d'imagerie expérimentale. Pour cela, deux solutions ont été mises en œuvre pour interpoler $\mathbf{u}_\epsilon$ et χ_s à chaque pas de temps du solveur. La première approche à consister à extraire un champ de vitesse de déformation à partir des images de la nage en vue de dessus soit en deux dimensions. Dans ce cas, la géométrie déformée à chaque instant nous vient directement de l'imagerie. Pour des simulations en trois dimensions, des reconstructions tridimensionnelles de chaque déformée sont nécessaire pour calculer le champs de vitesse de déformation entre chaque pas de temps d'acquisition. Pour ce faire, nous avons utilisé la théorie du transport optimal qui nous permet d'interpoler le déplacement du poisson entre chaque image expérimentale. Quant à la seconde approche, elle avait pour but de suivre la vitesse à l'interface de manière Lagrangienne grâce à des marqueurs Lagrangiens reconstruits à l'interface. La vitesse de pénalisation est alors entièrement définie par les vitesses Lagrangiennes des marqueurs. Cette méthode a nécessité de reconstruire des marqueurs Lagrangiens sur toutes les déformées expérimentales du poisson. Pour cela, nous avons d'abord généré un premier ensemble de marqueurs définissant la forme du poisson-zèbre. Ensuite, nous avons suivi ces marqueurs au cours du temps en s'appuyant sur les déformations de la ligne médiane (*midline*) au cours du temps, elles-mêmes extraites des images expérimentales. Dans tous les cas, les images expérimentales en deux dimensions ont été utilisées, après un travail de traitement d'image consistant principalement à recentrer chaque déformée dans son propre référentiel (étape de *Procrustes analysis*).

Démarche adoptée

Les expériences de locomotion ont consisté à filmer le mouvement de fuite d'un poisson-zèbre grâce à une caméra ultra-rapide (jusqu'à 15 000 fps) dont l'équipe du MRGM est équipée, placée au-dessus d'une cellule expérimentale unique au MRGM, qui peut fournir des images de très hautes résolutions. La réaction de fuite est initiée par une impulsion électrique (*Electric Field Pulse*) permettant de reproduire de manière consistante le phénotype caractéristique. Dans le cas d'une très faible hauteur d'eau dans la cellule expérimentale, le poisson-zèbre dégage une nage de fuite quasiment plane. En effet, on ne peut, à ce stade, modéliser que des nages planes étant donné le dispositif actuel pour suivre les mouvements du poisson. L'étape suivante naturelle a été le traitement d'image et notamment la segmentation du poisson-zèbre. Cela a pu se révéler fastidieux étant donné que certaines parties du poisson-zèbre et notamment la fine membrane

constituant la queue (*median fin fold*) sont transparentes. C'est pourquoi, nous avons combiné méthodes de seuillage et de classification (*clustering*). L'étape de *Procrustes analysis* était ensuite nécessaire pour obtenir la déformation du poisson-zèbre, seule autrement dit, sans les mouvements rigides de translation et de rotation. L'algorithme que nous avons utilisé était principalement basé sur la conservation de la quantité de mouvement. Seulement, de nombreuses corrections ont dû être apportées dans le cas des grandes déformations. Enfin, il est important de rappeler qu'actuellement, seules deux translations et une rotation peuvent être calculées à partir de l'imagerie 2D. Bien que des simulations en deux dimensions ont d'abord été envisagées, notamment en développant la première approche avec un algorithme de transport optimal, il nous a fallu rapidement considérer des simulations en trois dimensions afin d'avoir une modélisation de la nage plus réaliste. En deux dimensions, la forme du poisson-zèbre était directement issue des silhouettes segmentées alors qu'en trois dimensions, le volume 3D du poisson-zèbre a dû être reconstruit.

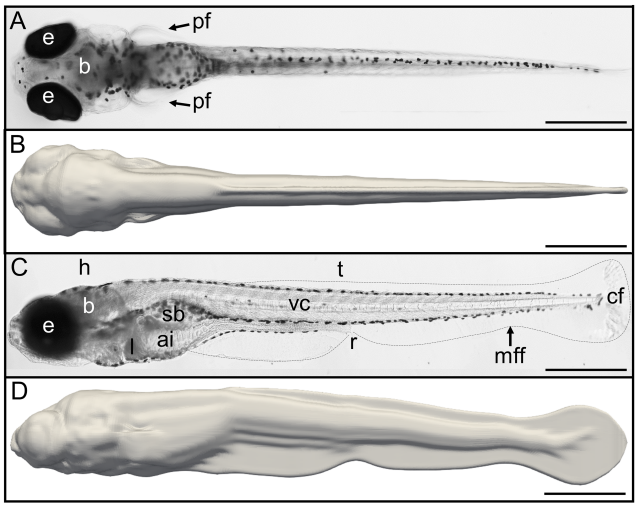


FIGURE 1: Comparaison des vues de profil et dorsale entre un eleuthero-embryo de poisson-zèbre de 5 dpf (Figs. (A),(C)) et la reconstruction 3D de la surface (Figs. (B),(D)). Nombre de marqueurs Lagrangiens sur la surface : 300 coupes transverses, 180 marqueurs par coupe. Abbréviations : *median fin fold* (mff), intestin antérieur (ai), cerveau (b), nageoire caudale (cf), yeux (e), tête (h), foie (l), nageoires pectorales (pf), rectum (r), vessie natatoire (sb), tronc (t), colonne vertébrale (vc). Échelle : 500 μm.

Pour cela, une multitude de coupes transverses issues d'une base de données a été utilisée pour reconstituer la morphologie complexe du poisson-zèbre, de la tête à la fin des nageoires pectorales. Pour la partie la plus postérieure, quelques coupes clés ont été générées puis une série de coupes transverses uniformément réparties sur l'ensemble du poisson-zèbre, a été constituée par interpolation grâce à une autre application du transport optimal pour l'interpolation de forme. Finalement, des marqueurs Lagrangiens ont été uniformément distribués à la surface de cette reconstruction 3D. Le résultat obtenu (Fig. 1) est très satisfaisant puisqu'on obtient une géométrie précise et très fidèle à la morphologie du poisson-zèbre. En particulier, notons que les nageoires pectorales ont été conservées, étant fixes pendant la nage de fuite, et la membrane a été considérée

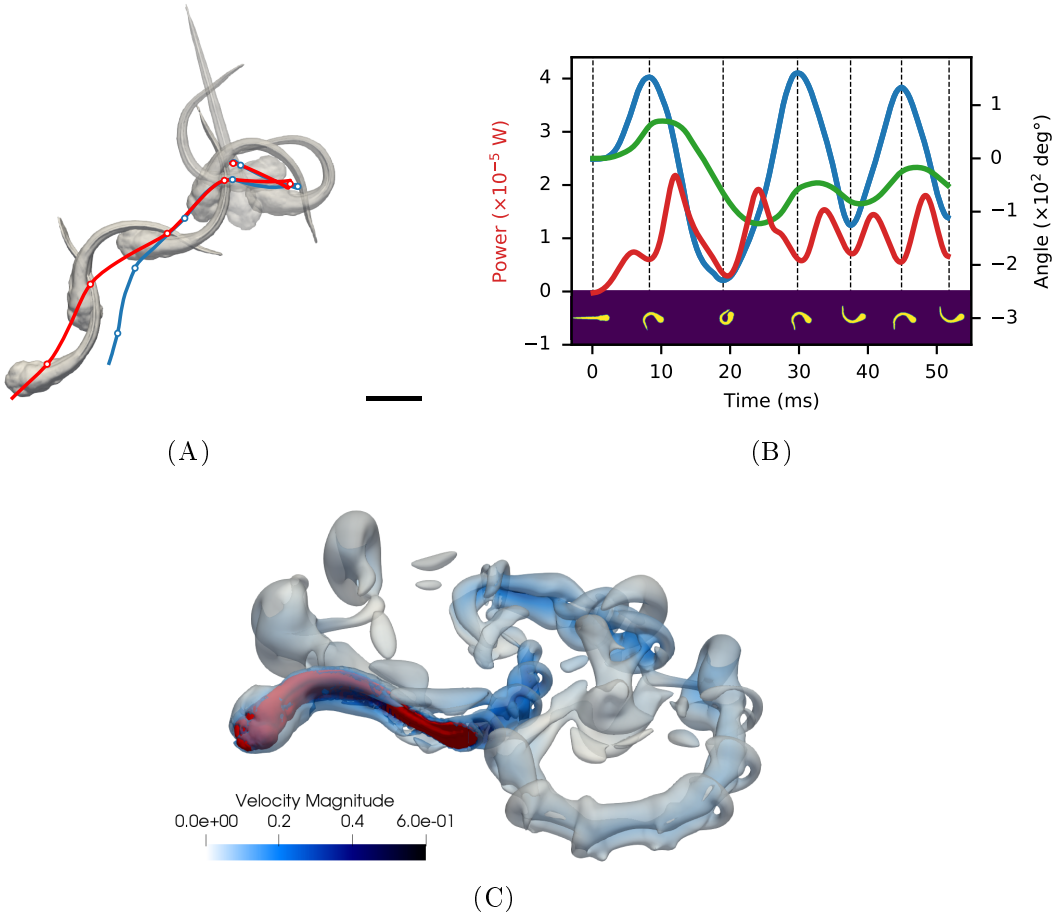
comme rigide et de même densité volumique que le reste du corps. Par la suite, cette reconstruction 3D a donc été déformée suivant les mouvements de la ligne médiane, en considérant que chaque coupe transverse reste orthogonale à la ligne médiane.

Principaux résultats

Avant de mettre en œuvre l'approche développée de manière massive sur des applications pour la biologie, nous avons d'abord réalisé de nombreuses validations des simulations numériques 3D de la nage sur des premiers cas tests. En effet, le code *NaSCar* que nous utilisons a déjà été largement validé dans le cadre d'une loi de nage classique. Toute l'originalité de cette nouvelle approche repose sur l'intégration des déformations expérimentales en trois dimensions, au sein du solveur fluide-structure. Par ailleurs, les snapshots déformés 3D ont été obtenus après un certain nombre d'étapes : traitement d'image, *Procrustes analysis*, extraction de ligne médiane, génération et reconstruction de marqueurs Lagrangiens. C'est pourquoi, diverses étapes de validation ont été nécessaires afin d'analyser la capacité de notre modèle à reproduire la trajectoire expérimentale de la locomotion, ainsi que la précision et la sensibilité de la solution.

Validation

La plupart des validations ont été réalisées sur un premier cas test où la déformation pendant la courbure en C est particulièrement importante. Après étude de la convergence, le pas d'espace a été fixé pour toutes les simulations. On peut voir que la trajectoire du centre de masse de la solution (Fig. 2 A) est bien très bien reproduite. En particulier, la partie la plus complexe du mouvement pendant le démarrage en C correspond extrêmement bien. Néanmoins, la trajectoire semble diverger à partir du début de la nage rapide. En fait, le poisson-zèbre se met à accélérer par rapport à la nage expérimentale. Plusieurs pistes ont été envisagées afin de comprendre la source de cette déviation. Par exemple, l'impact de la taille du domaine de calcul et des conditions aux bords a été étudiée. Des simulations bi-fluides ont aussi été réalisées pour modéliser la surface de l'eau comme une surface libre. Par ailleurs, l'influence d'autres paramètres comme la taille du poisson-zèbre, la viscosité du fluide ou bien la qualité de la *Procrustes Analysis* a été testée, malgré une rotation du corps plutôt en accord avec l'expérimental. En fait, la rotation n'est pas prépondérante dans la dernière phase de la réponse de fuite. En outre, divers facteurs comme la densité de masse pourraient affecter la position calculée du centre de masse. Cependant, aucune de ces simulations n'a été fructueuse. Trois pistes n'ont toutefois pas été levées. La première concerne l'hypothèse de mouvement plan et les deux autres points, la modélisation de la membrane transparente (*median fin fold*) qui joue probablement un rôle prépondérant dans la propulsion du poisson-zèbre. En effet, on a considéré dans ce travail, une



(A) (B)

(C)

FIGURE 2: Exemple de simulation 3D de la réponse de fuite du poisson-zèbre. **(A)** Trajectoires expérimentale et numérique du centre de masse. Les déformées 3D du poisson-zèbre sont indiquées en arrière-plan (Échelle : 1 mm). **(B)** Puissance dépensée, amplitude de courbure et rotation du corps. Les déformées correspondantes au mouvement sont indiquées par les images. **(C)** Illustration du sillage 3D généré par la nage de fuite du poisson-zèbre, à t = 46 ms. Q-criterion (Isocontour) coloré par la vitesse du fluide et poisson-zèbre représenté en rouge.

densité de masse homogène dans tout le volume 3D. Enfin, nous n'avons considéré aucune modélisation élastique, ce qui fait probablement défaut pour le *median fin fold*. Malgré cette légère accélération, une analyse d'ensemble de la solution a été menée, concernant l'écoulement du fluide (Fig. 2 C) ou bien l'allure de la puissance suivant la rotation ou la déformation (Fig. 2 B). Essentiellement, on peut remarquer que le poisson-zèbre produit un pic de puissance à chaque battement de queue, ou changement d'inflexion de la déformation. En outre, on peut voir un délai significatif entre la déformation et la rotation résultante. Le lien entre les différents pics de puissance et les différentes amplitudes de déformation resterait à examiner précisément afin d'améliorer la compréhension de la nage. Afin de tester la consistance de notre approche numérique basée sur l'expérience, nous l'avons ensuite appliquée à divers cas tests. Les résultats de la simulation numérique précédemment étudiée ont été retrouvés dans la plupart des cas, à un certain degré. Ainsi, toutes les trajectoires expérimentales ont été globalement très bien reproduites. Néanmoins, une accélération de la solution numérique a de nouveau

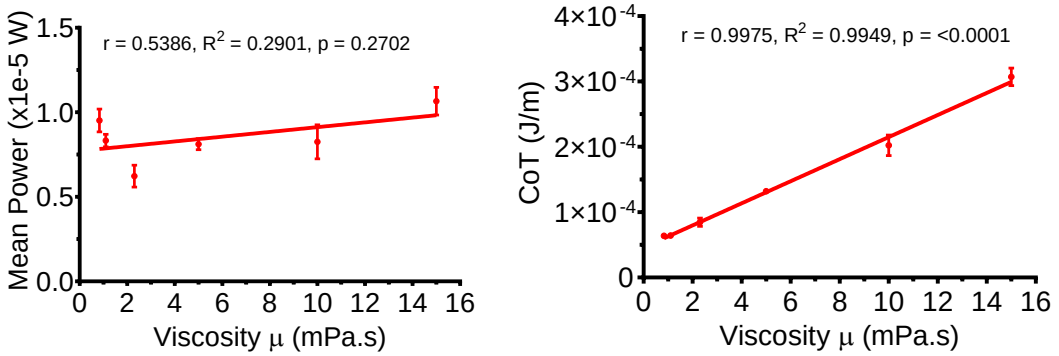
été perçue sur d'autres cas tests. Cela nous a amenés à considérer d'une part que la source de cette erreur provenait certainement d'un aspect de modélisation, plutôt que de questions numériques. D'autre part, cela démontre que nous retrouvons des résultats très consistants d'un cas test à l'autre. La reproductibilité de notre approche numérique nous a permis d'analyser les performances de nage pour des applications en biologie, dans la mesure où l'erreur numérique reste négligeable devant les effets biologiques à étudier.

Application à la Biologie

En termes d'application, deux objectifs étaient visés. D'abord, l'approche numérique avait pour but d'aider les biologistes à quantifier les performances de nage d'une réponse de fuite altérée, notamment par le calcul de performances énergétiques. Ainsi, la simulation numérique aidée de l'expérience, peut renseigner sur le coût de la locomotion pour atteindre une certaine distance ou bien fuir à une certaine vitesse par exemple. C'est pourquoi, cette approche peut parfaitement s'inscrire dans la mise en place d'un test d'effort. Cela consiste à mettre à l'épreuve la nage du poisson-zèbre afin d'en mesurer ses performances énergétiques. L'idée est ainsi, d'étudier les effets de certaines molécules à la lumière de l'impact énergétique.

La première application a naturellement consisté à calculer les performances de fuite du poisson-zèbre, lorsque soumis à une augmentation de la viscosité du milieu. Expérimentalement, un total de 115 réponses de fuite ont été filmées dans six milieux correspondants à différentes viscosités de fluide. Une base de données d'entre 10 et 15 films par condition de viscosité a donc été générée. Dans le cadre de cette première étude, la simulation numérique a été réalisée pour 3 réponses de fuite pour chacune des viscosités étudiées. A cela, un important résultat a été souligné par le calcul des performances énergétiques (Fig. 3) : la puissance dépensée moyenne semble rester stable sur la plage de viscosité étudiée. Seules des simulations basées sur des nages réelles peuvent nous apprendre une telle information, difficilement prévisible. En effet, cela implique que le poisson-zèbre s'est d'une certaine façon adapté à son milieu, probablement contraint par une quantité d'énergie limitée. D'autres simulations numériques basées sur des déformations fictives nous ont même laissées à penser que cette adaptation est significativement efficace. Par ailleurs, un second résultat remarquable concerne l'évolution du coût de transport en fonction de la viscosité (Fig. 3). Le coût de transport (CoT) définit comme le ratio de l'énergie dépensée au totale sur la distance parcourue lors de la fuite, est une quantité très pertinente pour mesurer une efficacité de la nage de fuite. Des simulations numériques réalistes ont permis de montrer que le coût de transport est très significativement corrélé à la viscosité du fluide. Cela est important d'une part en vue de la mise en place des test d'efforts et d'autre part, pour

quantifier de manière significative l'impact de fortes viscosités sur la réponse de fuite, que l'on ne peut mesurer avec la distance ou la vitesse, par exemple.



(A) (B)

FIGURE 3: Évolutions de variables calculées, associées à la dépense d'énergie, suivant la viscosité du fluide. Pour chacune des six conditions expérimentales de viscosité associées, trois simulations numériques ont été analysées statistiquement. **(A)** Puissance consommée en moyenne. **(B)** Coût total du transport : ratio de l'énergie sur la distance parcourue.

Une seconde application a ensuite été développée : l'étude des effets de composants neurotoxiques que l'on retrouve dans les pesticides ou dans les gaz de guerre par exemple, sur la nage de fuite du poisson-zèbre. Dans le cadre de cette étude, préliminaire aux applications biologiques réelles, notre méthode numérique n'a toutefois pas permis de discriminer, ni même de tirer de conclusion remarquable suite aux calculs des performances énergétiques. La très grande différence avec le cas précédent vient de la variabilité des effets biologiques étudiées, beaucoup plus faibles dans la plage de concentrations étudiées qu'avec la très grande plage de viscosités expérimentées.

Comme indiqué précédemment, nous avons aussi été en mesure de réaliser, grâce à la méthodologie développée, des simulations numériques fictives autrement dit, des simulations numériques où les déformations du corps du poisson-zèbre ne correspondent pas à celles observées expérimentalement. Ainsi, les performances de nages peuvent être comparées au vu des déformations utilisées. Cela a surtout permis de mettre en place un test d'effort numériquement (Fig. 4). En effet, dans ce travail, la viscosité a seulement été augmenté dans le modèle numérique. Cette première approche a permis de démontrer que l'on pouvait effectivement observer des différences significatives de performances entre deux réponses de fuites (associées à deux conditions expérimentales), lorsque les déformations associées sont immergées dans un fluide à très haute viscosité. La prolongation naturelle de cette étude serait évidemment de réaliser des tests d'efforts par l'expérimental en combinant molécules neurotoxiques et augmentation de la viscosité du fluide, pour comparaison.

Pour conclure, ce travail de thèse a d'abord consisté à développer une méthodologie permettant d'intégrer des données expérimentales : la déformation du poisson-zèbre ainsi que la forme de son corps à chaque instant, au sein d'un code de simulation numérique.

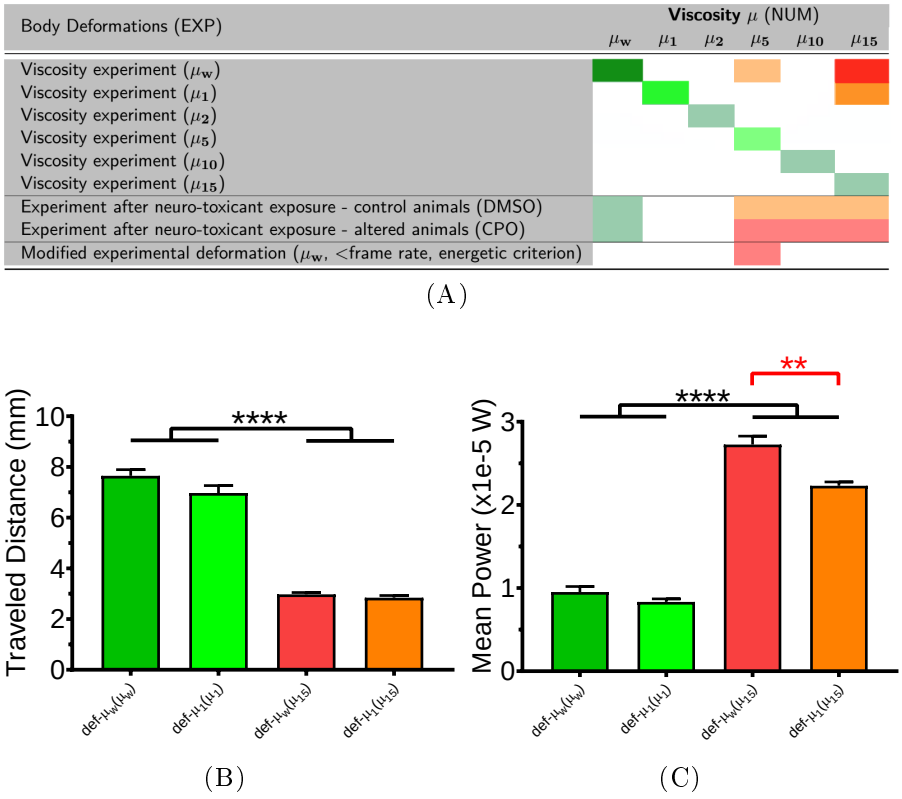


FIGURE 4: Mise en place d'un test d'effort. Toute simulation numérique repose à la fois sur les déformations expérimentales et sur le paramètre de viscosité. Alors, on peut simuler numériquement la nage du poisson-zèbre dans une viscosité différente de la condition expérimentale afin d'amplifier la dépense énergétique. **(A)** Six viscosités différentes ont été utilisées expérimentalement. Les cases colorées correspondent aux simulations numériques réalisées : vertes pour les simulations dites "réalistes" et rouge-orange pour les simulations dites "fictives". Pour chacune des conditions, les résultats de trois simulations numériques ont été analysées statistiquement. **(B)** Impact du test d'effort sur la distance parcourue. **(C)** Impact du test d'effort sur la puissance dépensée en moyenne.

Après s'être assuré de la validité et de la reproductibilité de notre approche numérique, par différentes études sur les paramètres du modèle et de la simulation notamment, de nombreuses applications de cette approche ont été démontrées *via* à la fois des simulations numériques dites "réalistes" et "fictives" telles que réalisées dans le cadre de la mise en place d'un test d'effort. Expérimentalement, deux études ont été menées : l'augmentation de la viscosité du fluide et la diminution de la concentration d'un neurotoxique particulier. En termes de perspectives, l'approche numérique pourrait certainement être améliorée en précisant les erreurs numériques dues à des questions de modélisation comme l'élasticité de la membrane entourant la queue ou bien l'impact de la réduction du nombre de degrés de liberté. Expérimentalement, de nombreuses expériences et simulations numériques seraient encore à mener afin d'augmenter significativement la taille des études en termes de nombre de réponses de fuite simulées. De plus, des tests d'efforts doivent être conduits expérimentalement afin de compléter les analyses commencées dans ce travail de thèse. Par ailleurs, un échantillonnage plus fin des gammes de concentrations et des simulations sur d'autres composants chimiques seraient pertinents pour avoir une meilleure idée du champs d'application de la méthode numérique développée.

List of Figures

List of Tables

Abbreviations

2D	**2** **D**imensions(al)
3D	**3** **D**imensions(al)
ALE	**A**rbitrary **L**agrangian **E**ulerian
ANOVA	**AN**alysis **O**f **VA**riance
CFD	**C**omputational **F**luid **D**ynamics
CoM	**C**enter **o**f **M**ass
CoT	**C**ost **o**f **T**ransport
CPO	**C**hlor**P**yrifos-**O**xon
CT	**C**omputed **T**omography
def-$\mu_x(\mu_y)$	escape response with **d**eformations observed in μ_x and numerical viscosity μ_y
dpf	**d**ays **p**ost-**f**ertilization
DMSO	**D**i**M**ethyl **S**ulf**O**xide
DNS	**D**irect **N**umerical **S**imulation
3-DoF	3 **D**egree **o**f **F**reedom
EFP	**E**lectric **F**ield **P**ulse
EFPMR	**E**lectric **F**ield **P**ulse **M**otor **R**esponse
fps	**f**rame **p**er **s**econd
HPC	**H**igh **P**erformance **C**omputing
IBM	**I**mmersed **B**oundary **M**ethods
IPC	**I**mage **P**oint **C**orrection
LS	**L**evel-**S**et
MAE	**M**onge-**A**mpère **E**quation
MEMPHIS	**M**odeling **E**nablers for **M**ulti-**PH**ysics and **I**nteraction**S**
MFF	**M**edian **F**in **F**old
MKP	**M**onge-**K**antorovich **P**roblem

MRGM	**M**aladies **R**ares [Rare Diseases] **G**enetic and **M**etabolism
NaSCar	**N**avier-**S**tokes **Car**tesian
OP	**O**rgano**P**hosphorus
PDE	**P**artial **D**ifferential **E**quation
PIV	**P**article **I**mage **V**elocimetry
SEM	**S**tandard **E**rror of the **M**ean
SPH	**S**mooth **P**article **H**ydrodynamics
SVD	**S**ingular **V**alue **D**ecomposition
WENO5	**W**eighted **E**ssentially **N**on-**O**scillatory **5** order-accuracy

Symbols

E_t	total expended energy	J
N_c	number of control points	–
N_e	number of body surface elements	–
N_s	number of body slices	–
n_t	number of last experimental image	–
N_θ	number of discrete angles per body slice	–
P_{max}	maximal power output	W(J.s^{-1})
P_{mean}	mean power output	W(J.s^{-1})
Re	Reynolds number	–
St	Strouhal number	–
$Tdef - pow$	average time delay between deformation input and power output	s
$Tdef - rot$	average time delay betweendeformation and rotation kinematics	s
$\mathbf{u}_\epsilon$	deformation velocity field	m.s^{-1}
$\mathbf{u_b}$	body boundary velocity	m.s^{-1}
X_a	arrival point of variables to be transported	–
X_d	departure point of variables to transport	–
χ_s	immersed body characteristic	–
$\delta_i t$	time between computational time and last experimental frame	s
$\Delta_i t$	time step of experimental acquisition	s
Γ_f	boundary of the computational domain	–
Γ_s	boundary of the immersed solid	–
λ	penalty factor	–
μ	fluid dynamic viscosity	Pa.s
ν	fluid kinematic viscosity	m^2.s^{-1}

ω	angular velocity	$\mathrm{rad.s^{-1}}$
Ω	computational domain	–
Ω_f	fluid domain	–
Ω_s	solid domain	–
ϕ	level-set function	–
ρ	mass volume density	$\mathrm{kg.m^{-3}}$
ρ_0	source image to be transported	–
ρ_1	target image to transport	–

Chapter 1

General introduction

1.1 *PhD* context and motivations

On the one hand, the Rare Diseases Genetic and Metabolism (MRGM) is an INSERM research group composed of biologists who actively study zebrafish neurobiology especially in terms of genetic, toxicology and pharmacology approaches. Therefore, zebrafish is used as an animal model to assess human diseases. For that purpose, they directly work with zebrafish animals, reared inside the laboratory. Currently, a custom-made software is used for analyzing multiple escape responses with a very high-speed camera. In the recent years, the MRGM team developed a specific electric field pulse (EFP) cell to stimulate zebrafish eleuthero-embryos in order to produce a strong, fast, and very stereotyped zebrafish escape response, also called C-start. On the other hand, the MEMPHIS INRIA team developed a high-performance computational fluid dynamics (CFD) code to simulate the flow around immersed bodies. As an application, they studied fish-like swimming by tuning the body geometry and its kinematics where a harmonic formulation was imposed. For the purpose of the study, only steady and periodic swimming was considered. However, they recently provided algorithms to use real-like kinematics based on given time snapshots, by focusing on optimal transportation algorithms. Since the last 5 years, the two research teams have initiated a collaboration which aim to combine numerical simulations with experiment-driven biological studies. In fact, CFD simulations of fish-like swimming are generally based on real-like body morphology and body kinematics but generally lacks of real-world experimental data. As for the experimental field, many imaging data can be generated but biologists would desire to use a numerical tool which could help and support in assessing and quantifying the swimming dynamics and energetics. Indeed, only kinematics can generally be estimated through experimental imaging which may not be sufficient to quantify the overall swimming performances. The numerical tools implemented by the MEMPHIS team and the experimental data provided by the biologists of the MRGM research group were significant enough to build an interdisciplinary collaboration, for a multidisciplinary

framework.
Afterward, my *PhD* was funded by an interdisciplinary grant to build and enhance the existing tools and potential interactions between experiment-driven research in biology, and computational and mathematical modeling. To that end, I interactively worked with two research teams for building and elaborating a numerical process based on experimental data which aims to explore biological questions and experiment-driven studies. The first collaborations and interactions between the MRGM lab and the MEMPHIS team have contributed to pioneer my *PhD* work.
Therefore, this *PhD* work is based on locomotion experiments and numerical simulations. Experimental locomotions consisted of video recording of zebrafish eleuthero-embryo escape response after an EFP stimulation. As a result, my *PhD* work is core towards both numerical modeling and experiment-driven biological studies. Finally, we aim to integrate the experimental data *i.e.* escape response videos within a highly validated CFD code to produce real-world numerical simulations. Note the body kinematics requires to be reconstructed in 3D in-between. Thus, the following state of the art first introduces the interests of both the zebrafish model for locomotion, and numerical simulations of fish-like swimming. Afterward, the state-of-the-art methodology for analyzing zebrafish escape responses, or performing self-propelled swimmer simulations is detailed. Following the state of the art, the general objectives of this *PhD* work are presented before closing the chapter with the outline of the present thesis.

1.2 State of the art

1.2.1 Locomotion of aquatic animals

Locomotion of aquatic animals first captivated biologists before dragging multiple and various research fields which now generate more and more interdisciplinary works. As observed in nature, the kinematics of swimming animals can vary significantly between species, according to the body morphology. First, two groups are generally distinguished among teleost fish species [107]: Body and/or Caudal Fins (BCF) and Median and/or Paired Fins (MPF) kinematics which denote the body parts mainly used for moving. Most species propel themselves with backward traveling waves along the body. However, the traveling wave can be restrained to the tail which denotes the so-called Carangiform swimming or moves across the entire body length, characterizing the so-called Anguiliform swimming. Actually, Sfakiotakis *et al.* [107] distinguished four swimming modes: anguiliform, subcarangiform, carangiform, thunneiform: from the eel-like to the tuna kinematics. Besides, the swimming mode can differ according to the developmental stage. Indeed, fish morphology undergoes significant changes from larval stages to adult age which considerably impacts the hydrodynamics of swimming [80]. Fish larvae have a specific shape which they employ to fulfill several needs. In the

early developmental stages, fish larvae do present multiple locomotion behaviors, often related to specific functional needs. In this regard, we refer to the recent review of larval fish swimming [130] which recapitulates the larval kinematic and energetic behaviors. In addition to the biological functions, the biomechanic powers are investigated. Regarding zebrafish (*Danio rerio*) larvae, the locomotor behaviors including fast-starts have been well-studied since few decades [21, 31, 89, 93]. The main question constantly raised concerns the evaluation of fish swimming performances, by assessing the power output required to move and propel its body alongside swimming velocity [104]. For instance, Sir James Gray asserted in 1936 [48] that the power requirement for dolphins, which are mammals, to reach theirs cruising speeds, had to be up to seven times higher than actually observed energetics. Afterward, the pioneering mathematical work of Sir James Lighthill [71–74] helped to understand the link between the body kinematics and swimming performances under simplified assumptions such as inviscid flows, anguiliform, and steady swimming which allowed to separate drag and thrust forces. Basically, the body forces were estimated using analytic formulations of drag and thrust. Therefore, fish locomotion has originally been mainly studied through empirical observations and theoretical analysis of hydrodynamics. The underlying questions concern the optimal morphology or kinematics for swimming efficiently. That is why, many studies originated from airfoil theories by reducing the fish body and its kinematics to a flapping wing [33, 117]. Note Gray was one of the first to link aquatic locomotion and foils. Performance studies across various marine species, analyzed Strouhal and Reynolds number relationships, and consistently found an optimal Strouhal number matching the empirical observations in nature [32, 40, 113, 117]. Meanwhile, digital particle image velocimetry appeared to assess the fluid flow velocity and the wake generated behind a steady fish swimming. At this stage, undulatory fish propulsion has been well-studied (see [62] for review) but the authors of the review highlighted the crucial need for computational methods and the integration of intern biological processes to understand the exterior propulsive biomechanics.

CFD has then emerged in the last decades to enlighten the links between swimming speed, body kinematics, and swimming efficiency [14, 15, 40]. We refer the reader to the review of [75] to comprehend how CFD methods can expand the understanding of fish swimming which in turn, significantly challenges the current numerical methods. In fact, insect flying and fish swimming constitute specific and unique natural cases of strongly unsteady phenomena within relatively steady flows [75]. For aquatic locomotion, unsteadiness is often generated by escape responses and complex maneuvers, fin motions, fish schooling, or turbulent flows. The latter case has been reviewed in [70]. Actually, fish propulsion mechanics can be decomposed into an active contribution generated by the body deformations and a passive part from the interaction between the surrounding hydrodynamics and body morphology [36]. To visualize, passive swimming is also called "dead fish" swimming. Experimental studies showed a passive body can effectively gain energy from upcoming vortices [5]. Recently, many numerical studies investigated into

the performances of fish schooling [42, 63, 64, 126]. Due to inherently numerical essence, CFD methods naturally led to optimization issues and engineering applications. Indeed, multiple numerical simulations of self-propelled swimmers aimed at optimizing the shape of the swimming body alongside its kinematics, regarding the resulting performance outputs [10, 12, 16, 43, 59, 59, 125]. Eventually, the numerical simulation can be oriented towards bio-inspired applications such as underwater vehicles [112, 122]. Note biomimicry also leads to the design of robotics. Among the bio-inspired robotics, we can find a category which tends to support fish swimming investigations. For instance, a fish body has been entirely or partly reconstructed to reproduce animal kinematics such as C-starts and test hypotheses on the generation of propulsion [4, 34, 78, 93, 118]. This class of robotics has emerged to support biologists in experiment-driven studies and give new biological insights [47]. In that regard, bio-inspired numerical simulations can also be used to provide information, non-accessible in experiments [66, 69, 81, 82]. Eventually, data-driven stochastic models which describe the three-dimensional trajectories in zebrafish locomotion can be found to reduce and refine the use of zebrafish animals according to the experiments [91, 92].

Since the last decades, zebrafish became a well-popular animal model regarding biological investigations and medicine applications. On the one hand, zebrafish animals do present strong similarities with humans in terms of biological processes involved in locomotion: a nervous system including brainstem, spinal cord, and muscles. Moreover, similar physiological and pathophysiological mechanisms can be found in zebrafish. Therefore, human diseases and toxicological molecules can be studied through the zebrafish model. On the other hand, zebrafish are very convenient as a research animal model. First, adult zebrafish sizes only a few centimeters in length and thus, a massive amount of zebrafish can be reared inside breeding facilities. Secondly, zebrafish presents high fertility and is oviparous. Furthermore, the development of zebrafish is external and extremely rapid for a vertebrate species. Early developmental stages can be observed under the stereo-microscope. Until an advanced stage of larvae development, the transparency of the body enables the observation of the establishment of body structures and the use of numerous genomic and biotechnological tools. Finally, zebrafish can easily be genetically modified. As a result, the zebrafish model drains multiple disciplinary fields. For example, a complete 3D reconstruction of a larval zebrafish brain has been performed to construct the neuronal map [52]. Besides, the zebrafish locomotion including the swimming kinematics, performances, and behaviors, is well-studied from aging and physiological fields [44] to toxicology and pharmacology [39, 115]. In fact, the zebrafish locomotion behaviors have been well-described and present very stereotyped patterns [21, 31, 89]. In particular, zebrafish eleuthero-embryo *i.e.* the developmental stage after hatching but before the first feeding, presents a highly stereotyped pattern when escaping predators or any form of danger. The so-called C-start movement characterizes the escape locomotion and is composed of three stages, described in Fig. 1.1. First, the fish body initiates with preparatory motion where it turns into a C-shape, also called the

C-bend. Afterward, the eleuthero-embryo pushes away a large amount of fluid to gain propulsion: this is the propulsive step initiated by the counter-bend. Finally, the fish body accelerates away from the danger by using fast-swimming undulations. Fast-starts can be ignited with tactile [39, 89], visual and acoustic [21, 22, 30, 55, 93] or electrical stimuli [22, 120]. The escape response from an electrical stimulus is ultra-rapid as the electric field pulse (EFP) directly affects the Mauthner cells and thus, the peripheral nervous system [111].

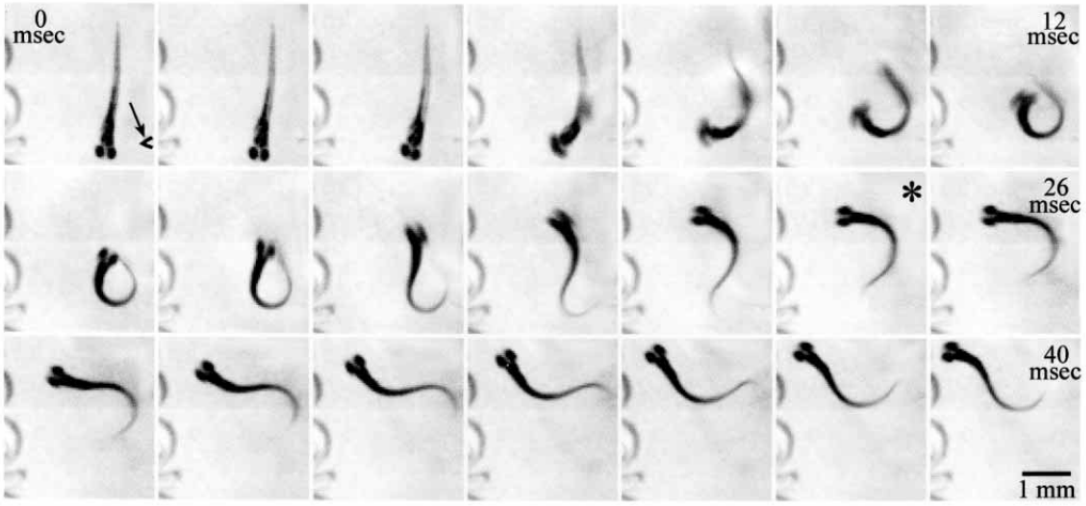


FIGURE 1.1: **Representation of a typical zebrafish C-start escape response, from [21].** Notice the large C-bend has a duration of $t = 12$ ms and the counter-bend also presents a C-shape (frame 13, asterisk). Note the eleuthero-embryo was stimulated by acoustic vibrations and video recording was performed at 500 frame per second (fps).

Similarly, electromyography has also been used for analyzing muscle activity and C-starts of *Polypterus senegalus* [120]. The study highlights how strong the C-bend shape can be with electric stimuli. According to the electric sensitivity, the tail gets more or less closer to the head. Note the rotation motion is also largely affected during the preparatory phase. Most of the teleost fish species can perform a C-start when escaping predators. In general, velocity and acceleration should be maximized as acceleration peaks and timing of propulsive power correspond to tail-beat frequency [31]. However, turning angles which are highly variable might have an important role, depending on the source of danger [31]. Therefore, fast-starts are the results of a trade-off between swimming performance, energy consumption, and escape behavior [31]. In particular, a correlation between the stimulus intensity and the final escape performance can be highlighted [120]. Strong and weak C-bends have been compared and showed the variability in strength is increased by the duration of stage 1 and the muscle activity during stage 2 is prevalent to the maximum velocity [120]. C-start kinematics has also been highlighted by a computational model which revealed it does maximize the traveled distance although higher escape distance could be reached with fictitious body curvatures [43]. Numerical simulations also unveiled C-starts are energetically inefficient [43]. Zebrafish eleuthero-embryos can also speeds-up using different curvatures, at the expense of swimming efficiency and

transport costs [67]. In addition, stage 1 produces essentially rotational motions while propulsive forces are generated during stage 2 with the highest velocity, power, and efficiency [67]. As for strong C-bends, zebrafish eleuthero-embryos might intercept their own wake without affecting the global swimming performance [68]. Eventually, double-bend escape responses have been defined to describe specific deformation patterns which increase the escape performance [31, 37]. The C-start escape response is particularly unique and reproducible when using EFP activation. As a result, the motion describes completely unsteady physical phenomena and the interaction with the fluid flow can barely be predicted. In that regard, CFD methods are key to understand the underlying biomechanics of strong and weak C-starts. From a fluid mechanics point of view, we can see during the two first stages of C-start, up to three jet flows are produced near the head, the mid-body, and the tail [17, 65, 67, 68, 90, 121] which then derive into vortices shed in the wake by the tail beats. During the fast-swimming stage, two rows of vortex rings can be observed both experimentally [90] and numerically [43, 65] which already has been well-described in steady swimming for anguiliform swimmers [15, 59]. Note the structure of the wake can significantly change for carangiform swimmers according to the Strouhal number [12, 14, 87, 94]. From an experimental point of view, any locomotion misbehavior can experimentally be observed which enables to screen a large number of pharmacological molecules on intoxicated and mutant animals when the locomotion is affected (see Fig. 1.2 issued from [39] and Fig. 1.3 from [115]). In the latter case, notice how fast-start and slow swimming kinematics show different and opposite impacts in the control locomotions, which illustrates how fast-starts do maximize the escape response and therefore enable experiment-driven studies based on altered escape swimming. The locomotion behavior of fishes can be altered either by genetics [22, 38, 39, 115] or the surroundings environment. For the latter case, we can find in the literature experiments with viscous fluids [28–30, 55].

As noticed by [79], zebrafish eleuthero-embryos and larvae swim within an intermediate flow regime when coasting, defined by a Reynolds number (Re) from 10^1 to 10^3. Note the authors defined the larval viscous regime as $Re < 300$. As the zebrafish grows, the swimming regime significantly changes and adult zebrafish operates naturally in inertial Reynolds regimes ($Re > 10^3$). We assume zebrafish eleuthero-embryos do maximize the muscle power output when performing C-starts. In [30], zebrafish escape responses have been challenged with highly viscous flows in order to analyze the resulting escape performances. The authors suggested the escape response presents an important decrease in maximal swimming velocity and an increase in the power output while stages timings based on the heading angle, did not seem impacted. In [119], the authors found a phase lag between body curvature and body muscle activation in high viscosity, when modeling the fish deformations as an elastic body.

As stated previously, computational models are designed to describe precisely fluid flows and aquatic locomotion but are challenged for reproducing real-world and highly

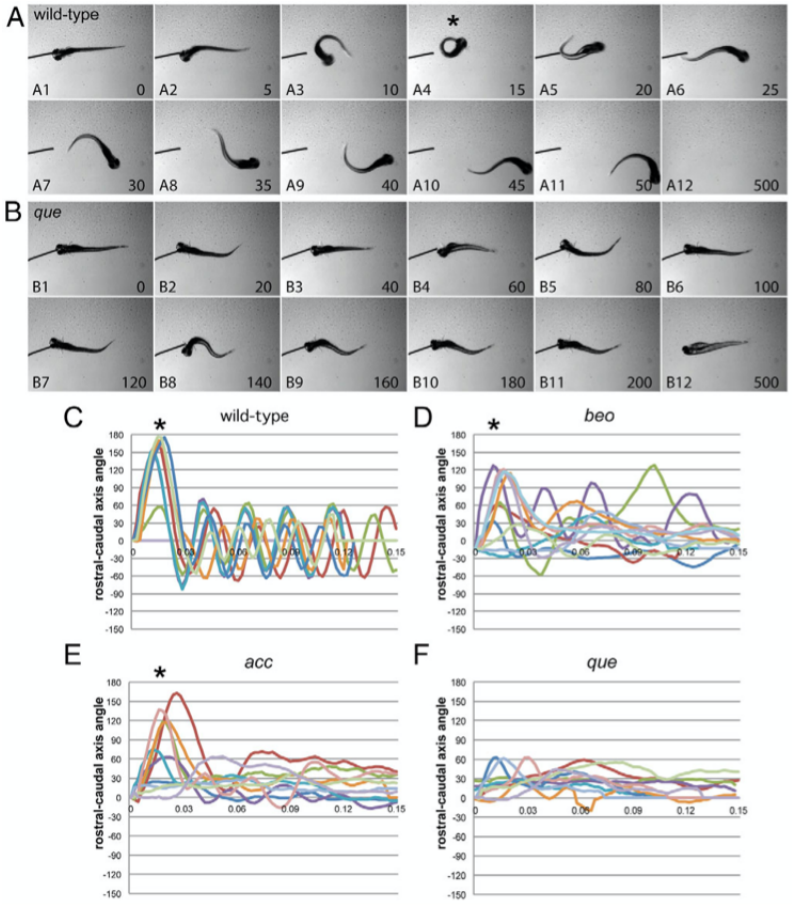


FIGURE 1.2: **Illustration of a biological locomotion study with mutant animals, from [39].** A wild-type 5 dpf zebrafish eleuthero-embryo shows a typical C-start escape response in response to a touch stimulus (**A**) while a *que*-mutant animal exhibits an abnormal behavior (**B**). Note times are indicated in ms. Ten body amplitudes are then represented for the three mutant animals: *bandoneon (beo)* (**D**), *accordion (acc)* (**E**) and *que* (**F**), and wild-type zebrafish (**C**). Note only *que* mutants cannot perform strong amplitude C-bend.

unsteady dynamics such as C-start behaviors. That is why, CFD methods are not commonly used for helping biological and experiment-driven research. As for the biology field, swimming energetics is generally unreachable despite the importance to analyze impacted zebrafish swimming behaviors. Therefore, my *PhD* project resides in the combination of advanced self-propelled swimmer simulations and access to real-world experimental data including the actual experiments with zebrafish eleuthero-embryos.

1.2.2 Methodology

We present in this chapter in detail the current state of the art of each discipline involved in this work. Three main contributions can be identified: the reconstruction of a 3D shape alongside its escape deformations, the integration of experimental deformations within a CFD code, and the interaction of our numerical modeling with biological investigations. The latter point of interest resides in the implemented applications

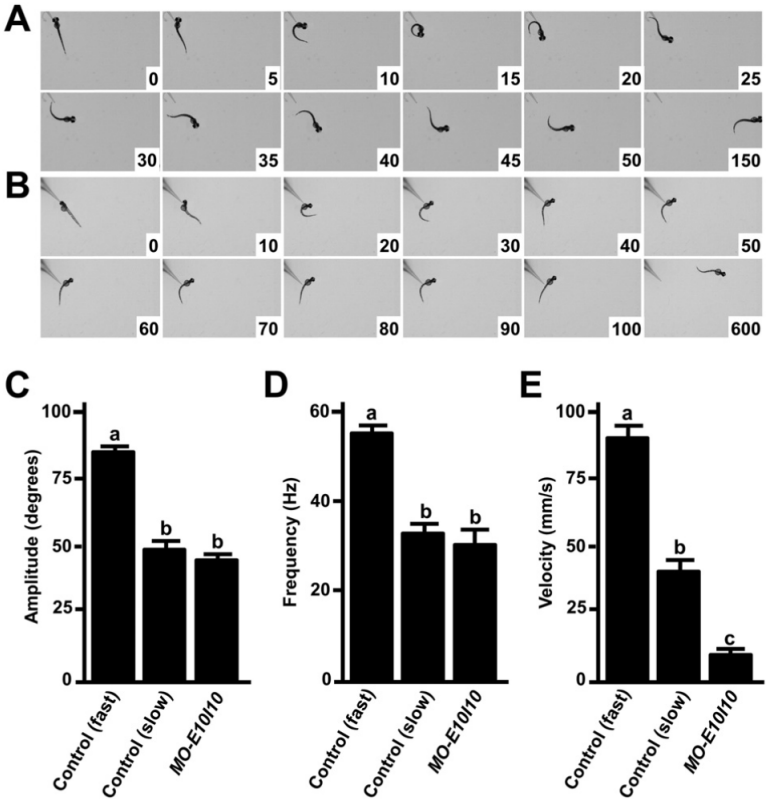


FIGURE 1.3: **Second example of a biological locomotion study with mutant animals, from [115]**. Comparison of dorsal views of high-speed videos between a wild-type 3 dpf zebrafish response and an *abhd12*-mutant zebrafish eleuthero-embryo abnormal response (**A-B**). Note the difference in modules timings, indicated in ms. Kinematics (**C-E**) such as bend amplitude (**C**), tail-beat frequency (**D**) and swimming speed (**E**), between control and morphant phenotypes raised significant alterations.

considering biological issues and enlightened the potential interdisciplinary impacts of this work. Beforehand, the general framework used by biologists to study zebrafish locomotion is presented. In particular, we detail the current context in the MRGM INSERM team to perform toxicological and pharmacological studies. Above all, we introduce the numerical framework in which our work falls such as self-propelled swimmer simulations, generation of 3D fish geometry for CFD, zebrafish C-start simulations, validation against real-world data. Finally, we focus on the work developed by the MEMPHIS INRIA team which implemented new techniques towards future combinations of self-propelled swimmer simulations and real-like data.

1.2.2.1 Methodology in biology research for swimming analysis

To study zebrafish locomotion, biologists have developed various experimental methods, especially to extract the body and swimming kinematics. Historically, the zebrafish kinematics has been analyzed through the midline kinematics. Two decades ago, the midline deformation was measured *via* the so-called head-tail angle (Fig. 1.4), estimated

with two segments only. Thus, the head-tail angle is maximal when the fish is straight undeformed. Alternatively, one can consider the bend amplitude ([115]) defined as the head-tail angle supplementary such that the bend amplitude is zero when the zebrafish is undeformed and straight.

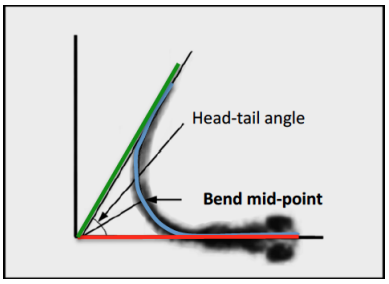


FIGURE 1.4: **Definition of the head-tail angle, from [21].** Note only three segments are required to compute the head-tail angle although a generalization with a finer midline approximation is also possible.

Afterward, the midline curvature has been assessed by manually digitizing the midline [17, 29, 30, 50, 85, 89, 90, 121] (Fig. 1.5). The center of mass was then approximated using the midline segments. From the midline kinematics, various kinematic parameters have been derived: tail-beat frequency, wavelength and wave amplitude, Re, swimming speeds, and curvatures. Note, the amount of work was certainly acceptable provided the low frequency of data acquisition ($\approx 500 - 1000$ frame per second (fps)). Besides, the curvature of the head remained zero for each swimming behavior recorded [89]. Therefore, the authors considered the head as strongly stiff. Besides, the particle image velocimetry (PIV) technique has been used repeatedly to evaluate the fluid flow patterns generated by the zebrafish eleuthero-embryo and larva's motion [29, 61, 85–87, 90, 94, 121] (Fig. 1.6). PIV methods provide an estimation of the full two-dimensional (2D) fluid velocity field although the obtained resolution and accuracy are low.

In the last decade, tracking methods have been developed to track accurately the swimming kinematics. In 2D [38], a silhouette-based method has been implemented by superimposing a predefined contour shape onto the actual zebrafish shape (see Fig. 1.7). In definitive, the modeling of proper geometry and a swimming motion have been required to estimate the optimal projection onto the deformed and moved shape with a Kalman filter. Thus, even occlusions have been easily handled with this model. Note in most of the 2D analyses, automated tracking, midline digitization or PIV calculations assume the swimmer does present a planar and horizontal locomotion. Although recent studies showed that eventually, 3D motions might occur during fast-starts [37, 93], we can generally consider the produced pitch and roll motions are negligible. As most of the fish species are inherently asymmetric especially with respect to the dorso-ventral axis, the stabilization of the body during strong maneuvers such as fast-starts still needs to be fully understood [13]. Eventually, pectoral fins could have an important role but stay in most of the experimental observations, completely adducted to the body during fast-starts [31, 89].

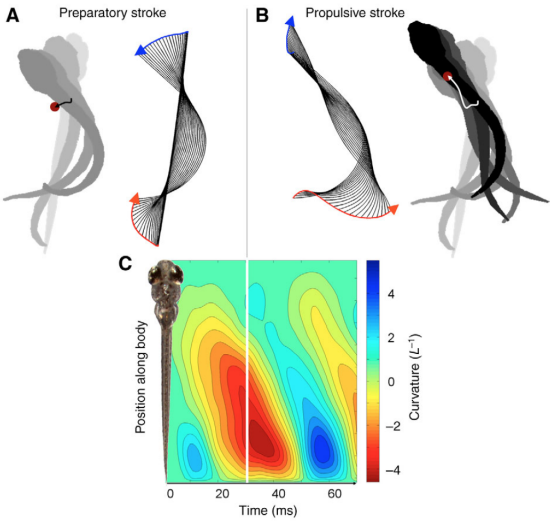


FIGURE 1.5: **Body deformations are generally computed with midline digitization and the calculation of midline curvatures.** The example extracted from [90], presents the C-start swimming kinematics of a 5 dpf zebrafish larva for the preparatory (**A**) and propulsive (**B**) strokes, represented by silhouettes and midline kinematics. The overall body kinematics is well-represented by curvature along the length of the fish and across time (**C**). Convex and concave motions are illustrated in blue and red, respectively.

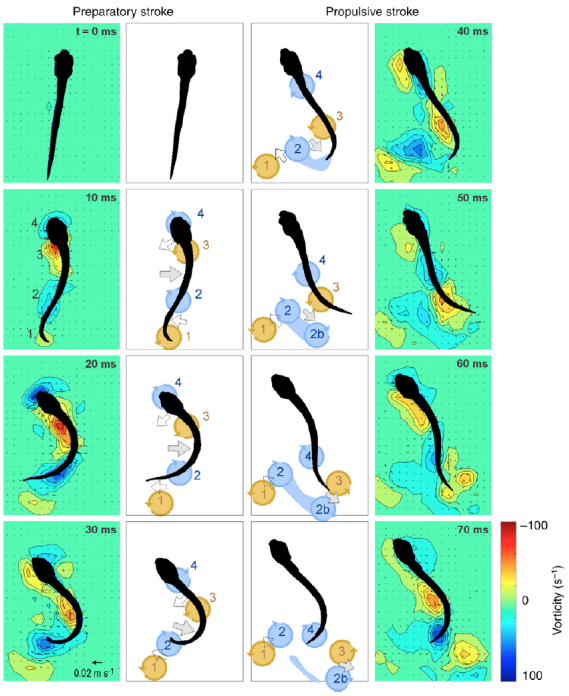
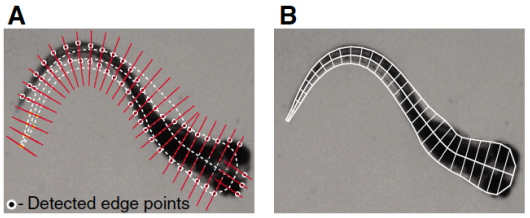
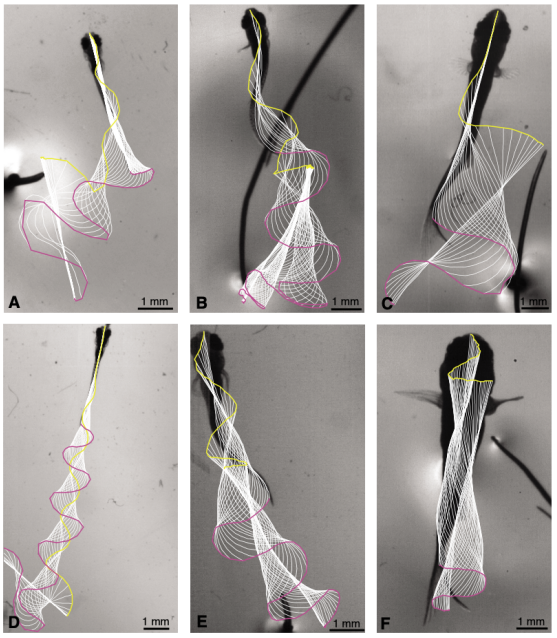


FIGURE 1.6: **Particle image velocimetry (PIV) visualization of the flow.** PIV has been widely used to analyze the wake structure. Essentially, PIV measurements provide an estimation of the velocity field of the flow. Note the sequence, from [90] is identical to Fig. 1.5.



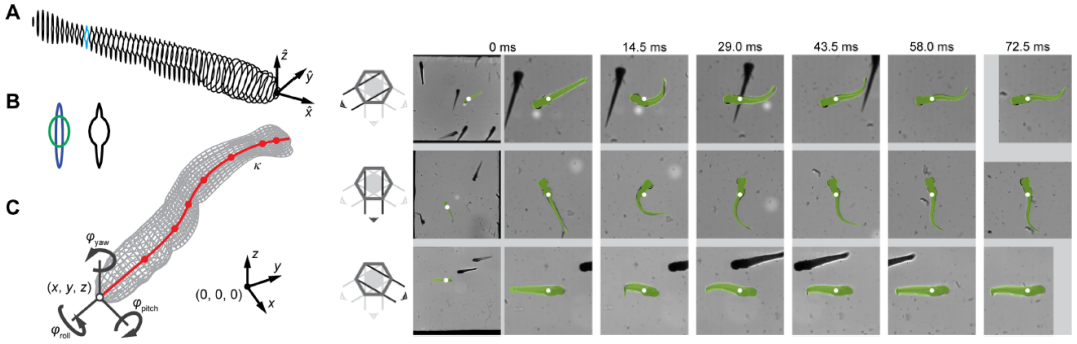
(A) Model matching.



(B) Automated 2D tracking.

FIGURE 1.7: **Example of a 2D automated tracking method, from [38].** The model is adjusted at each time step *i.e.* frame (Fig. 1.7A): first the model location is estimated (**A**) before fitting the midline of the model (**B**). Note the boundary may not correspond exactly due to small out-of-plane motions. We can see how the zebrafish is deformed for each video frame: cross-sections stay perpendicular to the estimated midline. The 2D contour is adjusted to each experimental frame with a specific projection-based research algorithm. The resulting tracking results (Fig. 1.7B) are based on zebrafish aged of 5 dpf (**A,D**), 15 dpf (**B,E**), 28 dpf (**C,F**). First row are wild-type zebrafish while second raw are *stocksteif* mutants.

In 3D, the surface of a fish has first been reconstructed manually [78, 93] before modeling the surface with ellipsoidal cross-sections [80] and tracking the midline deformations by using 2 orthogonal views [23]. Recently, a more accurate model has been created to automatically reconstruct the full experimental kinematics in 3D from iterative projections onto the targeted shape [124, 131] (see Fig. 1.8). Usually, tracking algorithms enable to enhance and automate the computation of swimming kinematics. In addition, accurate Lagrangian-like tracking enabled to estimate the dynamics and simple energetic quantities of the zebrafish locomotion by using inverse-dynamics methods [124, 131] (see Fig. 1.9). However, the power consumption remains to be accurately and reliably computed.



(A) 3D modeling. (B) Automated 3D tracking.

FIGURE 1.8: **Advanced automated tracking method based on projection-based iterations, from [131].** First, we can see (Fig. 1.8 A) the construction of the *in silico* zebrafish. The initial zebrafish 3D shape is formed by cross-sectional shapes (**A**), based on ellipsoidal parameterisation (**B**) of body (green) and median fin-fold (blue). Afterward, the 3D reconstruction is deformed (**C**) by prescribing a parameterisaion of the curvature along the midline (red line) at control points (dots). Transverse sections stay plane and orthogonal to the midline. Secondly, we can see (Fig. 1.8 B) the overlap between tracked *in silico*(green) and experimental shapes across time through three camera views. Thus, multiple orthogonal views are required to reconstruct and track a full 3D motion.

1.2.2.2 Self-propelled swimmer simulations

In order to obtain a precise computation of the swimming performance, CFD methods have been extensively used as they are specifically designed to numerically compute the fluid flow, the fish motion, and the resulting energetics from the flow equations. Indeed, the full understanding of active swimming and biomechanics has been enabled by the CFD field. Basically, CFD simulations consist of computing the solution of the governing equations at each time step. As outputs, the numerical simulation provides the pressure map and the 3D velocity field of the whole computational domain alongside the velocity of the immersed zebrafish body, and all the forces and torques exerted onto the body surface. As a result, we can recover the flow vorticity characterizing the vortex shedding and all the desired energetic quantities derived from the instantaneous power expended by the zebrafish to compensate hydrodynamic forces. Generally, the fluid velocity and pressure are solved by approximating the so-called Navier-Stokes equations while the body kinematics is computed from the classical Newton's laws which govern rigid motions. Nevertheless, the swimmer is also a deforming body with specific deformation kinematics. In the case of self-propelled swimmers, a numerical simulation needs to tackle three major bottlenecks to be realistic. On the one hand, the immersed fish body has to be handled adequately to obtain accurate, consistent, and fast computations. Indeed, the computational meshes and numerical methods employed are prevalent to obtain a proper simulation. As stated before, the solution of the governing equations is updated at each time step across the whole computational domain meaning the domain is discretized with a massive amount of points, which then define the computational mesh. Thus, all the computed quantities are located within the mesh

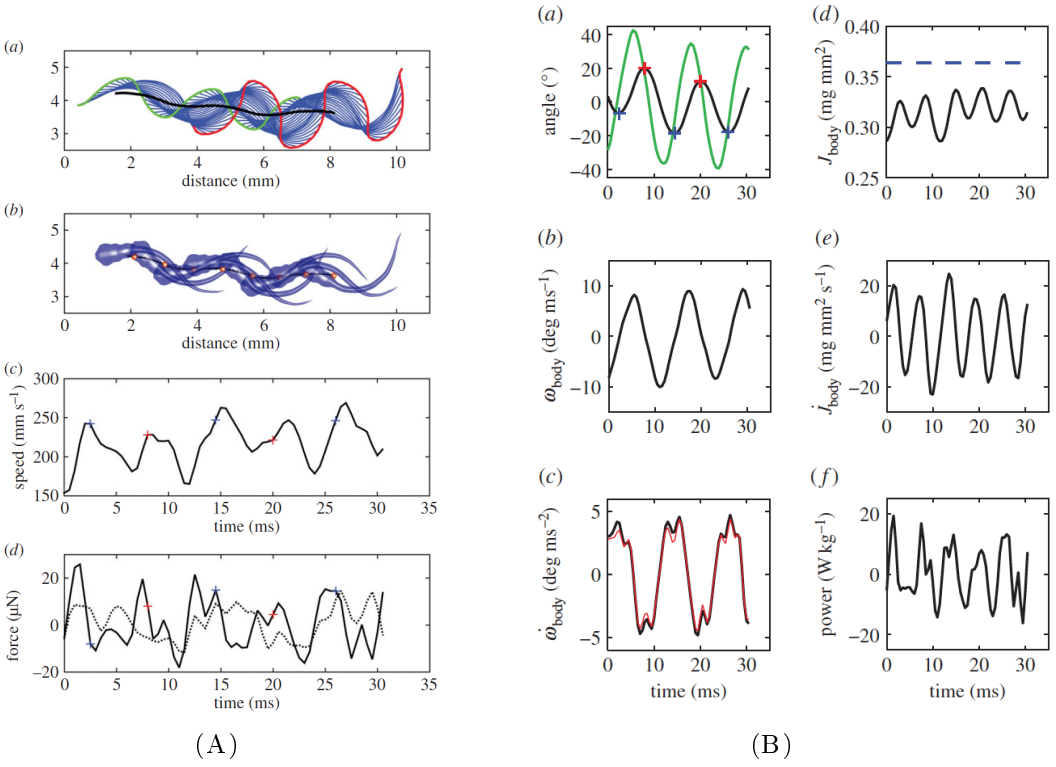


FIGURE 1.9: **Inverse-dynamics approaches have been used to estimate forces and power, from [124].** First, a swimming sequence of a 5 dpf zebrafish with body shapes (**b**)) and midline kinematics (**a**) are presented (Fig. 1.9 A). In addition, are computed the translational speed of the center of mass (**c**) and the forces (**d**) in the direction (continuous line) or orthogonal (dotted line) to velocity vector. Note blue and red markers correspond to extrema of body rotation and are not exactly synchronized with swimming speed maxima. Secondly, we can see (Fig. 1.9 B) rotational variables such as body (black) and heading (green) angles (**a**), angular velocity (**c**), angular acceleration (**e**) and the moments of inertia changes (**b,d**). Power can also be estimated through inverse-kinematics (**f**). Note differences between body angle and heading angle are particularly highlighted through blue and red markers. In particular, we can see a constant phase delay.

nodes. However, the information about the fish body such as its position or velocity, has to be accessible to the computational mesh. In fact, two classes of methods can generally be distinguished: body-fitted and immersed boundary methods. The first family denotes numerical schemes based on body-fitted grids *i.e.* the contour of the fish body is represented by mesh points. Instead, the latter class of methods provides the immersed body information implicitly *via* appropriate numerical schemes. Body-fitted methods are widely used to accurately represent the surface and facilitate the surface calculations. Nevertheless, the computational costs are often prohibitive for 3D moving and deforming objects such as large and fast fish turns. In such cases, the mesh might require adaptions or a complete remeshing step, to follow the fish movements accordingly. Instead, immersed boundary methods handle straightforwardly any body motions with simple and fixed meshes such as Cartesian grids. The main limitation resides in the accuracy provided at the interface. That is why, only a few

studies use body-fitted meshes in the literature for self-propelled swimmers. Apart from early tadpole [76, 77] and eel simulations [24, 59], a series of papers have performed arbitrary Lagrangian-Eulerian (ALE) simulations based on overset grids [58, 65–68]. Besides, multiple variations of the original immersed boundary method (IBM) (Peskin 1972 [99]) can be found in the literature. Since the last decade, the immersed boundary is taken into account with an original [119] or an hybrid [16, 17, 45] IBM, smooth particle hydrodynamics (SPH) [51], remeshing vortex methods [41, 43, 125], a constraint-based formulation [27, 98, 108] or a so-called Brinkman penalization method [9–12]. A complete review of IBM-based simulations can be found in [110]. On the other hand, the geometry of the zebrafish and its body deformations have to be prescribed thoughtfully as they are key to simulate real-like fish locomotion, especially if we are interested in real-world swimming performances. After a start in 2D simulations [10, 24, 119], 3D fish-locomotion simulations have emerged in the recent years. The first 3D reconstructions of fish geometry used simple mathematical shapes such as cubic splines [12, 43, 125] or ellipsoidal cross-sectional views [41, 51, 59, 108]. Note triangular meshes are not common with IBM excepted for specific methods [16, 17, 45]. The bluegill sunfish was one of the first species to be meshed in 3D with triangular elements, based on computed tomography (CT) scans [17]. The first real-like zebrafish eleuthero-embryo 3D reconstruction was obtained by manual digitization [58, 65, 67] (see Fig. 1.10). Moreover, an ellipse-based model has been calibrated with few manual digitization of the zebrafish surface [49]. As introduced before, an enhanced ellipse-based model computed through a projection-based and experiment-driven fitting algorithm [124, 131] has also been used for CFD simulations [129]. Eventually, 3D models have been reconstructed from actual images [11].

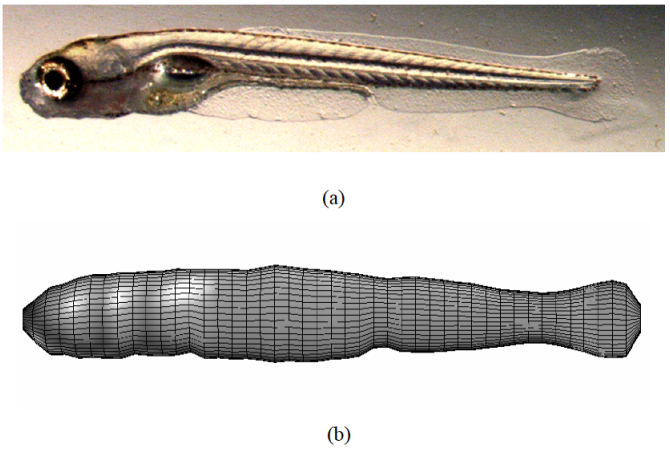


FIGURE 1.10: **First 3D zebrafish reconstructions have been generated manually.** Comparison between real picture (**a**) and computational shape (**b**) of a 5 dpf zebrafish. The picture was taken from [58]. Such realistic surface mesh were used to compute numerical simulations [58, 65, 67].

In general, 3D reconstructions of fish body deformations consider the fish body as a beam model where each cross-section remains orthogonal to the midline across the

deforming movement [12, 23, 30, 38, 131]. In addition, only rigid transformations affect each cross-section *i.e.* assuming neither shear stress nor dilatation are involved in the deformation process. Moreover, the volume mass density is considered as homogeneous in most of the 3D zebrafish modeling such as [131] even though the posterior of zebrafish eleuthero-embryos is surrounded by a thin membrane also called median finfold (MFF) and composed of different cells than the body cells. Furthermore, the so-called MFF which will later develop into median fins is a structure which highly contributes to propulsion and escape performances. However, only a few studies have modeled the complexity of this structure and its deformation behavior in the case of large body bending [123]. Therefore, the role of the MFF during fast-start responses would still require more investigation [75].

As for the body kinematics, most of the self-propelled swimming simulations have historically modeled the swimming deformations as an harmonic wave meaning the midline kinematics is modeled with a sinusoidal function including the tail-beat frequency and the desired envelope amplitude [9, 10, 12, 16, 24, 41, 43, 51, 58, 59, 66, 67, 69, 108, 125]. The formulation is then prescribed by altering either the midline curvature or its lateral excursion. Indeed, mathematicians first investigated the steady, periodic, and inline motions to separate thrust and drag and compute the swimming efficiency and propulsive power. Afterward, C-starts, burst-and-coast or turn motions observed for food capturing have been derived with custom amplitude functions [10, 43, 58, 67, 69]. Eventually, bio-inspired studies recovered the tail-beat frequency and amplitude envelope directly from experimental observations [66, 67]. In definitive, the body deformations have for a long time been represented by a swimming law to describe the undulatory motion, eventually calibrated with observations. Note the body deformations are imposed and therefore substitute to any elastic modeling step. Thus, only a few studies modeled the actual biomechanics which produce the body deformation. For instance, a neuro-mechanical model has been developed in [98, 119]. Besides, fish deformations can also be described by a beam model [25]. Eventually, the bending moment has been recovered from direct numerical simulations combined with a biomechanic model [129]. In the latter study, the body deformations are recovered *via* the swimming kinematics from the 3D experimental tracking meaning no rigid motion is then computed. Furthermore, notice unsteady behaviors and particularly strong C-starts constitute challenging kinematics to enforce directly with a mathematical model and that is why, studies [68] have recovered the body kinematics from actual experimental observations of C-starts, under strong assumptions. Indeed, the authors considered the body's rotation could be reduced to the so-called heading angle which basically denotes the swimming direction of the head. The limits of this assumption have been enlighten by the authors in [124, 131] (Fig. 1.9). Indeed, they found the heading angle is shifted in time relatively to the actual body angle, the delay being roughly half a period. Essentially, the body angle which also denotes the CFD rigid rotation has been estimated by integrating over the contribution of each mass element. To determine the rotation of

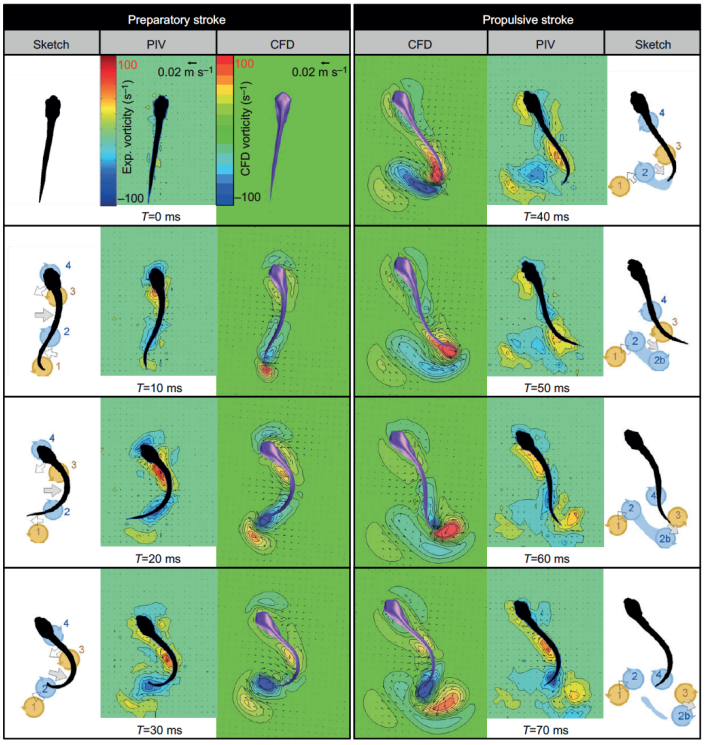
deformed body shapes, only a few alternatives can be found in the literature. In [50], a singular value decomposition (SVD) has been used to estimate the main orientation of the elongated body, in the case of slow swimming motions. Such algorithms can be generalized in specific situations to the so-called Kabsch algorithm which finds the optimal rotation matrix considering an affine transformation. This rotation searching also known as a *Procrustes problem*, is crucial to compute the deformation components of body kinematics. Finally, note the authors of [65] and [27, 49] used an experimental midline curvature as input to the numerical simulation of a larval zebrafish locomotion. The swimming kinematics might also be be directly extracted from the manually [17] or automated [129, 131] midline tracking from experimental videos. To our knowledge, the previous studies achieved an important and original step towards experiment-driven numerical simulations. Nevertheless, the rigid kinematics of the body was not calculated but prescribed.

The main limitations of self-propelled numerical simulations reside in the lack of proper validations against actual experimental data, especially in the case of C-start simulations. In [49], the experimental and numerical fluid flows near the pectoral fins have been highlighted and compared by using methylene blue dye. Most of the time, computational studies have been validated against velocity fields issued from PIV [17, 43, 66, 67] (Fig. 1.11). Only few have directly compared the actual experimental and numerical kinematics such as speed and trajectory (Figs. 1.13, 1.12, 1.14) [27, 65, 67, 68].

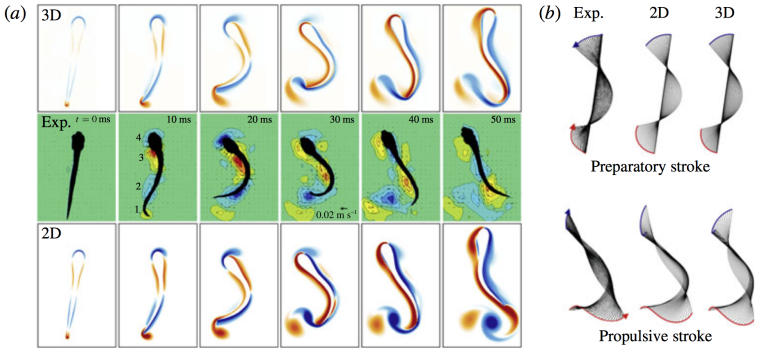
Finally, first developments have been initiated for reconstructing the 3D body deformations from a series of experimental images [11]. As a consequence, very few computational studies have performed numerical simulations of zebrafish larval aperiodic swimming and validated against experimental data to reproduce, support, and refine biological experiments.

1.2.3 The MRGM research group of the University of Bordeaux

The MRGM team is composed of biologists and hospital medical practitioners which notably work on the troubles and human diseases affecting the neuronal and locomotor system. To that end, the research group developed and works routinely with the zebrafish model to prospect the different behaviors induced by various zebrafish stimulations such as visual, acoustic, and electric stimuli, in response to genetic or environment changes (see for example Fig. 1.3). In particular, the MRGM team investigates the effects of organophosphorus (OP) compounds used, for example, as pesticides, flame retardants, or warfare nerve agents. This family of toxicants affects significantly the zebrafish escape locomotion (Fig. 1.15). The team notably developed a complete in-house cell for stimulating zebrafish eleuthero-embryos with an EFP system and recording the resulting locomotor behaviors with a high-precision recording system. The experimental EFP cell is particularly unique as entirely implemented by the MRGM team. To our



(A) A 2D PIV comparison for a 3D numerical simulation [67] .



(B) PIV comparisons for both 2D and 3D numerical simulations [43].

FIGURE 1.11: **Computational studies mainly use PIV experiments to validate escape swimming simulations.** Pictures are taken from [67] (Fig. 1.11 A) and [43] (Fig. 1.11 B). Note all simulations reproduce the same experimental swimming kinematics (Fig. 1.11 B, (**b**)), from [90].

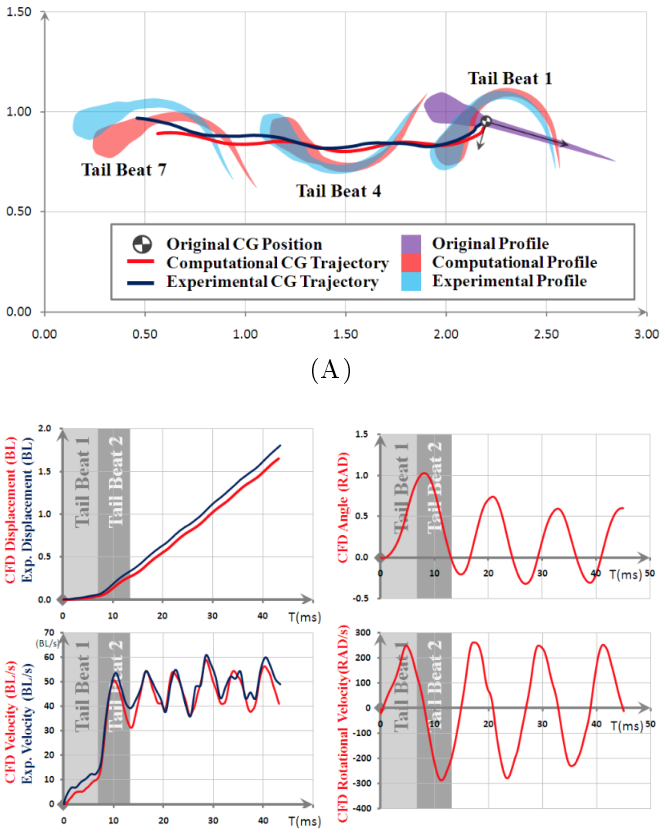
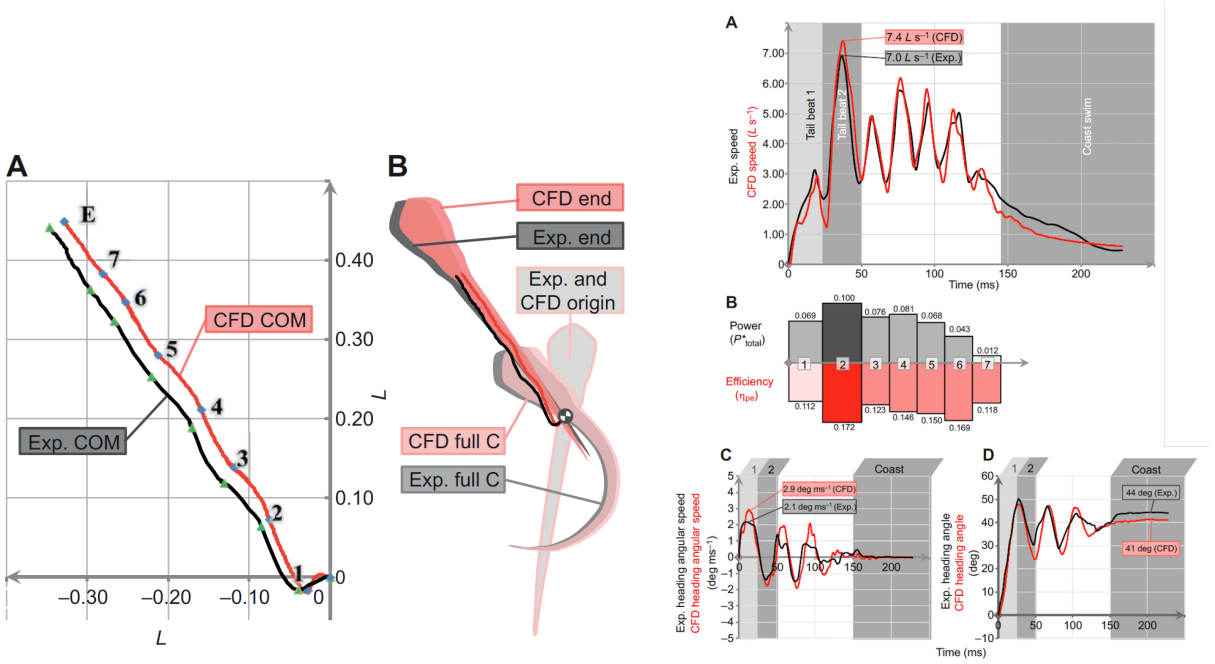


FIGURE 1.12: **A first computational study which validated against experimental kinematics, from [65].** The trajectory which includes seven tail-beats, was particularly well-reproduced (Fig. 5.17 A). Computational kinematics results of the swimming bout (Fig. 5.17 B) include the swimming velocity, displacement, angular velocity and body angle. Note the body rotation could not have been compared with experimental rotation as body deformations were approximated from the heading angle and not from the actual rotation.

knowledge, no similar cell has been constructed although the technique of EFP has previously been performed [111]. Regarding EFP videos, the biologists generally record multiple eleuthero-embryo escape responses for each stimulation and desire to know the response time to the signal, also called latency, the total traveled distance, and the overall displacement of each individual zebrafish eleuthero-embryo. Eventually, other kinematic quantities such as speed or deformation amplitudes (see Fig. 1.3 and Fig. 1.16) are also investigated to enhance the global understanding.

To that end, the team uses a custom software developed and implemented by the group of [22] which is able to track each mass density *i.e.* all individual eleuthero-embryos and compute the overall swimming kinematics and body deformations (see Fig. 1.17). In fact, the software does not track specifically the center of mass of each zebrafish but estimates on each video frame, the point the most central in the eleuthero-embryo's head. Regarding the standard resolution and height of the camera to the cell, the resulting error should be negligible. As for the body deformations, the in-house software fits on each video frame a few straight lines or segments to approximate the midline and compute



(A) (B)

FIGURE 1.13: **An important computational study which compared a C-start simulation against the experiment for seven tail-beats, from [67]**. On the left (Fig. 1.13 A), computational (red) and experimental (black) trajectory of the center of mass (**A**) are compared alongside the silhouettes of experimental (gray) and computational (red) zebrafish, taken at three times: initial, end of C-bend and final time. Note the whole swimming bout is approximately 150 ms long and the C-bend amplitude is not particularly strong. On the right (Fig. 1.13 B), are represented the swimming performances including speed (**A**), power and efficiency (**B**), heading angular speed (**C**) and heading angle (**D**). As a result, velocity and heading angle peaks seem to be associated with tail-beat frequency unlike heading angular velocity. Note no rotation comparison has been conducted as body deformations were approximated from the heading angle and not from the actual rotation. As for power and efficiency, only the average per tail-beat has been computed.

the so-called bend amplitude, supplementary to the head-tail angle which denotes the angle formed by the head segment and the tail segment. We refer the reader to [22] for further details about the tracking software implementation.

As a result, individual kinematics is not always satisfying but the average escape behaviors are well-captured by the in-house tracking software. In addition, occlusions when zebrafish trajectory lines are crossing and eleuthero-embryo's silhouettes are overlapping between each other, are well-handled in most cases. However, tracking methods which provide finer accuracy and better kinematic outputs are particularly interesting for biologists. Above all, any variable or advanced method which would help in understanding the fish behavior or even better, capturing particular locomotor misbehaviors are attractive towards biologists.

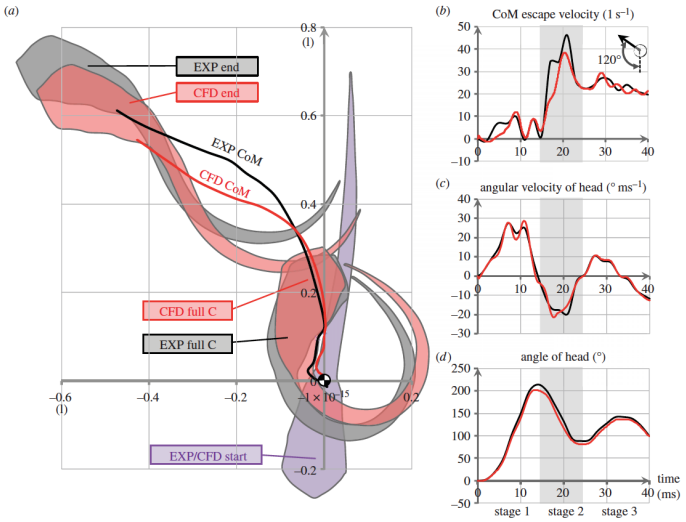


FIGURE 1.14: **A fictitious oriented computational study first validated against few experimental kinematics involving strong C-bend motions, from [68].** The considered swimming bout was formed by three tail-beats only which lasted approximately 35 ms. Overall, the comparison of center-of-mass trajectory (**a**) shows a well-reproduced kinematics although translational and rotation differences might be visible. As for kinematic variables, only the translation velocity (**b**), heading angular velocity (**c**) and heading velocity (**c**) have been compared with no proper comparison of body rotation which might be more important than heading angle.

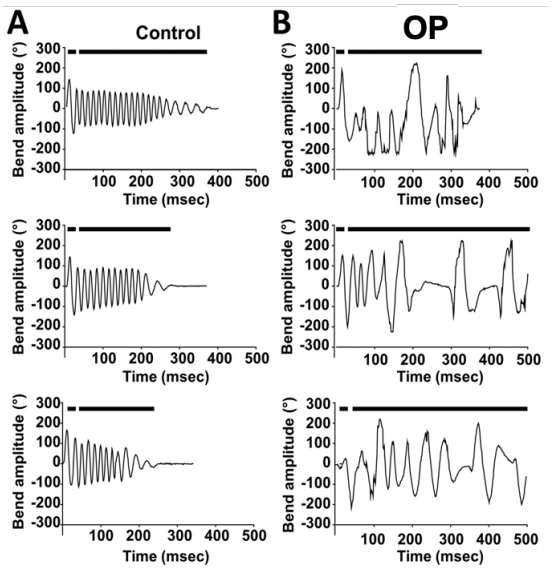


FIGURE 1.15: **Impacts of an organophosphorus toxic molecule on the bend amplitude, from [60].** The OP experiment has been conducted with three replicates (rows) for control (**A**) and exposed (**B**) animals. Note the zebrafish is exposed to the chemical molecule before stimulating and studying the resulting escape response. As a result, the bend amplitude was found significantly impacted especially regarding the tail-beat frequency and its regularity. In addition, the chemical compound effects are still visible after 300 ms while the escape response is completed for control motions.

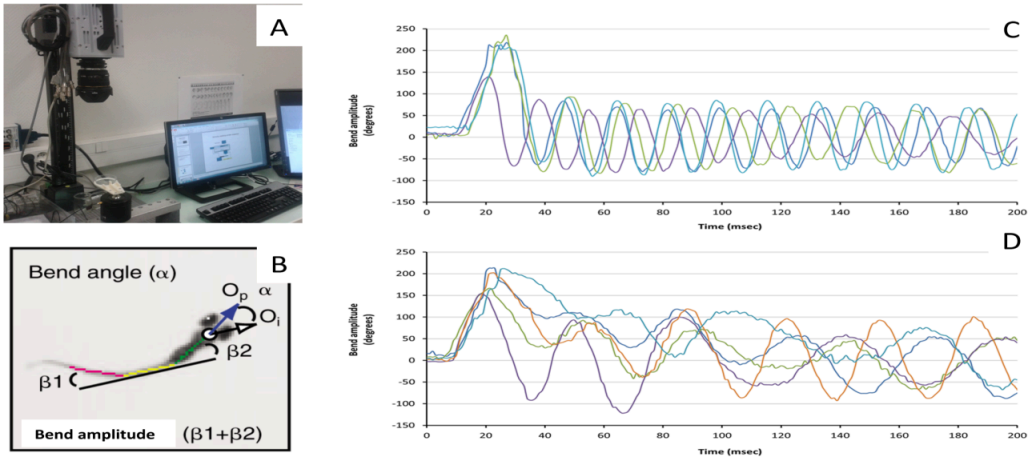
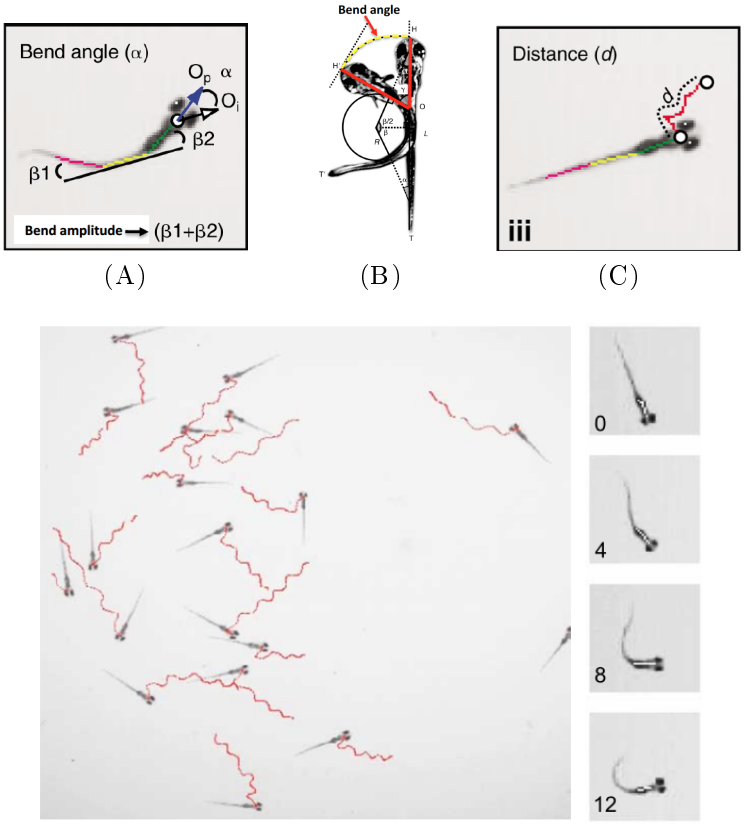


FIGURE 1.16: **The MRGM research team studies the impacts of pharmacological treatments on zebrafish locomotion through high-speed video and standard kinematic analyses.** In particular, the bend amplitude (defined in Fig. (**B**)) can be interesting to compare a standard escape behavior (**C**) and an abnormal escape response (**D**). Figs. (**C**) and (**D**) were obtained with the *FLOTE* tracking software [22].



(A) (B) (C)

(D)

FIGURE 1.17: **Insights of the main outputs of the custom and automated zebrafish tracking software named *FLOTE*, from [22].** Basically, the software tracks individually zebrafish eleuthero-embryos within a group escape response. For each eleuthero-embryo, the software tracks the head trajectory and few midline segments (Fig. 1.17 D), to compute the bend amplitude (Fig. 1.17 A), the heading angle (Fig. 1.17 B) and the traveled distance (Fig. 1.17 C).

1.2.4 The framework of the MEMPHIS team of the University of Bordeaux

In the same time, the MEMPHIS team worked since many years on modeling and simulating fish-like locomotion [8–12, 53]. Across these multiple computational studies, an in-house numerical code has been built for simulating the flow around multiple immersed, rigid, and deforming moving bodies in three-dimensions. The code called NaSCar (NAvier-Stokes CARtesian) is designed specifically to solve complex geometries and movements on a Cartesian grid. As several numerical studies in the literature about self-propelled swimmers, most of the fish-like simulations have been implemented by using a geometry derived from airfoils (Fig. 1.18) and imposing harmonic body kinematics to compute and study fish propulsion and swimming efficiency with steady and periodic motions. However, they recently focused on modeling real-like fish morphologies (Fig. 1.19) and above all, prescribing custom body kinematics using advanced mathematical techniques such as optimal transportation algorithms. The latter implementation combined the development of optimal transportation algorithms [18] with 3D fish reconstructions to extract body deformations directly from experimental imaging.

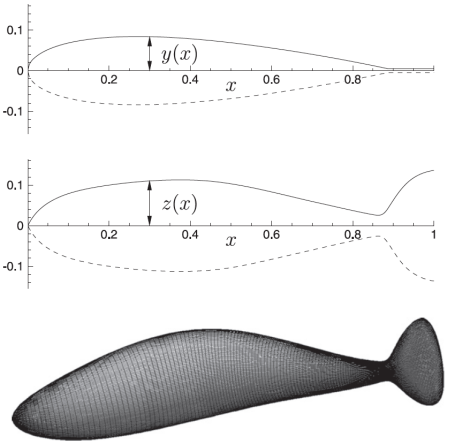
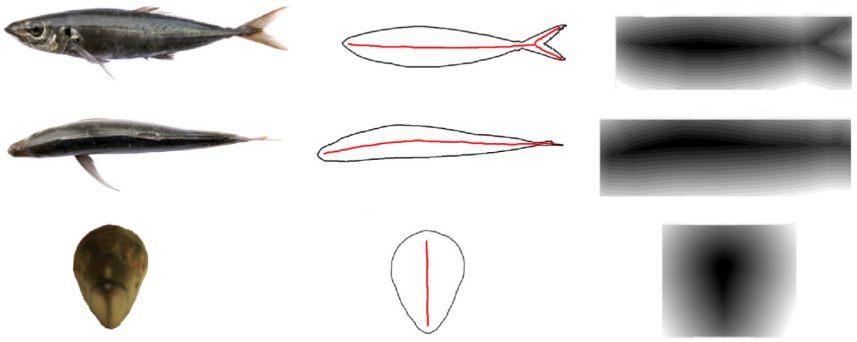


FIGURE 1.18: **Example of a 3D fish modeling from spline formulations, generated by the MEMPHIS team [12].** We can see how fish-like shapes have been derived from airfoil geometry. In addition, the surface mesh used is well-structured with multiple transverse slices along the fish midline.

Thus, both optimal transportation implementations and complex CFD simulations including the developments of numerical schemes and high performance computing (HPC) tools, are part of historical topics of the MEMPHIS research team.

(A) 3D reconstruction.



(B) Views of real photos, midline and level-set.

FIGURE 1.19: **3D reconstruction of a mackerel (from [11]) generated directly from images.** The mackerel (Fig. 1.19 A) is modeled from dorsal and lateral views where the contour and midlines are recovered from the computation of a level-set function (Fig. 1.19 B). The particular image-based procedure assumes the transverse silhouettes across cross-sections do not change significantly and thus can be modeled simply.

1.3 Main objectives

In definitive, we aimed to develop a numerical approach which combines direct numerical simulations and experimental imaging to compute realistic and experiment-driven simulations. Indeed, our primary goal was to drive the numerical simulations with experimental body kinematics to reproduce the escape kinematics of zebrafish eleuthero-embryos, observed in biological experiments. Note the multi-disciplinary approaches might be essential to fully understand the biomechanics involved in complex and biological systems. Actually, the final goal was to provide a numerical tool to biologists to support experiment-based biological studies about locomotion. To that end, biologists particularly desired to use modeling results which are robust and reliable, finer than experimental analysis, and eventually inaccessible *via* the usual tracking methods and kinematic analysis. In particular, mathematical and computational modeling can provide new insights about the swimming dynamics and energetics which drive the resulting kinematics. Besides, those quantities may be key to enlighten the biological mechanisms involved in escape locomotion and detect eventual escape misbehaviors or highlight swimming performance defects. In consequence, biological experiments need to be conducted to validate the numerical results, illustrate the interests of a global experiment-driven numerical approach, and eventually enrich the understanding

of toxicological and pharmacological drug impacts, by performing a large number of numerical simulations. Indeed, experimental kinematics are aimed to be embedded inside CFD simulations, precisely to study and support actual biological experiments.

1.4 Outline of the thesis

On the whole, various objectives need to be answered, from mathematical modeling to biological experiments and its eventual interactions. To that end, this work is divided into four main chapters to extensively expand the methodological process and provide a few biological applications. In fact, each chapter is part of a global development process despite particular disciplines. Indeed, we emphasize all chapters are complementary and essential to each other. For implementing this global approach, several disciplines were involved such as mathematical modeling, high-performance computing, numerical solving, biological experiments, image processing, estimation of experimental body kinematics, 3D modeling, animal modeling, biomechanics. We present in detail the questions raised by each topic throughout the main chapters.

Following this general introduction, the second chapter (Chapter 2) presents the mathematical framework including modeling hypotheses, governing equations, and numerical solving, especially regarding the implemented method to handle and integrate complex solid motions such as moving and highly-deforming zebrafish escape swimming. Essentially, this chapter provides general tools to understand the fundamentals of numerical simulations. In particular, we based our approach on Cartesian grids combined with a volume penalization method and the use of level-set techniques and Lagrangian descriptions of the solid interface. Note the existing CFD code was developed by the MEMPHIS team.

The third chapter (Chapter 3) details the major contributions I developed to embed actual experimental data to the CFD code. Basically, all the major steps which take place between initial experiments and final numerical simulations are presented. Across the thesis, it involved image processing algorithms, optimal transportation theory, *Procrustes Analysis* techniques, 3D modeling such as reconstruction, tracking, and deformations. Notice, each step was important and required investigation and validations.

Afterward, we present the validation of the global experiment-based numerical approach (Chapter 4) which includes validation of the general methodology, validation against experimental kinematics, sensitivity analysis for specific parameters, and the overall reproducibility of experiment-driven numerical results. This chapter is essential to evaluate the robustness, reliability, accuracy, and consistency of our method. In that regard, experimental swimming performances have to be well-reproduced to analyze further energetic quantities.

Therefore, we dedicated the last chapter (Chapter 5) to biological applications which mainly consisted of computing the energetic performances of experimental escape

kinematics. However, we altered the experimental body kinematics to challenge the zebrafish escape response and numerical modeling as well. In fact, two studies are presented: a viscosity study where the surrounding fluid viscosity has been modified, and the impact of toxicological drugs. Nevertheless, fictitious simulations will also be introduced to support in answering biological questions. Eventually, the implementation of effort tests will be introduced to link viscous flows and toxicant exposures, fictitious simulations, and actual swimming performances.

Finally, a general conclusion (Chapter 6) will close this work by recapitulating the main results regarding the model validation or the biological applications, the different methodological steps and the main limitations, and giving few insights about future perspectives.

Chapter 2

Implementation background of CFD numerical simulation

Numerical simulations consist of solving a set of equations which models a complex system from biological processes and physical phenomena, to fluid and structural mechanics, by approximating numerically the considered variables across time and space. The calculation of fluid flows is represented by the so-called CFD. In CFD, the complexity of the flow is characterized by the flow regime which is defined by Re: from steady and laminar states to highly turbulent flows. Direct numerical simulation (DNS) is a particular class of CFD which computes the most accurate solutions, directly from Navier-Stokes equations and with no additional modeling such as turbulence models or average quantities. In this work, we used direct numerical simulations to compute the flow generated by zebrafish swimming *i.e.* a moving and deforming body in water. To that end, we first need to precise the mathematical framework which models the system. Afterward, a computational mesh has to be defined to approximate the solution in a discrete space. Then, specific numerical schemes which represent the approximation algorithms, are considered according to the mathematical modeling and the computational mesh. The last step of the process concerns the actual implementation which often includes algorithmic and parallel programming issues, and the actual computation of the solution on supercomputers. Therefore, we describe in this chapter the general equations which govern the fluid flows and the rigid displacements of an immersed and moving rigid body alongside its discrete counterpart. In this work, we considered the solid body was governed by Newton's laws which could be computed in parallel to the approximation of incompressible Navier-Stokes equations. Besides, we detail how we handled the computational mesh and numerical schemes according to the solid body displacements. Essentially, we used a penalization method combined with a level-set representation and a Cartesian mesh. Finally, we precise how body swimming deformations are taken into account in the whole process, especially with the aim of embedding experimental body kinematics. The in-house finite-difference and Cartesian

solver was developed in the MEMPHIS Inria team by Michel Bergmann and has been validated several times across various studies. That is why we present only a brief overview of the numerical code and refer the reader to [9–12] for further details.

2.1 General description

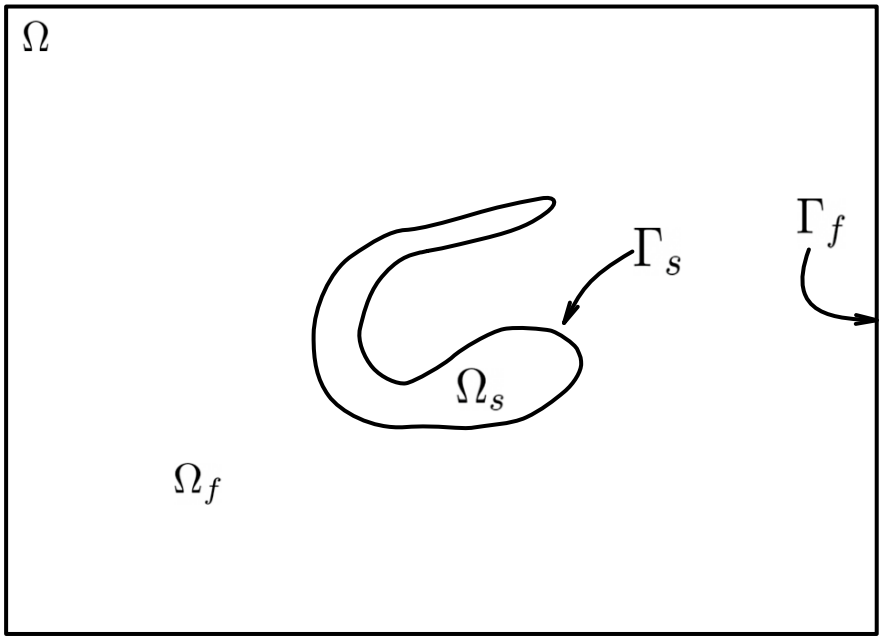


FIGURE 2.1: **Computational domain composed of a fluid domain surrounding an immersed solid body.**

Let Ω be a bounded computational domain and Ω_f and Ω_s be the fluid and solid domains, respectively as illustrated in Fig. 2.1. Thus, the domain Ω is partitioned as $\Omega = \Omega_f \cup \Omega_s$. Let define $\Gamma_f = \partial\Omega$ and $\Gamma_s = \partial\Omega_s$ as the respective fluid and solid boundaries. The general governing equations for the fluid flow are the incompressible Navier-Stokes equations, written as:

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega_f, \tag{2.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} \text{ in } \Omega_f, \tag{2.2}$$

where ρ, μ, $\nu = \mu/\rho$ are the density, the dynamic viscosity and the kinematic viscosity of the fluid, respectively. To close the system, initial and boundary conditions at $\partial\Omega_f = \Gamma_s \cap \Gamma_f$ are imposed. In fact, those equations form a complex system which does not have analytic solutions. Therefore, we need to discretize Ω_f to approximate the solution into a discrete space. We emphasize the boundary condition $\mathbf{u} = \mathbf{u_b}$ imposed at Γ_s is the most important to consider when implementing numerical schemes. Actually, the body boundary condition is crucial as its implementation is precisely what distinguishes most of the numerical methods. Eq. (2.1) describes the mass conservation equation with

the incompressibility condition. Eq. (2.2) is derived from the momentum conservation equation within an Eulerian framework. The left hand side of Eq. (2.2) is constituted of an acceleration term under an Eulerian form which can be written as a time derivative plus a non-linear convection term. As for the right hand side, it represents the main forces which control the flow formed by a pressure gradient and a diffusive viscous term. As a result, both fluid velocity $\mathbf{u}$ and pressure p need to be solved to know the state of the fluid across time.

Note Navier-Stokes equations can also be written under a dimensionless form to describe fluid flows with simple, dimensionless, and characteristic quantities. In that framework, we can analyze and compare flow solutions by considering the flow regime only, regardless of the dimensions of the physical problem for example. Let introduce characteristic quantities such as characteristic velocity U, time T, distance D and pressure P. Let define the associated dimensionless quantities: $\mathbf{u}^* = \mathbf{u}/U$, $t^* = t/T$, $\mathbf{x}^* = \mathbf{x}/D$, $p^* = p/P$. Incompressible and dimensionless Navier-Stokes equations are then written as:

$$\nabla \cdot \mathbf{u}^* = 0 \text{ in } \Omega_f, \tag{2.3}$$

$$\frac{\partial \mathbf{u}^*}{\partial t^*} + (\mathbf{u}^* \cdot \nabla)\mathbf{u}^* = -\nabla p^* + \frac{1}{Re}\Delta \mathbf{u}^* \text{ in } \Omega_f, \tag{2.4}$$

where the so-called Reynolds number Re is a dimensionless number defined as the ratio of inertia forces due to the convection over the viscous terms:

$$Re = \frac{\rho U \times D}{\mu} \tag{2.5}$$

Note in the case of fish-like swimming, we defined Re:

$$Re = \frac{\rho V \times l}{\mu} \tag{2.6}$$

where V denotes the average swimming speed of the fish body and l, the total length of the fish, at rest. The dimensionless form can be used to analyze the different terms in various flow regimes. At low Re, the viscous forces are predominant over the inertial term. On the opposite, highly non-linear effects are present in high Reynolds regimes which leads to turbulence. Re is an indicator of the flow regime ranging from the laminar and viscous flows to turbulent regimes. Zebrafish eleuthero-embryos generally evolve in intermediate flow regimes which correspond to a Reynolds range from $Re = 50$ to $Re = 900$ [89]. In what follows, we preferred the first form with the actual physical dimensions as we aim to simulate the data of actual experiments.

As explained, Navier-Stokes equations require to be discretized on a computational mesh, to be approximated. However, we want to simulate the flow generated around a solid and deformable moving body. Thus, the body movements need to be considered for choosing an adequate mesh combined with a proper representation of the interface. Indeed, the choice of computational mesh and body interface handling is crucial to properly impose

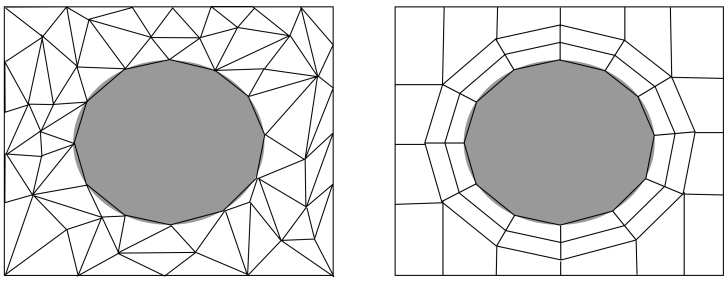
the boundary conditions of the fluid flow within the Navier-Stokes solver. Essentially, the solver needs to know the position of the body in order to adequately enforce the no-slip boundary condition $\mathbf{u} = \mathbf{u_b}$ at the interface. Hereafter, we briefly introduce various interface representations which involve different numerical solvers.

2.2 Interface processing

Before detailing numerical schemes, we present the representation of the body interface we used. As a preamble, we introduce various examples of meshes to handle an immersed body.

2.2.1 Preamble

Basically, the computational mesh is either adjusted to the body or the solid body is immersed within the main computational grid. That is why, two classes of numerical methods can generally be distinguished according to the considered mesh. In particular, the no-slip boundary condition is straightforward to prescribe in case of a so-called body-fitted mesh *i.e.* when mesh nodes are directly located on the fluid-structure interface (Fig. 2.2).

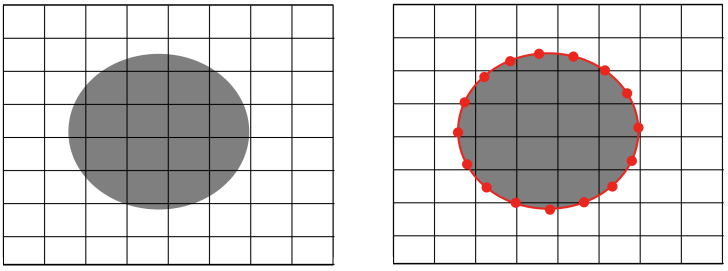


(A) Unstructured. (B) Structured.

FIGURE 2.2: **Representation of a cylinder in 2D, with a body-fitted mesh.** Note the mesh composed of triangular cells (Fig. 5.15 A) or quadrangles (Fig. 5.15 B) can be either unstructured (Fig. 5.15 A) or structured (Fig. 5.15 B).

The mesh is called structured when cells are regularly connected *i.e.* a repetitive pattern is visible while unstructured meshes often generated by automated mesh generators have irregular connectivity. Thus, structured and body-fitted meshes are easy to handle but require specific methods and geometries. Instead, unstructured meshes such as triangular grids are widely used to represent complex geometries. In both cases, the solid interface is well-represented by fluid nodes. However, the numerical method has to be adapted dynamically and its implementation might be difficult. While body-fitted meshes might be convenient to represent and communicate with complex structure boundaries, we can

see the mesh needs to be adjusted dynamically in case of a moving body. In consequence, the node distribution has to be computed again according to the body displacements and such operations are costly and require special attention. Solid bodies are then processed in a completely explicit way by adapting the body-fitted mesh. That is why, body-fitted methods often demand high computational costs, especially in the case of complex geometries and large body deformations such as actual fish body shapes and eleuthero-embryo escape responses.



(A) A Cartesian grid.

(B) Lagrangian markers.

FIGURE 2.3: **Sketches of Cartesian grids on top of cylinder in 2D.** Note the mesh is well-structured but not adapted to the body geometry. Eventually, Lagrangian markers might be computed (Fig. 5.17 B).

As opposed to body-fitted meshes, fixed meshes can be considered to circumvent complex and highly-demanding mesh adaptation. In particular, Cartesian grids which are structured meshes formed by rectangular cells, in 2D are particularly simple to use. However, the body interface is no longer captured by a fluid mesh. On the one hand, different numerical methods have to be considered to enforce the boundary conditions at the interface, in an implicit way. On the other hand, the computational cost required to mesh explicitly complex geometries and remesh large movements, vanishes. Furthermore, multiple bodies can be immersed in the computational mesh with no supplementary cost to numerical methods. The counter-part resides in the tracking of the body and the numerical accuracy reached at the interface. To that end, numerical methods have been developed to capture the interface *via* a level-set function for example, or track the interface *via* Lagrangian markers, for example. Indeed, the position of an immersed solid body is generally handled either with interface-capture or front-tracking methods. Besides, Cartesian structured grids are largely preferred in a massive parallel context regarding load balancing, communications loads, and memory usage for instance. In consequence, no supplementary work is required for simulating various experiments including different animals and body kinematics. In what follows, I will describe an immersed body can be well-captured by level-set methods as our numerical approach is entirely based on Cartesian grids combined with an implicit interface capture. Nevertheless, Lagrangian markers have also been used to represent the body interface and provide alternative and accurate methods to compute particular

quantities involving surface integrals for example. Thus, the Lagrangian representation is detailed afterward.

2.2.2 An Eulerian body representation: a level-set function

Two body-related information is mainly required: its position across time represented by the body characteristic χ_s, alongside the associated body velocity at the boundary $\mathbf{u_b}$. As we use an immersed boundary approach to take into account the body velocity, we also chose to represent the body implicitly *via* a so-called $level-set$ function ϕ, to simply capture complex swimming motions. The essence of a level-set function is to represent the signed distance to the body interface with a constant and normalized gradient $\|\nabla\phi\| = 1$. Therefore, the level-set is positive for all the Cartesian grid nodes located inside the body and negative for the fluid nodes while the level zero corresponds to the fluid-structure interface. This technique was first introduced by Osher and Sethian in 1988 [95] which designed the level-set approach to capture interfaces efficiently. With a level-set approach, the Heaviside function χ_s is then defined according to ϕ: $\chi_s = \mathcal{H}(\phi)$. The solid velocity $\mathbf{u_b}$ is an Eulerian velocity field which will be implicitly recovered when computing the velocity of Cartesian grid nodes located near the boundary, in particular. Therefore, the interface is transported by updating the level-set function ϕ at each time step with the computed velocity field $\mathbf{u}$. The transportation equation which captures the body motions is:

$$\frac{\partial\phi}{\partial t} + (\mathbf{u}\cdot\nabla)\phi = 0 \text{ in } \Omega \tag{2.7}$$

Note the outward normal unit vector can be recovered with a level-set approach with:

$$\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|} \tag{2.8}$$

Level-set techniques are important to capture any complex topology in a straightforward fashion. The main drawback of level-set methods resides in the loss of accuracy near the interface. On the one hand, the solid velocities need to be properly embedded into the computed global velocity $\mathbf{u}$. On the other hand, body volume (or surface in 2D) is not properly conserved across time as the signed distance property is not. This is a well-known property and that is why, the signed distance function should be recomputed from time to time when moving the level-set *via* Eq. (2.7). Several methods exist to generate the level-set field in a discrete mesh. When considering immersed solid bodies, fast-marching techniques have been developed [105, 106] to solve the so-called Eikonal equation :

$$\|\nabla\phi\| = 1 \tag{2.9}$$

Moreover, Russo and Smereka [101] developed an algorithm for solving the distance function, based on the following partial differential equation (PDE):

$$\frac{\partial \phi}{\partial t} = sgn(\phi^0)(1 - \|\nabla \phi\|)\phi(\mathbf{x}, 0) = \phi^0(\mathbf{x}) \tag{2.10}$$

We refer the reader to [101] to find further information about the original numerical implementation. The order of the method can be enhanced with the addition of high-accuracy numerical schemes such as ENO and WENO non-oscillatory method families. In the next chapter, details about the numerical method we used for generating a zebrafish level-set will be provided.

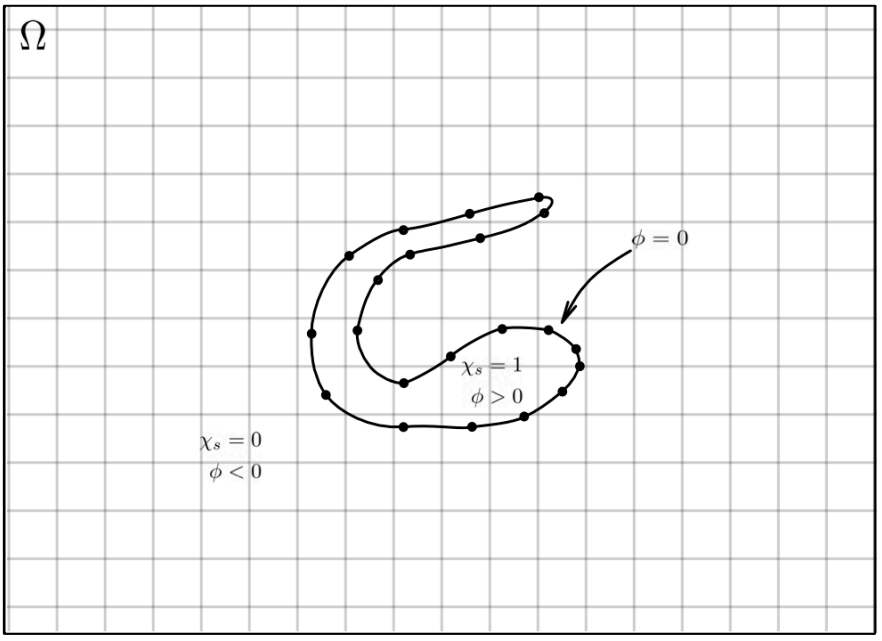


FIGURE 2.4: **Eulerian representation of the immersed solid body alongside Lagrangian markers.**

Note in the case where the immersed body is also meshed with Lagrangian markers in complement to the level-set approach as illustrated in Fig. 2.4, the level-set computation is straightforward as we can directly approximate the distance function at a Cartesian grid node by the minimal distance to the Lagrangian markers.

2.2.3 A Lagrangian body representation: a surface mesh

Alternatively, front-tracking methods use Lagrangian markers for describing the body boundary. A certain amount of discrete points are regularly distributed along the interface to describe its shape and motion. Indeed, the interface can be tracked across time if the velocity $\mathbf{u_b}$ of each Lagrangian point is known. In that case, the solid boundary

is transported *via* the Lagrangian description:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u_b} \tag{2.11}$$

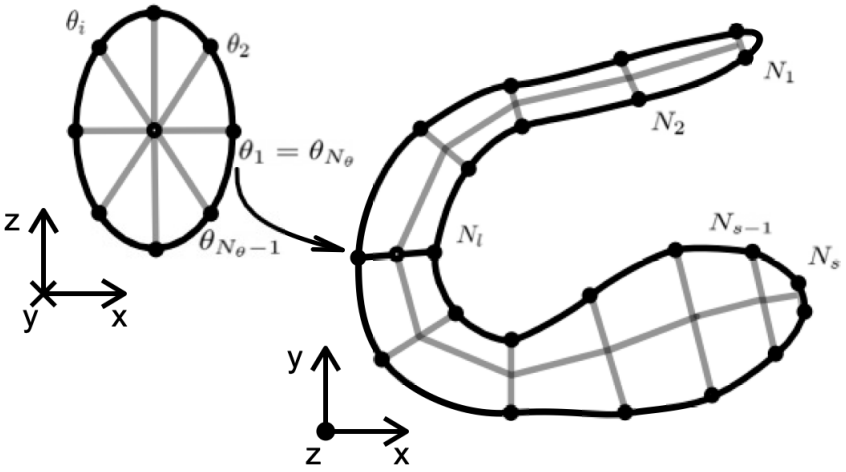


FIGURE 2.5: **Description of an immersed solid body with Lagrangian markers at the surface.** The θ rotation within each slice represents how the indexation is performed in 3D. In 2D, one single loop around the contour is sufficient.

In this work, we discretized the body surface on top of the level-set representation of the immersed body, to effectively track the boundary motions and enable accurate surface computations. We used a structured mesh with N_s transverse slices composed of N_θ points uniformly spread across the contour of each slice (see Fig. 2.5). In that way, each point at the surface can be tracked individually across time and we can use the surface mesh to directly transport the immersed body at each time step with the body kinematics at the surface. Therefore, only the velocity at the surface is required to transport the body with the previously described Lagrangian transportation 2.11. Afterward, the level-set function is then computed to provide the body characteristic required to penalize the velocity $\mathbf{u}$. The construction of a structured mesh which is based on the midline kinematics to orientate each slice adequately as within beam deformation models, ensures each point can be Lagrangian-like tracked across time. In definitive, the surface is meshed with a large number of surface elements formed by $N_s \times N_\theta$ points.

On the whole, the main drawback of fixed Cartesian grids and immersed body boundary, concerns the loss of precision near the interface as the computational grid does not exactly fit the immersed body. That is why several methods have been extensively developed to enhance the order of accuracy near the body boundary either in terms of body representation by sharp Lagrangian tracking or *via* the enforcement of the body boundary condition when solving the fluid governing equations. The latter issues are tackled by the class of immersed boundary methods.

2.2.4 Immersed boundary methods

Hereafter, we briefly present various ways of enforcing the interface boundary condition of the so-called immersed boundary methods (IBM). For further precision, we refer the reader to the complete reviews of IBM [84, 110]. The original IBM term was introduced by Peskin (1972) [99] to simulate biological blood flows but can generally encompass more globally all methods based on fixed Cartesian grids, which enforce the proper velocity of the interface into the fluid solver. In fact, IBM tend to transfer the effects of the immersed body boundary into the governing equations. First, notice governing equations are solved in the whole computational domain including the nodes of the body due to the inherently-immersed boundaries we consider. Thus, computations of interior or near-interface body nodes need to be modified accordingly within an IBM framework. Within the whole class of IBM, we can generally distinguish two approaches: the so-called *continuous forcing* and *discrete forcing* methods. The first modifies the continuous form of governing equations while the latter acts directly on the discretized system.

Discrete forcing approaches

Among the discrete forcing IBM, we can cite two well-known approaches: the so-called ghost-cell and cut-cell methods [84, 110, 134]. Generally, the ghost-cell method takes place within a finite-difference framework to model accurately the location of the body interface inside the Cartesian grid. In particular in the case of a curved interface which crosses Cartesian cells, one can define fluid and solid nodes alongside the so-called ghost nodes which correspond to solid nodes which have at least one fluid node among its neighbors. Ghost nodes then define the near-boundary region where the velocity needs to be corrected by using specific interpolations. Initial ghost-fluid methods used Cartesian interpolations while recent ghost-cells methods founded the interpolation along boundary normal direction [83]. Note in discrete forcing methods, the pressure boundary condition also needs to be imposed which is not required for continuous approaches. As for cut-cell methods [134], the goal is to provide a conservative approach by using control volumes formed by rectangular and trapezoidal cells defined by the crossing boundary. Indeed, we estimate each near-boundary cell is divided into two trapezoidal parts by the interface. Therefore, cut-cell methods are well-adapted for finite-volume schemes where flux have to be computed in each cell. In the case of near-boundary cells, the numerical flux are based on the control volumes. Notice for all discrete forcing methods, sharp interfaces are considered which requires proper tracking of the interface. Instead, continuous forcing methods may take into account the body boundary *via* sharp or diffuse interfaces.

Continuous forcing approaches

Actually, the original IBM developed by Peskin (1981) enforces a localized forcing term in the momentum equation for an elastic boundary, in a diffuse narrow-band near the interface while tracking the elastic interface in a Lagrangian way [84]. This method has then been extended to rigid interfaces by considering a spring model with an important stiffness. Secondly, more general approaches which are part of fictitious domain methods model the body regions as fluid regions with rigidity properties [46] while another class of IBM views the whole fluid region as a porous media. The latter encompasses the boundary condition through the addition of a Brinkman penalty term into the momentum equation. These methods denote the so-called penalization approaches [3]. On the whole, many continuous forcing approaches consider a diffusive interface. Finally, other continuous forcing methods do conserve the sharp-interface property by using a hybrid staggered-non-staggered approach (see [45, 110]).

As explained previously, in this work we considered a fixed Cartesian grid which we combined with a level-set approach, the tracking of Lagrangian markers, and the implementation of an IBM. Historically, in the region of Bordeaux and particularly in the Institute of Mathematics of Bordeaux (IMB), several research groups have developed, implemented, and extensively worked with penalization methods in a CFD framework [3]. The volume penalization method computes identical equations for both solid and fluid regions but penalizes the body region nodes accordingly. More details about the penalization method are provided in the next section.

2.3 Penalization method

In this section, we detail the penalized model of the incompressible equations of Navier-Stokes before unveiling the actual discretization and the numerical schemes that we mainly used in this work for the numerical resolution.

2.3.1 Penalized equations

As explained previously, volume penalization methods have been developed to take into account the effects of body velocity in the framework of a moving and deforming body, immersed inside a fixed and Cartesian computational mesh. Thus, in the penalization method, the numerical approach is monolithic: we see the whole domain as one single phase, viewed as a porous media in which the permeability is considerably lower inside the immersed body. The previous incompressible Navier-Stokes equations 2.1-2.2 are penalized as:

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega, \tag{2.12}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \mathbf{u} + \chi_s \lambda(\mathbf{u_b} - \mathbf{u}) \text{ in } \Omega \tag{2.13}$$

where χ_s and λ are the body characteristic function and the penalty coefficient, respectively. The body characteristic denotes the Heaviside function $\chi_s = 1$ in Ω_s, 0 otherwise. The penalty parameter λ we imposed to 10^8 can be seen as the inverse of a very low permeability coefficient. Thus, the boundary condition $\mathbf{u} = \mathbf{u_b}$ at Γ_s is implicitly imposed through the penalization term $\chi_s \lambda(\mathbf{u_b} - \mathbf{u})$. This penalized form of incompressible Navier-Stokes equations is equivalent to the initial system with a no-slip condition at the interface, in the limit $\lambda \to \infty$ [3]. Besides, the penalized system converges towards the initial Navier-Stokes system with a convergence rate in $\mathcal{O}(\sqrt{1/\lambda})$ [56].

As for the Γ_f-boundary conditions, we used a mix of wall and free-flow conditions, detailed in chapter 4. In our problem, the body and the fluid are initially at rest and no fluid flow is imposed through the domain boundaries. The previous penalized equations 2.12-2.13 are then discretized with finite-difference schemes described underneath. Basically, the numerical solution of the finest meshes tends to the real solution in the limit of numerical approximations. The purpose of numerical schemes is to converge to the proper solution with a certain order of accuracy while being numerically stable.

2.3.2 Discretization

Incompressible penalized Navier-Stokes equations have been discretized by considering a cell-centered and collocated arrangement of the primitive variables $(\mathbf{u}, p)$. We can see there is no explicit equation for solving pressure. In fact, pressure can be seen as a Lagrange multiplier for the incompressibility constraint. Therefore, the pressure is implicitly governed by the continuity equation 2.12. Historically, numerical approaches consisted of fractional-step schemes to process $\mathbf{u}$ and p. For instance, Semi-Implicit Method for Pressure Linked Equations (SIMPLE [97], [84]) conserved a pressure term for solving the momentum equation before applying a pressure correction. Instead, we preferred the so-called Chorin-Temam predictor-corrector scheme [26, 116] which consists of decoupling pressure and velocity by solving the momentum equation for the velocity and a Poisson equation for the pressure before correcting the velocity to satisfy the incompressibility constraint. Due to the continuity equation, no explicit pressure boundary conditions are required when solving the Poisson equation.

Let develop the so-called Chorin-Temam scheme which updates the primitive variables $(\mathbf{u}, p)$ from the previous time step, in multiple steps. To that end, let introduce intermediate variables $\mathbf{u}^*$, $\mathbf{u}^{**}$ and p^*. Let define the space-discretized convective and diffusive terms as: $C = (\mathbf{u} \cdot \nabla)\mathbf{u}$ and $D = \Delta \mathbf{u}$. The diffusive term has been discretized in time either explicitly or implicitly *via* a Crank-Nicholson scheme while a second-order accuracy Adams–Bashforth fractional-step scheme was used for the convective term. Indeed, an implicit implementation may be useful to circumvent the viscous stability

condition in case of highly viscous flow regimes while an explicit scheme is required for the convective term due to its non-linearity. As for the global temporal scheme, we used a standard first-order discretization. Finally, the penalty term is described implicitly by using a second-order ($\mathbf{u_{b_2}}$) or first-order ($\mathbf{u_{b_1}}$) accuracy for boundary conditions at the interface Γ_s. As a result, the previous penalized momentum equation 2.13 was discretized in time as:

$$\begin{aligned}\frac{\mathbf{u^{n+1}}-\mathbf{u^n}}{\Delta t} = &-(\frac{3}{2}C^n - \frac{1}{2}C^{n-1}) \\ &- \frac{1}{\rho}\nabla p^{n+1} \\ &+ \frac{\mu}{2\rho}(D^{n+1}+D^n) \\ &+ \chi_s^{n+1}\lambda(\mathbf{u_{b_2}^{n+1}} - \mathbf{u^{n+1}})\end{aligned}$$

Intermediate variables $\mathbf{u}^{**}$, $\mathbf{u}^{*}$, p^{*} are then introduced in the previous equation as:

$$\begin{aligned}\frac{\mathbf{u^{n+1}}-\mathbf{u^{**}}}{\Delta t} + \frac{\mathbf{u^{**}}-\mathbf{u^{*}}}{\Delta t} + \frac{\mathbf{u^{*}}-\mathbf{u^{n}}}{\Delta t} = &-(\frac{3}{2}C^n - \frac{1}{2}C^{n-1}) \\ &- \frac{1}{\rho}\nabla(p^{n+1} - p^n + p^n) \\ &+ \frac{\mu}{2\rho}(D^{*}+D^n) \\ &+ \chi_s^{n+1}\lambda(\mathbf{u_{b_2}^{n+1}} - \mathbf{u^{n+1}})\end{aligned}$$

In consequence, each term is incorporated within a solving step. First, we solve $\mathbf{u}^*$ during the *projection* step:

$$\frac{\mathbf{u_i^*}-\mathbf{u_i^n}}{\Delta t} = -(\frac{3}{2}C_i^n - \frac{1}{2}C_i^{n-1}) - \frac{1}{\rho}(\nabla p^n)_i + \frac{\mu}{2\rho}(D_i^* + D_i^n) + \chi_s^{n+1}\lambda(\mathbf{u_{b_{1,i}}} - \mathbf{u_i^*}) \quad (2.14)$$

where the convective $C_i = \sum_j \frac{\partial u_j u_i}{\partial x_j}$ and diffusive $D_i = \sum_j \frac{\partial^2 u_i}{\partial^2 x_j}$ terms are discretized in space with a third-order upwind and a second-order central finite difference schemes, respectively. Note we preferred upwind schemes for the convection term to have better stability than centered finite-difference schemes. At the end of this first step, we obtain the intermediate velocity field $\mathbf{u}^*$ which is *a priori* not divergence free and the pressure term is not properly defined. That is why, we need to perform a *correction* step by computing Eq. (2.12). Therefore, we want to find the pressure p^{n+1} such that $\nabla\cdot\mathbf{u}^{**} = 0$ using the equations:

$$\frac{\mathbf{u^{**}}-\mathbf{u^{*}}}{\Delta t} = -\frac{1}{\rho}(\nabla p^{n+1} - \nabla p^n) \quad (2.15)$$

$$\nabla\cdot\mathbf{u}^{**} = 0 \quad (2.16)$$

The divergence of Eq. (2.15) gives :

$$\nabla\cdot\mathbf{u}^{*} = \frac{\Delta t}{\rho}\Delta(p^{n+1} - p^n) \quad (2.17)$$

which provides the following Poisson equation :

$$\nabla \cdot \mathbf{u}^* = \frac{1}{\rho}\Delta\Phi \tag{2.18}$$

with $\Phi = \Delta t(p^{n+1} - p^n)$. The considered boundary conditions in $\partial\Omega$ for the Poisson problem are homogeneous Neumann conditions for the pressure in order to not impact the computed velocity during this *projection* step. Note a face-centered velocity U^* is required in 2.18 instead of the collocated variables:

$$\nabla \cdot \mathbf{U}^* = \frac{1}{\rho}\Delta\Phi \tag{2.19}$$

In that way, both velocities and Φ are face-centered and we limit the occurrence of spurious pressure oscillations due to pressure and velocity locations. The face-centered velocity needs to be computed with the proper corrections of pressure gradients regarding face-centered (fc) and cell-centered (cc) locations:

$$\mathbf{U}^* = \mathbf{u}^*{}_{fc} + \Delta t((\nabla p^n)_{cc} - (\nabla p^n)_{fc}) \tag{2.20}$$

The linear system formed by the discrete equations 2.19 is solved using iterative and Krylov subspace methods such as Conjugate Gradient (CG) or BiConjugate Gradient Stabilized (BCGS) methods. After this *projection* step, the pressure and velocity are then corrected as:

$$p^{n+1} = p^n + \frac{\Phi}{\Delta t} \tag{2.21}$$

$$\mathbf{u}^{**} = \mathbf{u}^* - \frac{1}{\rho}\nabla\Phi \tag{2.22}$$

Let assume the body velocity at the boundary $\mathbf{u_b^n}$ is known at time t^n. The last step regarding the fluid solver consists of transporting χ_s^n to χ_s^{n+1} and penalizing the velocities located near the body surface Γ_s to reach a second-order accuracy at the interface. The major issue with IBM is the proper approximation of the immersed interface *i.e.* proper fluid velocities near the interface. With no special correction, the solid velocity is enforced at Cartesian grid nodes which is only first-order accuracy in space as we do not model curved interfaces adequately. In what follows, I will only briefly present the image point correction (IPC) algorithm we used to recover a second-order accuracy near the interface. We refer the reader to [9, 53] for further details.

The computed velocity of near-boundary nodes also called the ghost cell (gc) is penalized with the symmetric point to the interface (see Fig. 2.6). Basically, we penalize each ghost node using the symmetrical value relatively to the interface and located inside the immersed body:

$$\mathbf{u_{b_2}} = \mathbf{u_{b_1}} - \phi(\frac{\partial \mathbf{u}}{\partial \mathbf{n}})|_{\phi=0} \tag{2.23}$$

$$\frac{\mathbf{u^{n+1}} - \mathbf{u}^{*}*}{\Delta t} = \chi_{gc}\lambda(\mathbf{u_{b_2}} - \mathbf{u^{n+1}}) \tag{2.24}$$

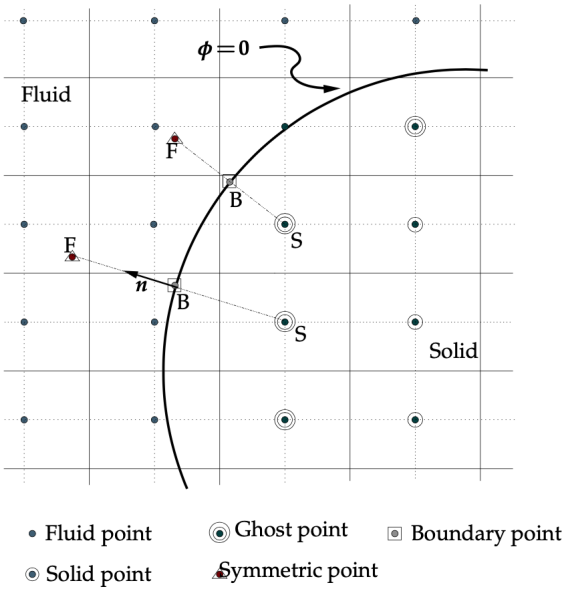


FIGURE 2.6: **Sketch of the IPC method in a two-dimensional case, from [9].** The IPC is applied to the Ghost points using the symmetric points. The value of the symmetric point is based on an interpolation of its four neighboring fluid points. The original penalty method is applied on the other solid points. Figure and caption extracted from [9].

where $\mathbf{u}^{**}$ is the velocity computed at the end of the correction step which we penalize with the IPC method. Note we use the level-set function to recover the distance to the interface and the outward normal unit vector. After this correction step, all the collocated variables $\mathbf{u}$ and p are updated to time $t^{n+1} = t^n + \Delta t$. Notice the time step Δt can be variable according to stability conditions determined with the fluid velocity. Finally, we described how the fluid velocities are computed and updated in each time step when the body velocity of the previous time iteration is known alongside the position of the solid body boundary Γ_s. Note the body boundary velocity field $\mathbf{u_b^n}$ which is also processed with an Eulerian point of view, constitutes one of the remaining unknown for updating the fluid velocity from u^n to u^{n+1}, alongside its position *via* the body characteristic χ_s^n. Insights about body computations are provided in the next section.

2.4 Body dynamics

2.4.1 A deforming body: decomposition of body velocity

In general, I previously introduced the immersed body as a moving and deforming solid object. Indeed, we did not model the body boundary as an elastic interface. In fact, most of the time, we described the immersed body as solid and rigid. Nevertheless, we remind the reader the goal is to study fish-like swimming and more precisely, experimental zebrafish escape swimming. For self-propelled bodies, rigid kinematics which is generally issued from Newton's laws cannot be sufficient to describe an auto-propelled motion.

Indeed, in that case, we do not describe the motion of a rigid body which undergoes the surroundings state. Instead, we study the movements which are generated by the body itself. That is why, supplementary information is required to represent the body deformations. For a moving solid object, the body velocity at the surface $\mathbf{u_b}$ is composed of a translation velocity $\bar{\mathbf{u}} = \mathbf{u_{CoM}}$ and a rotation velocity $\mathbf{u}^\theta$. Considering a solid object, the latter is given by $\mathbf{u}^\theta = \omega \wedge \mathbf{r}$, ω being the angular velocity and $\mathbf{r} = \mathbf{x} - \mathbf{x_{CoM}}$ is the distance to the center of mass. As for a deforming body, we assumed the body velocity can be decomposed into the rigid velocities and an additional deformation velocity denoted $\mathbf{u}_\epsilon$:

$$\mathbf{u_b} = \mathbf{u_{rigid}} + \mathbf{u}_\epsilon = \bar{\mathbf{u}} + \mathbf{u}^\theta + \mathbf{u}_\epsilon \tag{2.25}$$

The rigid terms are determined by the solid dynamics and the computation is described in what follows. As for the deformation velocity, the information is generally prescribed by imposing movements from Lagrangian markers or midline kinematics and curvature. For example, harmonic formulations can constitute an easy and simple way to implement fish-like kinematics, especially for modeling a steady, cyclic and in-line motion. Herein, the essence of this work resides in the computation of both the deformation term $\mathbf{u}_\epsilon$ and the body shape. In particular, this work differs from the literature by computing the body deformations from experimental imaging. Indeed, the choice of $\mathbf{u}_\epsilon$ is crucial to obtain realistic body kinematics and that is why, we preferred to directly use experimental observations. To that end, experimental videos were first post-processed and segmented to obtain the 2D body shape at each experimental frame. Afterward, a *Procrustes Analysis* step was required to realign each frame and subtract any rotation between consecutive frames. The *Procrustes Analysis* aimed at recovering and isolating the deformation motions. Besides, we investigated various techniques to extract the deformation velocity from experimental data and recover the body characteristic. These major contributions developed for both 2D and 3D simulations, will be the focus of the next chapter. Actually, we extensively used two main methods which either compute the experimental deformation velocity field $\mathbf{u}_\epsilon$ or provide surface Lagrangian markers which enable to track the body kinematics through experimental imaging. That is why, the discretization and computations of body dynamics may differ according to the Eulerian or Lagrangian point of view.

2.4.2 Rigid body dynamics, forces and torques

The rigid body kinematics can directly be computed from Newton's laws - the law of conservation of momentum for solid objects. Indeed, the translation and rotation velocities of the center of mass are recovered with the solid mechanics equations :

$$\rho_s V_s \frac{\mathrm{d}\bar{\mathbf{u}}}{\mathrm{d}t} = \mathcal{F}^{\mathbf{hydro}} \tag{2.26}$$

$$\frac{\mathrm{d}J\omega}{\mathrm{d}t} = \mathcal{M}^{\mathbf{hydro}} \tag{2.27}$$

where $\rho_s = \rho$ is the solid mass density, V_s the solid volume, J is the inertia matrix and $\mathcal{F}^{\mathbf{hydro}}, \mathcal{M}^{\mathbf{hydro}}$ the hydrodynamic forces and torques exerted onto the solid body. The hydrodynamic forces and torques are computed by integrating the hydrodynamic tensor $\mathbb{T}$ over the surface:

$$\mathbb{T}(\mathbf{u}, p) = -p\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \tag{2.28}$$

$$\mathcal{F}^{\mathbf{hydro}} = -\oint_{\Gamma_s} \mathbb{T}(\mathbf{u}, p) \cdot \mathbf{n}\mathrm{dS} \tag{2.29}$$

$$\mathcal{M}^{\mathbf{hydro}} = -\oint_{\Gamma_s} \mathbb{T}(\mathbf{u}, p) \wedge \mathbf{r}\mathrm{dS} \tag{2.30}$$

Afterward, the numerical calculation of hydrodynamic terms may differ according to the representation of the body. Indeed, the body surface is not well-described when using implicit and fully Eulerian approaches such as level-set techniques while Lagrangian markers might precisely be computed to discretize and directly integrate surface integrals. Both paradigms are then enlightened in what follows.

2.4.3 Power

The instant consumed power due to hydrodynamic forces is simply derived from the computed boundary velocity $\mathbf{u_b}$ and the hydrodynamic tensor $\mathbb{T}$ based on fluid variables $(\mathbf{u}, p)$ as:

$$P = \int_{\Gamma_s} (-\oint_{\mathrm{dS}} \mathbb{T}(\mathbf{u}, p) \cdot \mathbf{n}\delta\mathrm{S}) \cdot \mathbf{u_b}\mathrm{dS} \tag{2.31}$$

where $\mathbf{n}$ denotes the outward unit vector to the body surface. Note the torques and angular velocity are implicitly introduced within this formulation as $\mathbf{u}^\theta$ is present in $\mathbf{u_b}$. Similarly, the deformation velocity also contributes to the generation of hydrodynamic forces and torques *via* the penalty term and thus, is taken into account in the power output calculations. As a result, instant power quantifies the power required by the zebrafish body to overcome the surroundings forces and move its body through. As noticed, this power incorporates both the contributions of passive and active parts represented by rigid and deformation kinematics, respectively. Therefore, it is important to note that the power output cannot exactly neither well-represent the available internal biological power nor characterize the power instantly consumed by the muscles and locomotor activity of the zebrafish animal. However, the whole motion is originated from the production of body kinematics only provided no exterior source nurture the fluid flow motion and the body swimming movements. In consequence, the consumed power does effectively represent the total power required by zebrafish eleuthero-embryos to swim. Furthermore, the total work expended between time t_i and time t_f can be

derived by integrating the instant power over time:

$$E = \mathcal{W}_{total} = \int_{t=t_i}^{t_f} P\mathrm{d}t \approx \sum_{t=t_i}^{t_f} P_i \mathrm{d}t \tag{2.32}$$

Note the continuous quantity P is described numerically by the time-discrete power P_i. Basically, the hydrodynamic forces represent the forces exerted by the surrounding fluid onto the immersed body. The output power is the power the body has to expend in order to move through the fluid forces at time t. The work is the consumed power over time while expended energy E is the total amount of work. Generally, the total amount of work which represents the total expended energy, can be particularly important to analyze the energetic efficiency of a system by separating the useful work from the rest. Thus, the classic Froude efficiency which is defined as:

$$\eta_{Fr} = \frac{\mathcal{W}_{useful}}{\mathcal{W}_{total}} \tag{2.33}$$

can also be computed in the case of self-propelled swimmers by determining the parts of the work which contribute the most to propulsion for example. In the case of simple deformation laws such as harmonic traveling waves, the useful work is formed by forces which drive the body forward, associated with the forward velocity. Given the straightforward definition, simple body kinematics was often imposed to study swimming performances. However, only the global amount of work can be computed in case of complex body deformations such as zebrafish escape responses as we cannot generally distinguish the most important contributions to escape swimming performances. Therefore, alternative energetic quantities need to be defined to assess the escape swimming performance. As fishes produce body kinematics to travel, one can also define the cost of transport (CoT) which corresponds to the energy expended per distance unit:

$$CoT = \frac{E}{d_{total}} \tag{2.34}$$

where d_{total} is the total traveled distance between time t_i and time t_f in the sense of the total sum of infinitesimal distances traveled across time and not the raw Euclidean displacement between final and initial time. The cost of transport is particularly attractive for escape kinematics as it measures the energetics invested by zebrafish to travel its center of mass over a certain distance. Thus, the CoT is minimized for large traveled distances or low energetic consumption. Notice, the escape swimming speed may be required to be analyzed alongside CoT to provide a finer understanding of escape swimming performances.

Besides, we defined the flow vorticity as the quantitative variable which measures the rotation of the flow to enlighten where the vortices are located in the wake. Thus, the fluid vorticity is given by:

$$\mathbf{\Omega} = \nabla \wedge \mathbf{u} \tag{2.35}$$

Finally, let define the Q-criterion as :

$$Q = \frac{1}{2}(\|\Omega^2\| - \|S^2\|) \tag{2.36}$$

where Ω and S are the symmetric and asymmetric parts of the velocity gradient, respectively which also represent the vorticity and rate of strain tensors. The Q-criterion is often preferred to the vorticity when illustrating the generated wake behind a self-propelled swimmer.

As we presented the general governing equations alongside important energetic quantities for fish-like swimming, we detail in what follows the different calculations performed for both Eulerian and Lagrangian points of view.

2.4.4 Swimming body dynamics: a fully Eulerian framework

Hereafter, we present how body dynamics are computed at each time step, when the state of the fluid flow is known and only an *Eulerian* representation is used through the level-set function. Again, we remind the reader the body deformation velocity field $\mathbf{u}_\epsilon$ which is used to recover $\mathbf{u_b}$, is extracted from experimental imaging data and more precisely, locomotion videos. Therefore, we denoted $\Delta_i t$ the acquisition time between two consecutive experimental frames and $\delta_i t$ the time between the current computational time and the last experimental image.

As explained previously, the rigid velocity composed of the translation and rotation motions, is computed at each grid nodes within the body *i.e.* where the level-set function ϕ is positive. In addition, a global deformation velocity must be computed. Basically, we know the initial and final density distributions ρ_0 and ρ_1 and we need the deformation velocity field to transport ρ_0 to ρ_1 during the time step $\Delta_i t$ (see Fig. 2.7). This specific requirement is a key approach which was originally initiated in 2D to extract the deformation velocity from experimental video frames. The numerical solution is the optimal transportation and will be the focus of an entire section in the next chapter. Let assume $\mathbf{u}_\epsilon$ is known. At each time step, the level-set function needs to be transported to its proper position regarding the rigid and deformation velocity of the body.

Transportation of the body

For the deformation velocity, the level-set is actually implicitly deformed by using the experimental video frames. Nevertheless, computational time and acquisition time rarely correspond. That is why, the deformation velocity is also used to interpolate intermediate snapshots of the level-set. In this work, level-set snapshots are interpolated by optimal transportation (Fig. 2.7). Afterward, the level-set function needs to be transported according to the rigid velocity. In definitive, the level-set function is updated from the

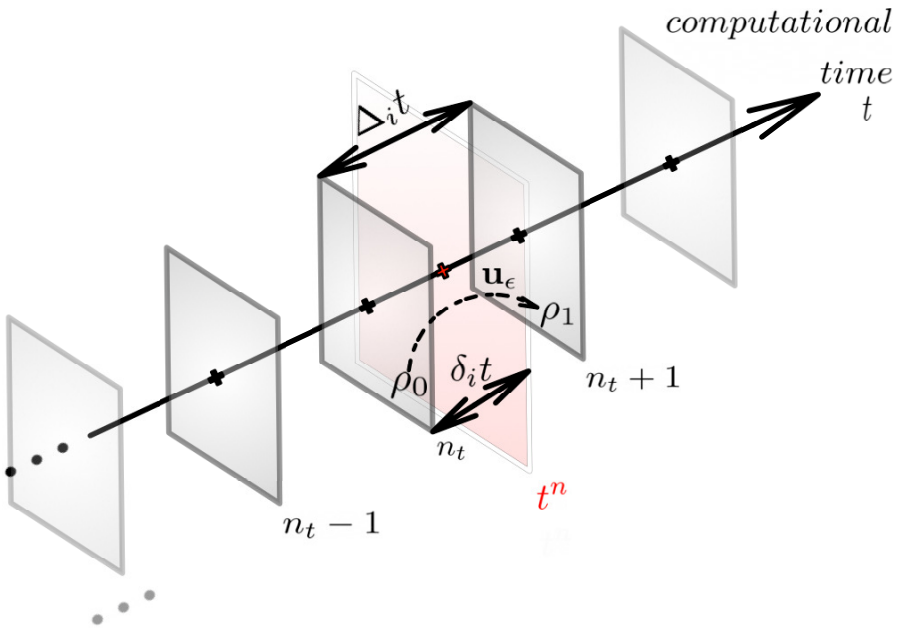


FIGURE 2.7: **Recovering the body shape at each time step from experimental frames of level-set fields.** N_t video frames are available. Each computational time step is located between two experimental frames n_t and $n_t + 1$. With an Eulerian representation, each frame is a level-set function of the body which describes the mass density distribution ρ. We denotes ρ_0 and ρ_1 the two density distributions at n_t and $n_t + 1$. The deformation velocity field $\mathbf{u}_\epsilon$ defines the transportation of ρ_0 to ρ_1. Note the index i refers to experimental images.

previous time step by updating its shape from the deformation velocity field, and its location by rigid transportation.

The level-set function can be transported by solving the advection equation 2.7. To circumvent the issues of volume conservation explained previously, we used a Lagrangian point of view to transport the level-set function of each Cartesian grid node adequately (Fig. 2.8):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \frac{\mathbf{x_i^{n+1}} - \mathbf{x_i^n}}{\mathrm{d}t} = \mathbf{u_i} \tag{2.37}$$

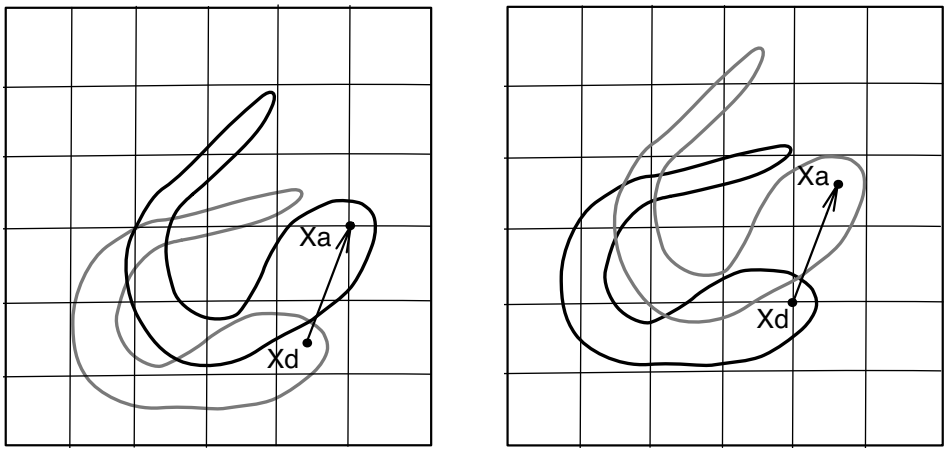
which are derived from the Lagrangian form of the transportation equation 2.7:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = 0 \tag{2.38}$$

Notice a grid node is rarely transported to another grid location. That is why, Eq. 2.37 is actually written as:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \frac{\mathbf{x_a} - \mathbf{x_d}}{\mathrm{d}t} = \mathbf{u_i} \tag{2.39}$$

where $\mathbf{x_d}$, $\mathbf{x_a}$ denote the departure and arrival locations, respectively. Thus, the grid node associated with the rigid velocity $\mathbf{u_i}$ can be viewed either as the departure (semi-Lagrangian) or the arrival point (Lagrangian) (see Fig. 2.8). The numerical



(A) Lagrangian scheme.

(B) Semi-Lagrangian scheme.

FIGURE 2.8: **Sketches of Lagrangian and semi-Lagrangian transportation schemes, with Cartesian grids.** For a Lagrangian scheme (Fig. 2.8 A), grid nodes are at arrival locations while the departure point is considered at the grid node for a semi-Lagrangian transportation (Fig. 2.8 B). In both cases, we represented in black the silhouette associated to the Cartesian grid; previous or next contour being in gray.

scheme is then defined according to the adopted point of view. In the Lagrangian transportation case, a direct solution consists of finding the original particle location at time t^n: $\mathbf{x_d} = \mathbf{x_a} - \mathrm{d}t\mathbf{u_i^n}$ before interpolating the signed distance *i.e.* level-set function and the deformation velocity from the nodes surrounding $\mathbf{x_d}$. Indeed, the deformation velocity field also needs to be transported to follow its associated level-set function and deform it in the next time step. Thus, level-set and deformation properties of $\mathbf{x_a} = \mathbf{x}_i^{n+1}$ are directly issued from interpolation at $\mathbf{x_d}$. However, we may encounter troubles for singular cases where a fluid node becomes a solid node. Indeed, in this particular case, the deformation velocity is not well-defined. To overcome such numerical issues, we implemented a semi-Lagrangian scheme coupled with a remeshing particle advection method [132]. The semi-Lagrangian method consists of performing the opposite trajectory: we transport the particle located at $\mathbf{x_d} = \mathbf{x_i}^n$ to its new position at time t^{n+1}: $\mathbf{x_a} = \mathbf{x_i}^n + \mathrm{d}t\mathbf{u_i}$ before re-distributing the transported properties of the particle onto the Cartesian grid. In that way, we can transport both the signed distance function and the deformation velocity of each grid node.

Remeshing particle methods have been developed to solve Euler equations with Lagrangian fluid particles and recover the global variables such as density and momentum onto a Cartesian grid. The purpose is to conserve the fluid particle properties while the particles move across the Cartesian mesh. Once transported, the particle is remeshed within a certain zone of influence as illustrated in Fig. 2.9. The distance function is remeshed by using interpolation kernels which can be found in [132].

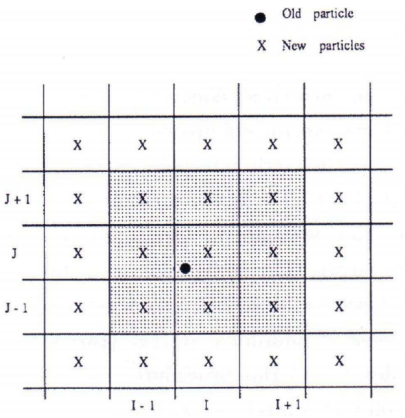


FIGURE 2.9: **After a Lagrange-like transportation step, the particle named "old particle" is remeshed onto the Cartesian mesh, from the *Lisl Weynans' PhD* [132].** The particle quantities such mass, momentum and energy participate to the "new particle" calculations.

Discretizations

As the computational mesh considered herein is not body-fitted and the body surface cannot be approximated by an implicit representation, we computed hydrodynamic forces and torques *via* alternative formulations based on a dynamic control volume $\Omega_f(t)$ surrounding the immersed body, which are actually equivalent to Eqs. 2.29-2.30:

$$\mathcal{F} = -\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_f(t)} \mathbf{u}\mathrm{d}V + \int_{\partial\Omega_f(t)} (\mathbb{T} + (\mathbf{u}-\mathbf{u_f})\otimes\mathbf{u})\mathbf{n}\mathrm{d}S - \int_{\partial\Omega_s} ((\mathbf{u}-\mathbf{u_b})\otimes\mathbf{u})\mathbf{n}\mathrm{d}S$$

$$\mathcal{M} = -\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_f(t)} \mathbf{r}\wedge\mathbf{u}\mathrm{d}V + \int_{\partial\Omega_f(t)} \mathbf{r}\wedge(\mathbb{T} + (\mathbf{u}-\mathbf{u_f})\otimes\mathbf{u})\mathbf{n}\mathrm{d}S - \int_{\partial\Omega_s} \mathbf{r}\wedge((\mathbf{u}-\mathbf{u_b})\otimes\mathbf{u})\mathbf{n}\mathrm{d}S$$

where $\mathbf{r} = \mathbf{x} - \mathbf{x_{CoM}}$. Forces and torques applied onto the body are then computed from the momentum conservation equation through the control volume. Similarly, the resulting instant power can also be computed *via* the control volume approach. Basically, the power is computed from the fluid kinetic energy and the dissipation of viscous forces through the control volume. We refer the reader to [10] for more details. In this work, we will prefer to compute energetic quantities *via* surface integrations by using discrete Lagrangian markers as we meshed the zebrafish body surface with a particularly important amount of elements.

2.4.5 Swimming body dynamics: an Eulerian-Lagrangian framework

In this section, we present how body dynamics are computed at each time step, when the state of the fluid flow is known by using Lagrangian markers in addition to the level-set function. Note the surface mesh formed by Lagrangian markers and previously described in section 2.2.3, is required for computing accurate body-related quantities: surface forces and torques, body volume, body center-of-mass, body inertia matrix.

Transportation

In the case of a Lagrangian framework for describing the contour of the body, the level-set function can directly be computed from Lagrangian markers, at each time step. Therefore, there is no need to deform the level-set function *via* a deformation velocity field. As for rigid transportation, Lagrangian markers are directly transported by rigid velocity. However, two pieces of information still need to be computed: the boundary velocity required for fluid penalization and the intermediate snapshots corresponding to computational time. On the one hand, intermediate shapes between each consecutive acquisition time are straightforward to obtain by linear interpolation of Lagrangian markers for example. On the other hand, boundary velocity is simply computed at each time step, by the consecutive positions of Lagrangian markers. Therefore, Lagrangian markers have to be accurately tracked across time. The generation of proper Lagrangian markers on the body surface is detailed in the next chapter.

Discretizations

As the surface of the body is well-described with discrete Lagrangian markers, the hydrodynamic forces and torques can be directly computed by integrating the hydrodynamic tensor $\mathbb{T}$ over the surface elements $\mathrm{ds_e}$:

$$\mathcal{F}^{\mathbf{hydro}} \approx -\sum_{e\in\Gamma_s} \mathbb{T}(\mathbf{u},p)\cdot\mathbf{n}\mathrm{ds_e} = -\sum_{l=1}^{N_s}\sum_{\theta_i=1}^{N_\theta} \mathbb{T}(\mathbf{u},p)\cdot\mathbf{n}\mathrm{ds_e} \tag{2.40}$$

$$\mathcal{M}^{\mathbf{hydro}} \approx -\sum_{e\in\Gamma_s} \mathbb{T}(\mathbf{u},p)\wedge\mathbf{r_e}\mathrm{ds_e} = -\sum_{l=1}^{N_s}\sum_{\theta_i=1}^{N_\theta} \mathbb{T}(\mathbf{u},p)\wedge\mathbf{r_e}\mathrm{ds_e} \tag{2.41}$$

where $\mathbf{r_e} = \mathbf{x_e} - \mathbf{x_{CoM}}$, $\mathbf{u}$ and p are calculated from Lagrange interpolations.As previously, notice that forces and torques could have also been computed by using the control volume approach briefly described previously. Nevertheless, we preferred the surface integration method given the density of the surface mesh we constructed.

2.4.6 Free swimming

In definitive, we obtain a system of six equations: three equations for the $\bar{\mathbf{u}}$-components plus three equations for the ω-components. Considering such a system, the swimming of the fish body is entirely determined by six equations. However, if we want to impose a specific motion we can reduce the system by locking the desired degree of freedom. For instance, two rotations and one translation have to be locked to obtain a (xy)-planar

motion which is written as:

$$\begin{cases} \rho_s V_s \frac{\mathrm{d}\bar{u}_x}{\mathrm{d}t} = \mathcal{F}_x^{hydro} \\ \rho_s V_s \frac{\mathrm{d}\bar{u}_y}{\mathrm{d}t} = \mathcal{F}_y^{hydro} \\ \bar{u}_z = 0 \end{cases} \tag{2.42}$$

$$\begin{cases} \omega_x = 0 \\ \frac{\mathrm{d}J\omega}{\mathrm{d}t} = \mathcal{M}_y^{hydro} \\ \omega_z = 0 \end{cases} \tag{2.43}$$

Most of our 3D simulations have been performed with an imposed planar motion as 3D motions are very sensible to any slight angle variation which can lead to unstable 3D swimming. Furthermore, large 3D rotations are produced in case of an asymmetric body geometry as with fish geometries. In fact, these spurious rotations and instabilities are mainly due to the geometrical shape of the posterior and the tail of the fish which produces the propulsion. Indeed, the tail is generally very thin relatively to its width which creates natural unstable 3D motions. As for the actual swimming motion, fishes can follow a planar motion but they have biological resources to be able to stabilize themselves which we did not model. For example, pectoral fins can be important for stabilization.

2.5 Implementation

The whole *NaSCar* code where the previous numerical elements have been implemented is an $in-house$ solver developed and validated by Michel Bergmann within the MEMPHIS team. *NaSCar* constitutes a Navier-Stokes solver to compute fluid flows around immersed bodies, on top of a Cartesian grid. Basically, the body is represented by a level-set but a surface mesh with Lagrangian markers can be provided to enable surface computations. Similarly, the CFD code can be run either in 2D or in 3D. In this work, we started with 2D simulations and a full Eulerian approach before going into more accurate and realistic 3D simulations with Lagrangian markers.

Experimental data consists of fish deformations extracted from escape swimming videos. In 2D, the data are composed of 2D level-set describing the immersed body shape. In addition, the deformation velocity field is also required as input to the CFD code. Instead, in 3D, input data are a series of 3D deformed snapshots corresponding to the experimental video frames, described by a surface mesh and Lagrangian markers. At each time step, two pieces of information are needed from the experimental imaging data: the instant body shape and the deformation velocity generated by body kinematics to penalize the computed fluid velocities and transport the body geometry (Eulerian). The explicit deformed geometry of the body which is either a contour in 2D or a surface in 3D, is required to directly calculate the deformation velocity and to circumvent

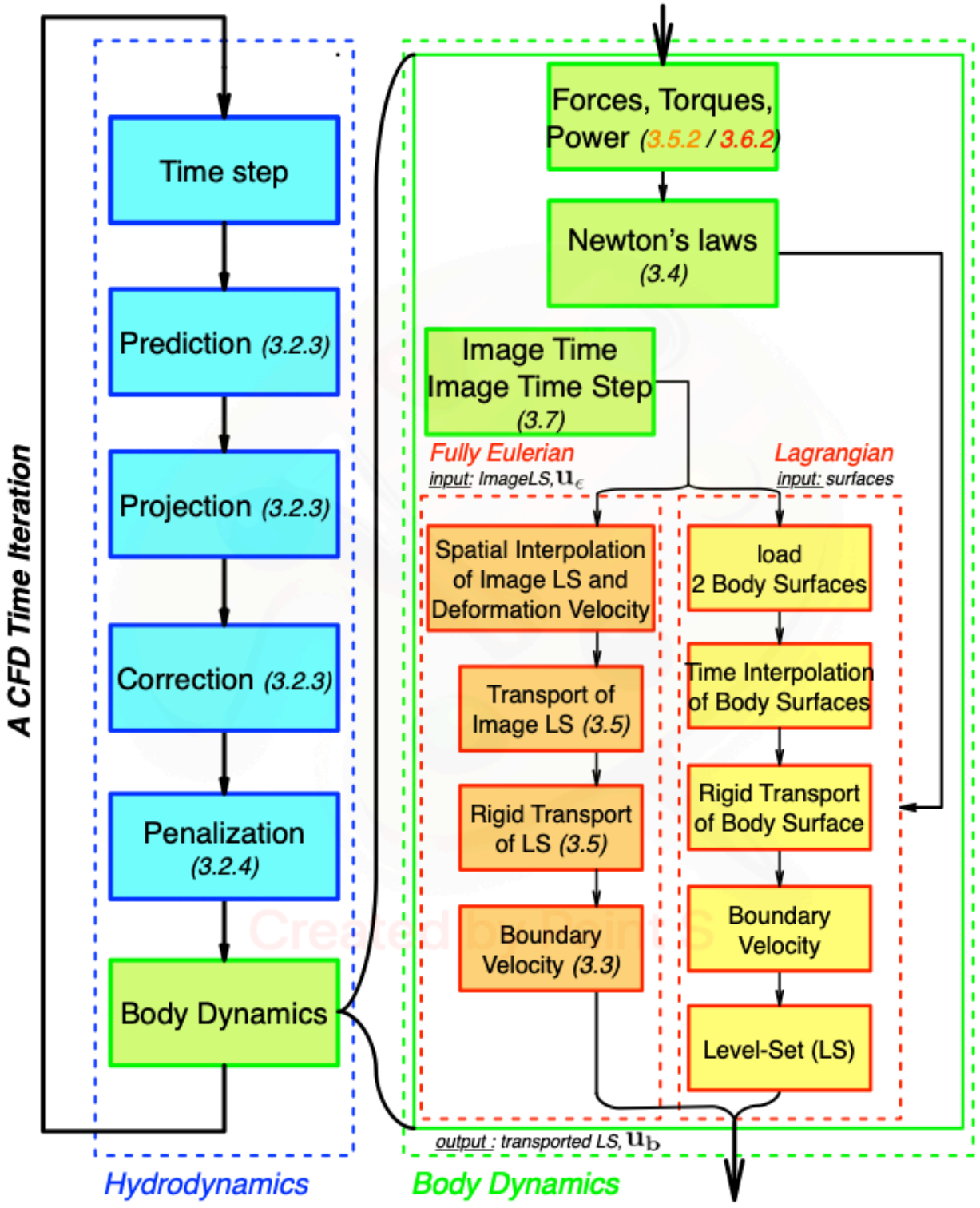


FIGURE 2.10: **Algorithm illustrating the interaction of the different numerical components and the link to the experimental data.**

the transportation to the next time step with the deformation velocity (Lagrangian). Indeed, the body deformation is also tracked while transporting a deformed body shape by rigid velocities which enables to implicitly transport the level-set by updating it. The interpolation at a computational time of the input data is performed either using the deformation velocity in 2D (Eulerian) or with simple linear interpolation of Lagrangian markers (Lagrangian). Therefore, the next chapter focuses on the extraction and reconstruction of body shapes and body deformation velocities from experimental video imaging of zebrafish eleuthero-embryo escape locomotion.

Chapter 3

Generation of input data from experimental videos

This chapter presents all the different contributions which shape the input data of the CFD code, from the experimental videos of actual escape locomotion to the computation of deformation velocity in 2D or the complete reconstruction in 3D of the swimming video. This elaboration process begins with the experimental data. A complete view of the experimental procedure is provided. The image processing algorithm is then briefly detailed. The essence of this step is to realign the video frames onto the center of mass to remove the rigid displacements using a *Procrustes Analysis* step. Afterward, two classes of methods were developed. On the one hand, we can derive the deformation velocity by using the theory of optimal transportation. On the other hand, we can generate body surface meshes with Lagrangian markers from the extraction of the midline. In 3D, a second step of *Procrustes Analysis* needs to be performed, especially to take into account the 3D center of mass into the computation of rotation. A review of a few validations required at each step is provided but most of the definitive numerical process validations will be provided in Chapter 4.

3.1 Experimental set-up

The experiments have been run with wild-type zebrafish eleuthero-embryos aged of 5 days post-fertilization (dpf) and reared within the zebrafish housing facility of the MRGM lab. Basically, the experiments consisted of recording the zebrafish escape locomotion after being stimulated by an EFP. In this section, we briefly present the animal housing facility before detailing the experimental setup for video recording.

(A) Insight of zebrafish breeding facilities.

(B) Zebrafish eleuthero-embryo production unit.

FIGURE 3.1: **Zebrafish are reared inside and by the MRGM lab.** Adult zebrafish are reared in large tanks while zebrafish eleuthero-embryos and larvae are bred inside suitable aquariums, after hatching.

3.1.1 Zebrafish husbandry

The 5 dpf eleuthero-embryos were obtained by natural mating and raised in embryo water containing 86 μg/ml Instant Ocean (Aquarium Systems, Sarrebourg, France), 0.55 mM $CaSO_4$, $2H_2O$ (Alfa Aesar, CAS n°10101-41-4), dissolved in reverse-osmosis purified water, at 28 ± 1 °C, with an 11L:13D photoperiod. At 5 dpf, the zebrafish eleuthero-embryos were transferred in highly-filtered fish water, before performing an electric field pulse motor response (EFPMR) test and recording the resulting escape response on the top of the experimental cell. Beforehand, fluid solutions or chemical compound exposure were eventually prepared to experiment specific conditions. Further details are provided in Chapter 5 regarding the experimental conditions used.

3.1.2 Ultra high-speed camera setup

During each recording session, the temperature of the experimental chamber was controlled and fixed at 28 ± 2 °C. The recording set-up was formed by an EFP module synchronized with an ultra-rapid camera, installed on the top of an experimental cell (Fig. 3.2). All experimental imaging was then processed by the Photron FASTCAM Viewer software (PFV4). The experimental cell consisted of circular-like in-house electrodes mounted inside a 5 cm-diameter Petri dish (Fig. 4.7 A).

Two ultra-high-speed cameras have been used. First, we used a Photon FASTCAM SA3-120K camera (Photron USA Inc., San Diego, CA, USA) for which we essentially tested our numerical approach: from image processing to optimal transportation results. This

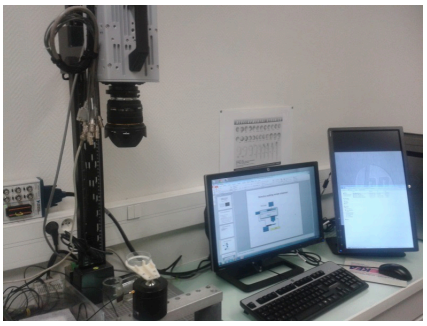


FIGURE 3.2: **The MRGM lab is equipped by two ultra high-speed FastCam cameras which can record up to 15,000 fps.**

camera was used with a 256×256 image resolution and 15,000 fps acquisition frequency which was considerable to obtain very close frames. Then, we used a second high-speed camera namely Photron FASTCAM Mini WX100, for quality enhancement, which could also reach up to 15,000 fps. However, a frame rate of 10,000 fps was found more suitable for our experimental procedure. All ultra-high-speed cameras were mounted with a Sigma 105 mm F2.8 EX DG macro lens and positioned on the top of the experimental cell (40 cm-height). With the latter camera, the image resolution (512×420) enabled to cover a window of approximately $13.1 \times 10.700\,\mathrm{mm}^2$. We calibrated the pixel dimensions by using a graduated ruler at a millimeter scale. Regarding the important resolution, we found the pixel size approximately corresponded to $\delta_{px} = 0.026\,\mathrm{mm}$ and $\delta_{px} = 0.030\,\mathrm{mm}$, for Mini WX100 and SA3-120K, respectively. Given the zebrafish eleuthero-embryos are sized of approximately 4 mm at 5 dpf, the camera window covered less than three times the eleuthero-embryo's length. As a consequence, we needed to track efficiently the zebrafish movements, especially after the stimulation. Regarding the high image resolution and the relative distance of the camera, the cell lighting needed to be adjusted accordingly. Indeed, we also equipped the experimental cell with a lighting system underneath, to increase the image contrast between the zebrafish silhouette and the fluid background. The plate was illuminated from below by an LLUB White LED Blacklight 50×50 (PHLOX, Aix-en-Provence, France) adjusted to 15% using a Gardasolft RT 220-20 led light controller(Gardasoft, Cambridge, UK). The light intensity in the experimental platform, measured using an ILT1400 radiometer (International Light technologies Inc., Peabody, MA, USA), was approximately 800 $\mu\mathrm{W/cm}^2$.

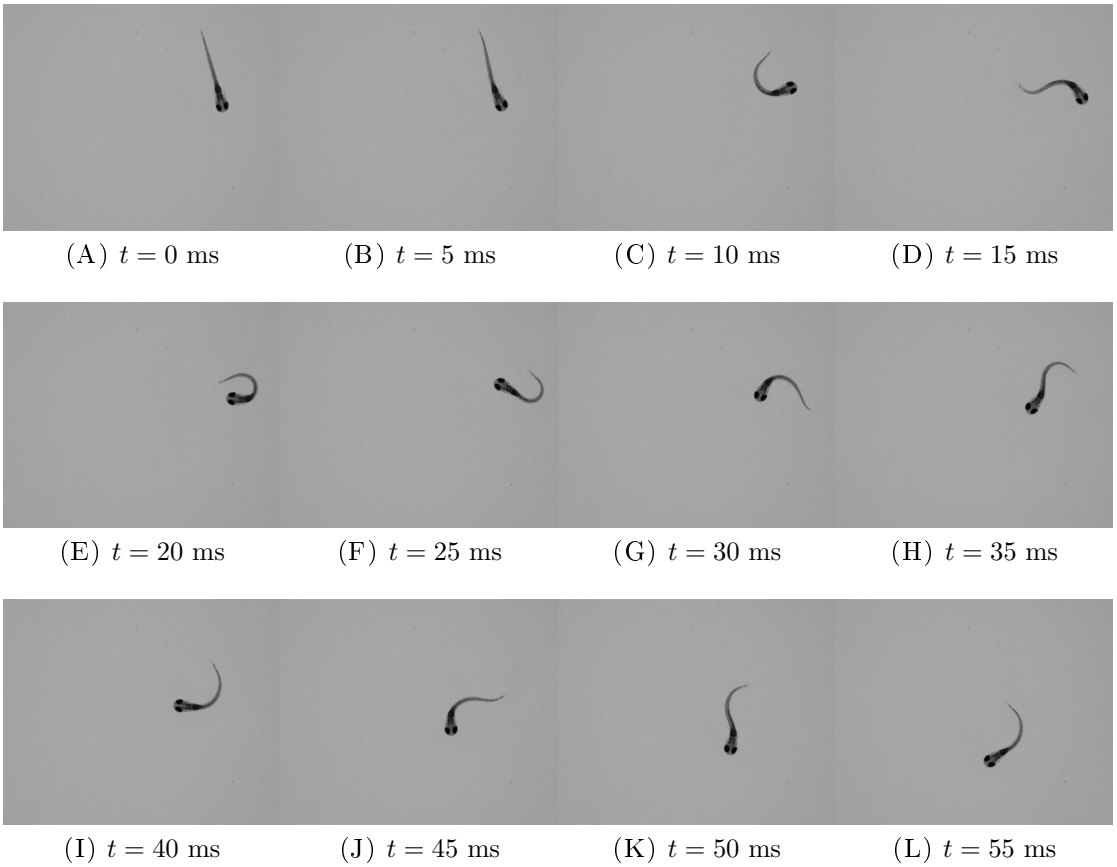
In this work, we used ultra-high-resolution images in time and space, mainly due to our numerical process to track the zebrafish motion and above all, the stimulation system we used. In that regard, zebrafish eleuthero-embryos do have a slow cruising swimming motion and can perform slight accelerations composed of turns, especially to reach food or when responding to visual stimulations. However, the purpose of the experimental set-up is to stimulate zebrafish eleuthero-embryos in order to obtain an EFPMR which produces large, ultra-fast, and stereotyped fast-start escape deformations. As a consequence, an important resolution was required to capture accurately the body deformations along with the zebrafish silhouette. Pools up to five eleuthero-embryos may be positioned in

the stimulation cell. However, we only considered individuals for biological experiments (Chapter 5) to avoid altered experimental conditions in terms of imaging or locomotor behavior. Thus, the eleuthero-embryo was individually positioned in the stimulation chamber and a 20 V EFP was applied for 10 ms, driving an ultra-rapid escape response. Individual videos were recorded for each experimental condition in order to calculate mean measurements and thus integrate the variability of escape responses and biological animals. All statistics were conducted by using GraphPad Prism v8.02fig, GraphPad Software, Inc.
As explained, the camera was synchronized with an EFP system to start the recording while initiating the electric pulse in the experimental cell. The actual shape of the custom electro-magnetic signal is known but not shown as the EFP cell developed by the MRGM team will be published. For each experiment, the Petri dish was filled with 4 ml of pure fluid which approximately corresponded to a fluid height slightly above $h = 2$ mm (without the presence of electrodes). In that way, zebrafish eleuthero-embryos can move freely but the motion tends to stay horizontal and planar within such small heights of fluid. Indeed, the major constraint of our model resided in the three degree-of-freedom (3-DoF) motion as we are not able yet to track full 3D motions. In addition, despite our efforts for filtering many times the fluid solution, micro particles may still be present and pollute the video and a relative thin layer of fluid could limit the amount of micro-contaminant. Eventually, the image focus was facilitated when only recording planar motions.

3.1.3 A fast-start escape response

Hereafter is presented a standard zebrafish escape locomotion in water (Fig. 3.3), recorded by the Mini WX100 ultra-rapid camera at $10,000$ fps. In this case, the eleuthero-embryo was recorded individually to avoid any perturbation from the surroundings. We can see the eleuthero-embryo is considered at rest when the body is straight undeformed. Note we easily measured the total eleuthero-embryo's length by using the first image frame. The tail tip is formed by a thin transparent membrane which can be visible with such high-resolution images. In this experimental case, the eleuthero-embryo was sized of approximately 4.2 mm. Note all measured lengths are given in total length *i.e.* from the snout tip to the tail tip, including the MFF.

Although only a few frames are represented, we can clearly identify the three different phases of a stereotyped escape response: the C-bend and counter-bend motions followed by the fast-swimming stage. Normally, the C-bend deformation is larger than the counter-bend one. However, these preparatory and propulsive stages are variable from one zebrafish eleuthero-embryo to another, especially considering each eleuthero-embryo can perceive the EFP differently. In what follows, we largely base the description of our numerical process on these particular video frames. After recording such swimming



(A) $t = 0$ ms (B) $t = 5$ ms (C) $t = 10$ ms (D) $t = 15$ ms

(E) $t = 20$ ms (F) $t = 25$ ms (G) $t = 30$ ms (H) $t = 35$ ms

(I) $t = 40$ ms (J) $t = 45$ ms (K) $t = 50$ ms (L) $t = 55$ ms

FIGURE 3.3: **Example of an escape locomotion (fast-start) of a 5 dpf zebrafish eleuthero-embryo.** We can see how fast and large the body kinematics is, in reaction of an electric stimulus. The zebrafish eleuthero-embryo is initially at rest (Fig. 3.3 A) and initiates the so-called C-bend motion (Figs. 3.3 A-3.3 C) where a C-shaped is formed (Fig. 3.3 C). Then, the eleuthero-embryo engages a counter-bend motion (Figs. 3.3 C-3.3 E). Note the counter-bend deformation (Fig. 3.3 E) is particularly significant in this escape swimming. Generally, the C-bend constitutes a larger body deformation. Finally, multiple fast-swimming cycles can be observed (Figs. 3.3 E-3.3 L). The timings of the three characteristic stages of escape swimming are given for illustrative purpose. Individual recording at 10,000 fps.

videos, each image needed to be processed according to the image quality and the swimming kinematics.

3.2 Image processing

After the recording of each individual escape swimming with ultra-high frame rates, from 10,000 to 15,000 fps, we conserved between approximately 500 and 1000 video frames per experimental recording as we only focused on the first stages of the fast-start: C-bend, counter-bend and only two cycles of fast swimming which corresponds to four tail-beats. As explained before, the image resolution was directly related to the chosen acquisition frequency: 512×420 at 10,000 fps and 256×256 at 15,000 fps. A custom Matlab code has been developed to process the experimental videos. Basically, each

frame is segmented and analyzed afterward, essentially to realign all zebrafish eleuthero-embryos and eliminate the swimming rigid motion produced between each consecutive image. The algorithm has been supported by multiple variations, implemented to handle a large spectrum of video quality and image formats.

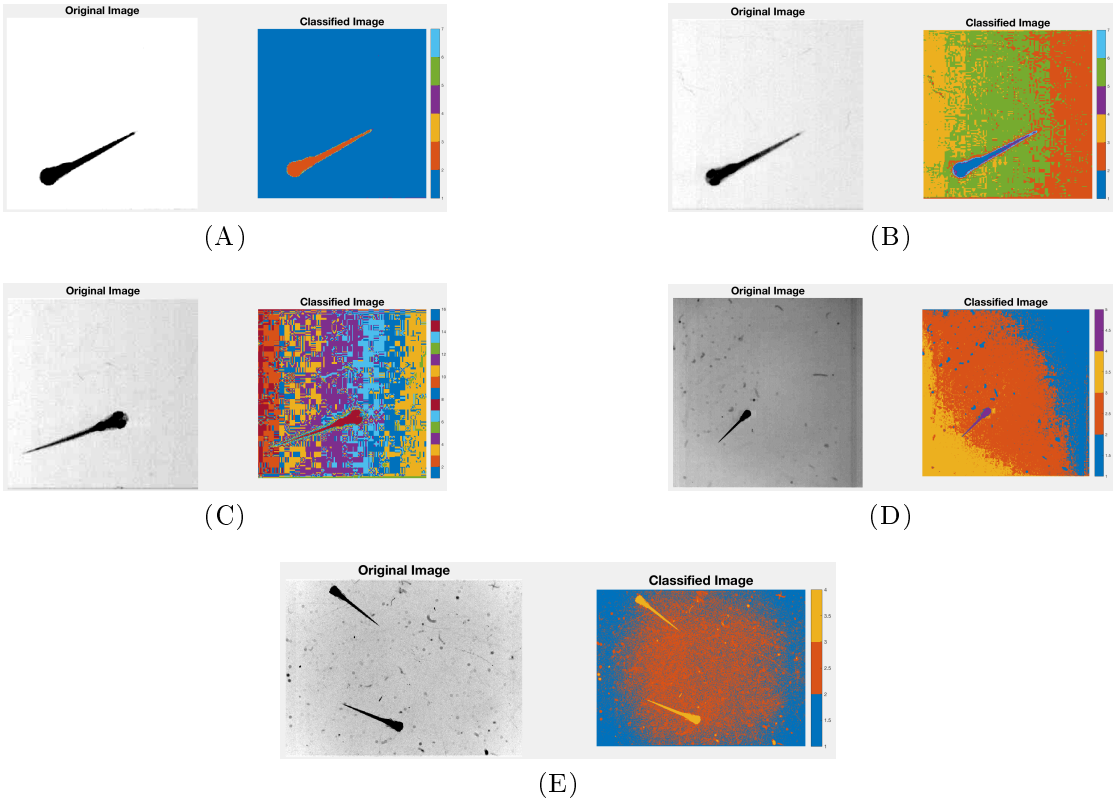
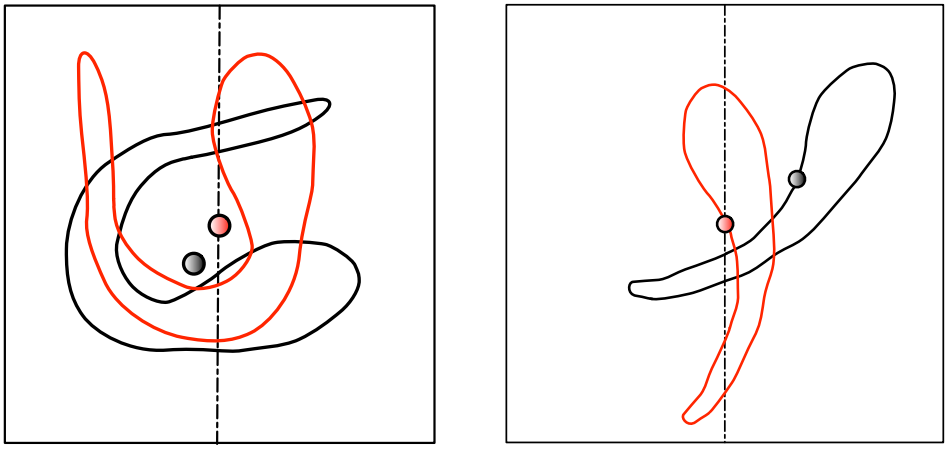


FIGURE 3.4: **Various illustrations of the impact of video quality on the computed segmentation.** All computed clusters are represented for four video examples. For the first case, a k-means algorithm has been used with multiple clusters, on the original video (Fig. 3.4 B) and the pre-processed video (Fig. 3.4 A). Second (Fig. 3.4 C) and third (Fig. 3.4 D) show the variation of background quality impact. Finally, a different camera has been used for recording the last classified image (Fig. 3.4 E).

The image segmentation process represents a key step to extract the proper zebrafish silhouette from the background pixels. To that end, we used clustering methods namely k-means although a simple pixel-wise threshold could also be considered. However, clustering methods have been found more robust to properly separate the zebrafish pixels from the background pixels. Indeed, threshold methods had two main drawbacks. First, the definition of the threshold value was not straightforward to be accurate enough near the body boundary. Indeed, when using such high-resolution images, the eleuthero-embryo's boundary was not easily-defined. Moreover, we needed to choose a threshold independently to each frame and experimental video. Finally in our case, special attention was required to not drag spurious contaminant particle masses into the silhouette of the zebrafish body. Similarly, the tail of the zebrafish needed to be accurately captured in each frame. When considering threshold methods, the tail was regularly

vanishing. That is why, we eventually over-classified the pre-processed video before automatically separating the set of clusters into two regions: the zebrafish eleuthero-embryo and the fluid background. In that way, we ensured to segment adequately, regardless of the video quality. Indeed, the quality of the segmentation process could highly differ from one experimental video to another (see Fig. 3.4). Therefore, we implemented a robust and reliable segmentation process to automatically extract the zebrafish mass across different videos. Our image processing is structured with three steps. First, we compute the rigid motion before smoothing the experimental data. Finally, we proceed to segment each frame while subtracting the rigid kinematics. The first image processing step is presented in detail in the first loop of Algorithm 1. As explained previously, the center-of-mass coordinates can only be defined pixel-wise, at first. Similarly, the tilt angle is very noisy considering how high the acquisition frequency is. That is why, we considered to smooth the swimming parameters before realigning each slice accordingly.



(A) A large deformation example.

(B) A small deformation example.

FIGURE 3.5: **Illustration of the *Procrustes Analysis* for two different body deformations.** For subtracting the rigid motion, center-of-mass positions of the considered body silhouette (black contours) are first computed (black points). Note red contours and red points denote the output contours and center of mass, after subtracting rigid kinematics. The translation part of the rigid transformation consists of re-centering the center of mass. The rotation part resides in the computation of the global body angle relatively to vertical, for example. The body angle is also called tilt angle. Notice the tilt angle is straightforward to estimate in case of small deformations (Fig. 3.5 B) and ill-defined for strong deformations (Fig. 3.5 A).

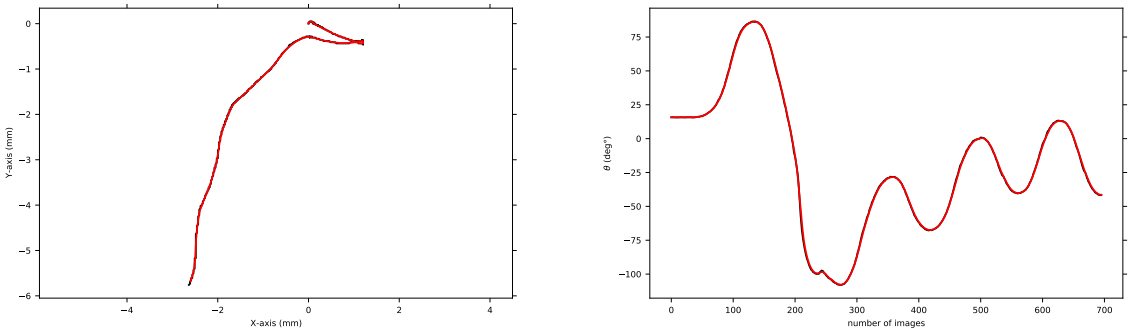
The calculation of rigid swimming kinematics are performed *via* the *Procrustes Analysis* step. Actually, the *Procrustes Analysis* is key to our experiment-based approach as this step enables to recover the body deformations *i.e.* body kinematics, from experimental imaging (Fig. 3.5). For instance, Fig. 3.6 represents the rigid kinematics *i.e.* center-of-mass translation and rotation, of the previously introduced experimental case (Fig. 3.3). The main goal of a *Procrustes Analysis* step is to estimate the rigid transformation between two consecutive body silhouettes in order to subtract all rigid kinematics from

body silhouettes. As a result, all center-of-mass positions are realigned to the frame center and all body tilt rotations are subtracted to fit the initial silhouette orientation. Note the straight silhouette of the initial frame has also been rotated to fit the vertical line. We emphasize the overall difficulty raised by the *Procrustes Analysis* process resides in the computation of the tilt angle (Fig. 3.5). In fact, the computation may be straightforward for a straight body (Fig. 3.5 B) but the problem can also be ill-posed in the case of large deformations (Fig. 3.5 A). In the latter case, the global body angle may not be well-defined. Therefore, spurious rotation motions might be introduced for large deformations (see Fig. 3.6 B, between images 200-300), when computing the rigid deformation between straight and deformed body silhouettes. That is why, a correction algorithm has been developed to accurately estimate the body rotation of strong deformations, by using the surrounding experimental body kinematics. This additional step will be detailed further at the end of the chapter (see section 3.5).

Algorithm 1: Image processing: segmentation and *Procrustes Analysis*

Input: Video images (top-views of zebrafish swimming), smoothed rigid swimming kinematics
Data: 2D grayscale images, JPEG or TIFF formats
Output: Segmented and realigned data: the body deformations
foreach *Image* **do**
  SEGMENTATION
    Adjust image histogram, *threshold* low and high intensity pixels
    ***Cluster* using k-means algorithm** (number of classes between 3 and 15)
    Threshold clusters in two groups, *filter*
  RIGID KINEMATICS
    ***Compute* the Center-of-Mass** coordinates
    Resize and *Center* on a 300×300 grid
    Reorient with the last computed tilt angle
    ***Compute* the new tilt angle**:
      interpolation from Barycentric coordinates and the pixel-wise tilt angle, relatively to the vertical
    Reorient by subtracting the tilt angle
SMOOTHING
  Smooth the discrete **tilt angle** series
  Smooth the pixel-wise **Center-of-mass** series
foreach *Image* **do**
  SEGMENTATION
    Adjust image histogram, *threshold* low and high intensity pixels
    ***Cluster* using k-means algorithm** (number of classes between 3 and 15)
    Threshold clusters in two groups, *filter*
  SUBTRACTING RIGID KINEMATICS
    Resize and ***Center*** on a 300×300 grid, using the smoothed Center-of-Mass
    Reorient by subtracting the smoothed tilted angle
    Resize and *Center* on a 200×200 grid (optional vertical shift), *filter*

Finally, the last step consisted of realigning each video image with the smoothed swimming kinematics after the segmentation step. This second loop (see Algorithm 1)



(A) smoothed center-of-mass Trajectory. (B) smoothed Angle.

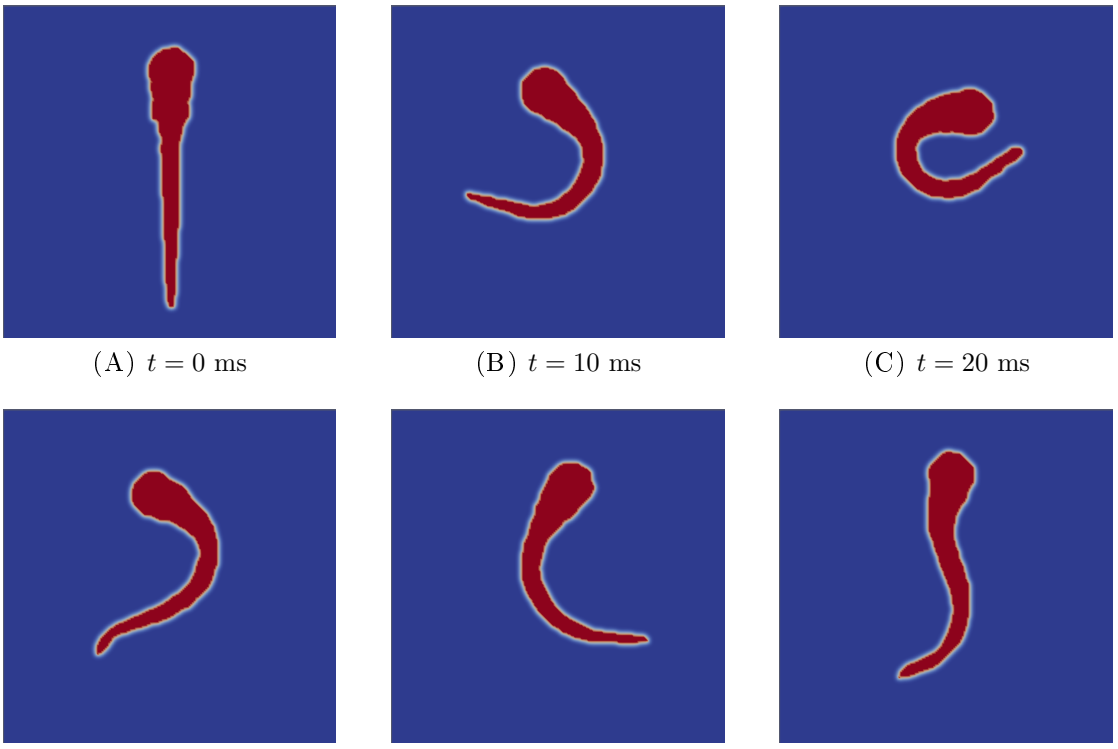
FIGURE 3.6: **Rigid kinematics computed *via* the Procrustes Analysis step.** The center-of-mass trajectory (Fig. 3.6 A) and the tilt angle (Fig. 3.6 B) have been smoothed (red) to eliminate the data noise (black).

is nearly identical to the first one, except for the computation of rigid kinematics. The center of mass was directly approximated using the average coordinates of the body silhouette. Similarly, the global rotation motion which we defined as a tilt angle *i.e.* the representation of the body angle at rest, was computed as a barycentric quantity by evaluating the tilt angle of each pixel, relatively to the center of mass. Indeed, we weighted each pixel with the distance to the center of mass, to define the barycentric coordinates. Notice we did not directly rotate the current image according to the initial frame. Instead, we subtracted each angle relatively to the previous image. As a result, the subtracted angle was well-defined. Finally, we centered each segmented image inside a 200×200 grid. Eventually, a slight shift in length is performed if required.

For illustration purpose, we represented in Fig. 3.7 the segmentation and realignments results based on the previously introduced experimental video (Fig. 3.3).

3.3 Optimal transportation

In this section, we briefly describe the optimal transportation theory and the main interest for our applications. First, the optimal transportation theory was considered to extract the deformation velocity field from the segmented and aligned 2D video frames. Secondly, recent variations of optimal transportation problem provided an efficient solution for shape interpolation issues. After the brief introduction to optimal transportation, we present how we actually used those algorithms. For more details, we refer the reader to the papers our work is based upon [6, 18, 109]. In particular, the regularized problem we used for 3D reconstruction of zebrafish shape [109] is detailed in Appendix A, alongside various illustrations of theoretical and numerical advances. The optimal transportation algorithms developed by the MEMPHIS team [18] have been



(A) $t = 0$ ms (B) $t = 10$ ms (C) $t = 20$ ms

(D) $t = 30$ ms (E) $t = 40$ ms (F) $t = 50$ ms

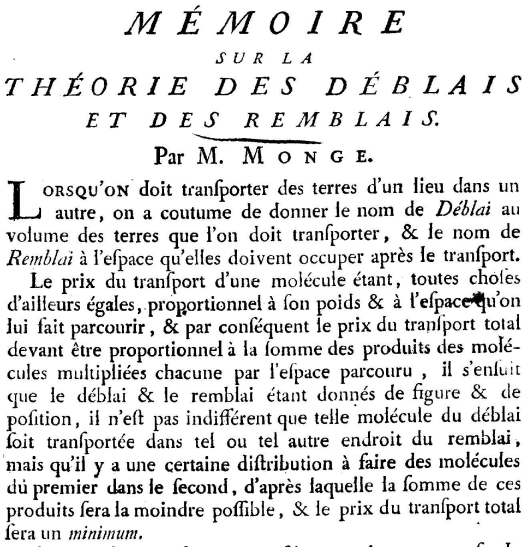
FIGURE 3.7: **Illustration of the image processing output: segmented and realigned zebrafish silhouettes.** Various morphological operations and boundary filters have been performed to regularize the body boundary shape. We represented the set of video frames corresponding to Figs. 3.3 A (Fig. 3.7 A), 3.3 C (Fig. 3.7 B), 3.3 E (Fig. 3.7 C), 3.3 G (Fig. 3.7 D), 3.3 I (Fig. 3.7 E) and 3.3 K (Fig. 3.7 F). The color scale is defined by zeros (blue) in the background pixels and ones (red) inside the zebrafish body. Segmented output images are sized of 200×200 pixels.

used to extend the work initiated by Bergmann *et al.* [11] which originated the idea of computing the deformation velocity field from images.

3.3.1 Preamble

The question of optimal transportation is an old problem originated by the French mathematician and engineer Gaspard Monge for real-world mass transfer issues (1781, [88]). Then, Leonid Kantorovich (1942, [57]) transferred the problem of optimal transportation to economics as a resource allocation problem which ignited new interests with the raise of linear programming. Afterward, Yann Brenier rewrote the optimal transportation issue with a dynamic formulation issued from a fluid mechanics point of view, which dragged new fields of interest. More recently, Cedric Villani contributed to provide new insights about optimal transportation theory [127, 128] and the multiple links to fundamental mathematics and physics with entropy questions. In the last decade, several disciplines gained interest in optimal transportation theory and its applications, such as image processing, machine learning and computer graphics, computational

geometry. For computational optimal transportation, we refer the reader to reviews [96, 100, 102].



MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.
Par M. MONGE.

LORSQU'ON doit tranſporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit tranſporter, & le nom de *Remblai* à l'eſpace qu'elles doivent occuper après le tranſport.

Le prix du tranſport d'une molécule étant, toutes choſes d'ailleurs égales, proportionnel à ſon poids & à l'eſpace qu'on lui fait parcourir, & par conſéquent le prix du tranſport total devant être proportionnel à la ſomme des produits des molécules multipliées chacune par l'eſpace parcouru, il s'enſuit que le déblai & le remblai étant donnés de figure & de poſition, il n'eſt pas indifférent que telle molécule du déblai ſoit tranſportée dans tel ou tel autre endroit du remblai, mais qu'il y a une certaine diſtribution à faire des molécules du premier dans le ſecond, d'après laquelle la ſomme de ces produits ſera la moindre poſſible, & le prix du tranſport total ſera un *minimum*.

FIGURE 3.8: **Title page of the Monge's *Memoire* introducing the optimal transportation problem.** [88]

Monge introduced the optimal transportation problem with the *Mémoires sur les déblais et les remblais*, in 1781 [88] (Fig. 3.8). The initial question was to transport a certain amount of soil to another location by considering the cost to transport each elementary molecule of mass. Monge defined the cost of transport $c(\mathbf{x}, \mathbf{y})$ of each elementary molecule with the mass m, to move from site $\mathbf{x}$ to emplacement $\mathbf{y} = T(\mathbf{x})$ as proportional to the distance d between initial and final locations: $c(\mathbf{x}, \mathbf{y}) = md(\mathbf{x}, \mathbf{y}) = m\|T(\mathbf{x}) - \mathbf{x}\|$. As a result, the total cost to transport a pile of sand from one place to another was defined as: $\mathcal{C} = \sum_{\mathbf{x}} d(\mathbf{x}, T(\mathbf{x}))m$. The Monge's goal was to find the optimal transportation map $T : \mathbf{x} \mapsto \mathbf{y} = T(\mathbf{x})$ for which the total cost of transport $\mathcal{C}$ is minimum.

Let introduce the notation of [109] with $X \subset \mathbb{R}^n$ be a metric space which represents a bounded domain such as a surface or image plane. Let define $d : (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \mapsto \|y - x\|_{\mathcal{L}^p} \in \mathbb{R}^+$ a $\mathcal{L}^p$-distance function, μ, ν be two probability measures on X and ρ_0, ρ_1 be two mass density distributions on X. The Monge problem is then defined as finding the optimal transportation map T such that T transports μ to ν: $T_{\#}\mu = \nu$ and T minimizes the cost function:

$$\mathcal{C} = \int_X c(x, T(x))d\mu(x) = \int_X d(x, T(x))^p d\mu(x) \tag{3.1}$$

When considering the transportation of mass densities ρ_0 to ρ_1, the cost functional (3.1) becomes:

$$\mathcal{C} = \int_X c(x, T(x))\rho_0(x)dx = \int_X d(x, T(x))^p \rho_0(x)dx \tag{3.2}$$

Note Monge originally considered the problem with $p = 1$. In what follows, we will only consider the Euclidean distance ($p = 2$): $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\| = \sqrt{\sum_i (y_i - x_i)^2}$. Obviously,

the additional implicit constraint was the mass conservation during the transportation:

$$\int_X \rho_0(x)dx = \int_{T(X)} \rho_1(x)dx \tag{3.3}$$

During the mid-XX^{th} century, major contributions have been developed by the Russian mathematician and economist Leonid Kantorovich which led to a Nobel prize in economics in 1975. Basically, Kantorovich considered the optimal transportation problem as a resource allocation question. In that case, the original distribution of resources needs to be transported to a new configuration, optimally. Unlike Monge which searched for the optimal displacement of mass particles, Kantorovich desired to transport an amount of resource from production sites $\mathbf{x}$ to final sites $\mathbf{y}$ by minimizing the total cost of the transport $c(\mathbf{x}, \mathbf{y})$. In consequences, the total cost to allocate all resources from the different production sites to final sites is then defined as: $\mathcal{C} = \sum_{\mathbf{x}} \sum_{\mathbf{y}} c(\mathbf{x}, \mathbf{y}) p_{xy}$ where p_{xy} denotes the amount of resources allocated from site $\mathbf{x}$ to site $\mathbf{y}$. Obviously, the underlying constraint resides in the conservation of resources during the allocation operation: $\sum_x (\sum_y p_{xy}) = \sum_y (\sum_x p_{xy})$. As a result, Kantorovich aimed to find the allocation map p while minimizing $\mathcal{C}$. In the discrete case, we have to minimize the quantity $\mathcal{C} = \sum_{\mathbf{i},\mathbf{j}} c_{ij} p_{ij}$ considering the transportation and cost matrices under the conservation constraints $\sum_i p_{ij} = p_j$, $\sum_j p_{ij} = p_i$. The solution of the discrete problem has been well-studied and can be computed by using linear programming such as the so-called Simplex algorithm. Let consider the initial and final density distributions μ and ν also called marginals, and π a transportation plan *i.e.* a probability measure on $X \times X$. The quantity $\pi(\mathbf{x}, \mathbf{y})$ denotes the amount of resources or mass transported from position $\mathbf{x}$ within the initial configuration μ to position $\mathbf{y}$ inside the arriving distribution ν. The so-called Monge-Kantorovich problem (MKP) formulation consists of finding the optimal transportation plan π such that $\int_y \pi(\mathbf{x}, \mathbf{y}) = \mu(\mathbf{x})$ and $\int_x \pi(\mathbf{x}, \mathbf{y}) = \nu(\mathbf{y})$ which minimizes the total cost:

$$\mathcal{C} = \int_{X \times X} c(\mathbf{x}, \mathbf{y}) d\pi(\mathbf{x}, \mathbf{y}) \tag{3.4}$$

When considering the cost functional as the Euclidean distance, we can define the so-called Wasserstein metric [109]:

$$\mathcal{W}_2(\mu, \nu) = (\inf_{\pi} \int_{X \times X} \|\mathbf{y} - \mathbf{x}\|^2 d\pi(\mathbf{x}, \mathbf{y}))^{1/2} \tag{3.5}$$

Obviously, the $\mathcal{L}^2$-Wasserstein distance can also be defined for the initial Monge formulation:

$$\mathcal{W}_2(\rho_0, \rho_1) = (\inf_{T} \int_X \|T(\mathbf{x}) - \mathbf{x}\|^2 \rho_0(\mathbf{x}) d\mathbf{x})^{1/2} \tag{3.6}$$

The optimal transportation problem has then been revisited by Benamou and Brenier [6] with a fluid mechanics point of view. The major contribution of Benamou and Brenier was to introduce a time variable within the optimal transportation problem of Monge

and provide a dynamic optimal transportation formulation. Let consider a fluid bounded domain $\Omega \subset \mathbb{R}^n$ and the fluid density $\rho(\mathbf{x}, t)$ and velocity $\mathbf{v}(\mathbf{x}, t)$ fields. In fluid mechanics, the continuity equation is written as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{3.7}$$

Let ρ_0 and ρ_1 be the two initial and final mass density distributions corresponding to time $t = 0$ and $t = 1$:

$$\rho(\mathbf{x}, t = 0) = \rho_0(\mathbf{x}) \tag{3.8}$$

$$\rho(\mathbf{x}, t = 1) = \rho_1(\mathbf{x}) \tag{3.9}$$

Under the previous framework, the Wasserstein distance can then be written as:

$$\mathcal{W}_2^2(\rho_0, \rho_1) = \inf_{(\rho, \mathbf{v})} \int_{t=0}^{1} \int_{\Omega} \|\mathbf{v}(\mathbf{x}, t)\|^2 \rho(\mathbf{x}, t) d\mathbf{x} dt \tag{3.10}$$

Brenier proved the existence and uniqueness of the solution T of problem (3.6) when T is derived from a convex function Ψ: $T = \nabla \Psi$ [20, 127, 128]. Notice that when combining this result with the constraint (3.3) and its corresponding Jacobian ∇T, we obtain the so-called Monge Ampère Equation (MAE), a non-linear and elliptic equation written as:

$$\rho_0(\mathbf{x}) = det(\nabla T(\mathbf{x})) \rho_1(T(\mathbf{x})) = det(\nabla^2 \Psi(\mathbf{x})) \rho_1(\nabla \Psi(\mathbf{x})) \tag{3.11}$$

Note Benamou and Brenier also proved in [6] the Lagrangian coordinates $X(x, t)$ describing the optimal path are given by:

$$X(x, t) = x + t(\nabla \Psi(x) - x) = (1 - t)x + tT(x) \tag{3.12}$$

This well-known property also called McCann's interpolation illustrates how the optimal path is provided by straight-line trajectories. Thus, rotating movements are not well-captured with the optimal transportation theory (see Appendix A). Besides, it shows how optimal transportation solution enables to simply interpolate and recover intermediate densities between ρ_0 and ρ_1. Thus, optimal transportation is basically a displacement interpolation method for mass densities. Indeed, it is important to note classical linear interpolation does not compute a displacement interpolation when applied to mass densities (Fig. 3.9).

In fact, the displacement of mass density cannot be directly interpolated as we do not have a proper Lagrangian representation of mass particles. That is why, optimal transportation is important to find the optimal transportation map and thus, the unique associated velocity $\mathbf{v}$. As previously, Brenier demonstrated the existence and uniqueness

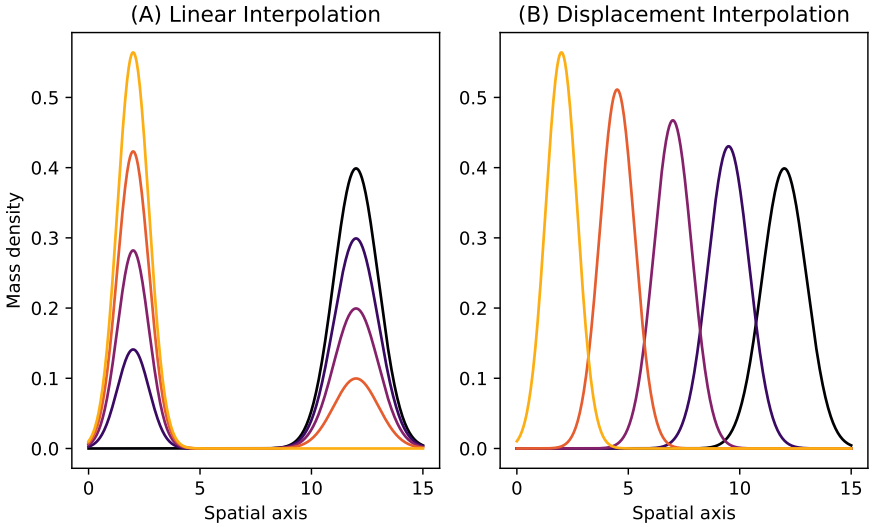


FIGURE 3.9: **Comparison between linear and Wasserstein interpolations for computing intermediate Gaussian densities.** Initial ($t = 0$) and final ($t = 1$) densities ρ_0 and ρ_1 are represented in yellow and black, respectively. Both initial and final density are normalized. We represented three intermediate densities, corresponding to $t = 0.25$ (red), $t = 0.5$ (violet), and $t = 0.75$ (purple).

of the solution $\mathbf{v}$ of the problem (3.10) and $\mathbf{v}$ is derived from a potential Φ:

$$\mathbf{v} = \nabla\Phi \tag{3.13}$$

More precisely, the potential Φ denotes a Lagrange multiplier which verify the Hamilton-Jacobi equation:

$$\partial\Phi + \frac{1}{2}\|\nabla\Phi\|^2 = 0 \tag{3.14}$$

Equation (3.14) shows the optimal solution $\mathbf{v}$ can be computed from pressureless Euler equations.

Besides, the fluid mechanics approach enlightens how the Monge-Kantorovich problem is related to a minimization of energy. Indeed, the term $\frac{1}{2}\rho\|\mathbf{v}\|^2$ simply represents the kinetic energy. Actually, optimal transportation problems may also be viewed through the Least Action Principle which can be stated as: "Nature, in making its effects, always acts by the simplest means possible." or in French:
“ Maintenant, voici ce principe, si sage, si digne de l’être suprême: lorsqu’il arrive quelque changement dans la Nature, la quantité d’Action employée pour ce changement est toujours la plus petite qu’il soit possible. ” - *Maupertuis, 1744 "Accord between different laws of Nature that seemed incompatible" (Mémoires de l’Académie Royale des Sciences Année 1744 pp. 417-426, 1748)*
Although theoretical results might effectively link optimal transportation to the Least Action Principle developed by Lagrange, optimal transportation can essentially be interpreted as a minimization of energy between two states, which in return might be viewed as a consequence of the Least Action Principle applied toward natural phenomena

such as fish displacements and associated biomechanics. Therefore, the optimal path to transport a density distribution ρ_0 to ρ_1 is given by minimizing the kinetic energy across time. Initially, we focused our interest on the optimal transportation theory to compute the deformation velocity field which transports the zebrafish segmented mass ρ_0 to the consecutive body density ρ_1, based on the assumption zebrafish eleuthero-embryos naturally follow an optimal path to deform themselves.

3.3.2 Numerical approaches to optimal transportation problems

In the last decade, optimal transportation methods have emerged and become well-popular in applied mathematics across multiple and various disciplinary fields from economics, image processing, and fluid mechanics to population dynamics and machine learning. Optimal transportation problems span over full discrete, semi-discrete, and continuous problems. Therefore, more and more numerical methods have been developed specifically for each problem and to enhance the overall precision and computational time and efficiency, especially for "real-world" problems with a massive amount of data. In what follows, I briefly introduce recent approaches to compute optimal transportation solutions. Nevertheless, I strongly refer the reader to complete and exhaustive reviews such as [96, 100, 102] for further details.

As stated before, since the mid-XX^{th} century, the allocation problem raised by Kantorovich was tackled by linear algebra and combinatory theory, and linear optimization. For instance, discrete resource allocation problems can be solved by simplex, Hungarian or auction algorithms. Note linear programming represents costly methods with a computational complexity of $O(N^3 \log N)$. Semi-discrete optimal transportation problems which consider Dirac masses either for the source or the target distribution have been studied *via* Laguerre or Voronoi diagrams.

As for full continuous optimal transportation problems, the choice of the numerical approach depends essentially on the nature of the problem. First, a robust computational algorithm was designed and developed by Benamou and Brenier (BB) to compute mass density and velocity along the optimal path between two configurations, by solving an augmented Lagrangian within the fluid mechanics framework [6]. Alternative methods derived from the Douglas-Rachford (DR) algorithm, have then been considered when solving the general minimization problem $\min_x F(Kx) + G(x)$. In particular, proximal operators and primal-dual proximal splitting methods have also been used for solving an optimal transportation problem [96]. For more details, we refer to review of [96]. More recently, a monotone and consistent numerical method called MA-LBR (for Monge-Ampère with Lattice Basis Reduction) has been developed to discretize the Monge-Ampère operator (3.11) [7], which takes advantage of the lattice geometry *via* the calculation of M-obtuse superbases. The recent advances about relaxed optimal transportation algorithms we used are detailed in Appendix A. Gradient-based

minimization methods were also implemented to optimize a mass-preserving mapping by multiple iterations [2]. In what follows, we mainly describe one specific approach based on pressureless Eulerian equations and linearization of the Monge-Ampère equation. Both Eulerian and Lagrangian implemented approaches [18] are based on that idea which consists of converging toward the optimal solution from an initial guess.

A pressureless Eulerian approach

As introduced previously, the optimal solution $T = \nabla\Psi$ of the initial Monge's problem verifies the MAE equation (3.11) and the solution $(\rho, \mathbf{u} = \nabla\Phi)$ of the dynamic formulation (3.10) for transporting the mass density distribution ρ_0 onto ρ_1, verifies the pressureless Euler equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{3.15}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = 0 \tag{3.16}$$

$$\rho(\cdot, t = 0) = \rho_0(\cdot), \rho(\cdot, t = 1) = \rho_1(\cdot) \tag{3.17}$$

Bouharguane et al fully developed in [18] both Lagrangian and Eulerian approaches by providing two efficient numerical methods for solving "real-world" optimal transportation problems. On the one hand, the Lagrangian approach consists of Newton iterations after linearizing the MAE operator, embedded into a continuation algorithm to enhance the robustness and convergence of the algorithm. On the other hand, the Eulerian point of view for computing optimal transportation solutions is designed to determine a fluid velocity field. Given the nature of the experimental data on which we desire to apply in parallel multiple optimal transportation calculations, we preferred the Eulerian algorithm to circumvent supplementary issues regarding the continuation process. Thus, we focused our attention onto the pressureless Eulerian approach implemented (Fig. 3.10) and detailed in [18]. Basically, the iterative algorithm consists of finding $(\rho^n, \mathbf{u}^n)$ such that $\rho^n \to \rho_0$ in the limit $n \to \infty$, while the transportation operation T gives $T_{\mathbf{u}^n}(\rho^n) \to \rho_1$. The initial guess is (ρ_0, u_0) which transports ρ^n to $\tilde{\rho}^n$ *via* the momentum conservation (3.16) is corrected with the term $\tilde{\mathbf{u}} = \nabla\tilde{\Phi}$ by solving the following Poisson problem derived from equation (3.15):

$$\nabla\left(\frac{\rho_1 + \tilde{\rho}_1^n}{2}\nabla\tilde{\Phi}\right) = \tilde{\rho}_1^n - \rho_1 \tag{3.18}$$

where $\tilde{\rho}^n$ denotes the density after the transportation of ρ^n. Note $\tilde{\rho}^n$ is not equivalent to ρ_1 while $\mathbf{u^n}$ is not optimal. Afterward, we compute backward equation (3.16) to transport $\tilde{\rho}^n$ with $\mathbf{u^n}$ onto the initial plan, which gives ρ^{n+1} and the corrected velocity $\mathbf{u^{n+1}} = \mathbf{u^n} + \alpha\tilde{\mathbf{u}}^{\mathbf{n}}$. The complete Eulerian algorithm described in [18] can be found in Fig. 3.10.

Algorithm 3. Eulerian iterative algorithm.

1. $n = 0$;
2. initialize u_0;
3. for initial data ρ_0 and $u_0 = X^n$, compute ρ_1^n and u_1^n by numerically solving system (3.3)–(3.4);
4. compute $\tilde{\Psi}$ by solving equation:
$$\nabla \cdot \left(\frac{\rho_1 + \tilde{\rho}_1}{2} \nabla \tilde{\Psi} \right) = \tilde{\rho}_1 - \rho_1$$
5. integrate backward in time system (3.3)–(3.4) with final conditions $\tilde{\rho}_1$ and $\tilde{u}_1$ to advect backward the velocity $\tilde{u}$ and get at initial time the velocity correction $\tilde{u}_0^n$ matching with ρ_0;
6. $X^{n+1} = X^n + \alpha\, \tilde{u}_0^n$;
7. $n = n + 1$;
8. go to 3 if convergence is not attained;

FIGURE 3.10: **Eulerian-based algorithm for approximating a pressureles Eulerian solution, from [18].** Note the equation system referred at steps 3 and 5 are Euler equations (3.15-3.16-3.17). The algorithm can be explained as an iterative approach to reduce the numerical error between the expected density after advection and the targeted mass distribution.

As for the numerical schemes used, we considered a second-order centered finite-difference scheme for solving equation (3.18) while the pressureless momentum equation (3.16) was computed *via* the remeshing particle method introduced in the previous chapter [132]. Hereafter is presented the interpolation Kernel $\mathcal{K}$ (3.20) we considered to remesh each quantity α onto the Cartesian mesh after a Lagrangian-like advection:

$$\tilde{\alpha}_i = \sum_q \alpha_q \mathcal{K}\left(\frac{x_i - x_q}{dx}\right) \tag{3.19}$$

where:

$$\mathcal{K}(x) = M_4'(x) = \begin{cases} 1 - 5x^2/2 + 3|x|^3/2 & \text{if } |x| \leq 1 \\ (2 - |x|)^2(1 - |x|)/2 & \text{if } |x|1 <\leq 2 \\ 0 & \text{if } |x| \geq 2 \end{cases} \tag{3.20}$$

Note both Lagrangian and Eulerian numerical methods are based on the assumption mass densities ρ_0 and ρ_1 are close enough. Indeed, the Lagrangian technique is founded on the linearization of the velocity map while the Eulerian approach improves the initial velocity guess by estimating the adequate correction term. Therefore, the implemented algorithms are more robust and present satisfying and fast convergence results for closed mass distributions. In [18], computational methods have essentially been validated against the literature such as the BB algorithm, on standard test cases with gaussian transportation (Fig. 3.11).

In particular, the authors highlighted how fast Lagrangian and Eulerian algorithms are (few seconds) with respect to the BB algorithm (a few minutes). In our case, the computational time is particularly important given the number of optimal transportation problems we need to perform on real-world data meaning high-resolution datasets. That is why, in what follows, we preferred those specific computational methods which focus on computational time at the expense of robustness or convergence accuracy. To conclude, the Lagrangian and Eulerian algorithms have shown we can find the optimal path

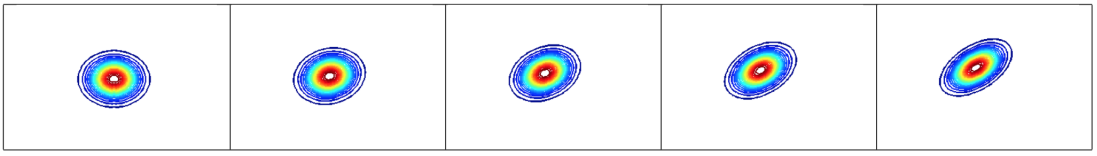


FIGURE 3.11: **Illustration of an Eulerian-based solution of optimal transportation between two gaussian distributions, from [18].** Notice the initial and final gaussians do have essentially the same support as the gaussian center is not translated. However, the final gaussian is deformed with respect to the initial density. The optimal path of the gaussian to transport (*i.e.* deform) $\rho(t=0)$ onto $\rho(t=1)$ is shown at $t=0.25$,$t=0.5$ and $t=0.75$, for a 200×200 grid size.

to deform a certain mass distribution onto the targeted deformation. Eventually, a more significant amount of real-world transportation tests, based on complex and large datasets could be considered to challenge the numerical algorithms.

3.3.3 Application: recovering the deformation velocity field

The previously described Eulerian approach has been used to extract the 2D velocity field from the experimental data of zebrafish locomotion. At the beginning of the present chapter, we specifically explained how experimental images were post-processed, segmented, and realigned *via* the *Procrustes Analysis* step. As a result, we obtained the overall silhouette of the zebrafish eleuthero-embryo at each image time step $t_{n_t} = n_t \times \Delta_i t$ (we refer the reader to Fig. 2.7). Two pieces of information are still required to complete each iterative time step (see the algorithmic sketch in Fig. 2.10): the deformation velocity $\mathbf{u}_\epsilon$ at each computational time step and the zebrafish silhouette. Actually, only the deformation velocity field required to deform the segmented eleuthero-embryo $\rho_0 = \rho(t = n_t)$ onto the target silhouette $\rho_1 = \rho(t = n_t + 1)$, is needed to interpolate the intermediate silhouette and velocity fields at time t^n. Indeed, the optimal path follows a straight line between times n_t and n_t+1 which enables straightforward optimal transportation interpolations. Thus, we tackled the initial question of recovering the 2D deformation velocity field (Fig. 3.12) with the optimal transportation theory and more precisely, the Eulerian algorithm developed by [18].

In definitive, we used optimal transportation theory mainly for two applications. On the one hand, we extracted a 2D velocity field for each video snapshot which is crucial to model the deformation velocity and to interpolate the intermediate zebrafish silhouettes, in the CFD code. On the other hand, we used the regularized optimal transportation for reconstructing the 3D shape of zebrafish eleuthero-embryo, based on a transverse-slice database. In particular, we implemented a massive amount of Wasserstein interpolations to generate the intermediate cross-sections. Indeed, the second application of optimal transportation is presented hereafter which concerns the interpolation of mass densities, required for reconstructing the 3D shape of a 5 dpf zebrafish eleuthero-embryo. In the next section, we detail the reconstruction of a 3D zebrafish shape from a massive amount

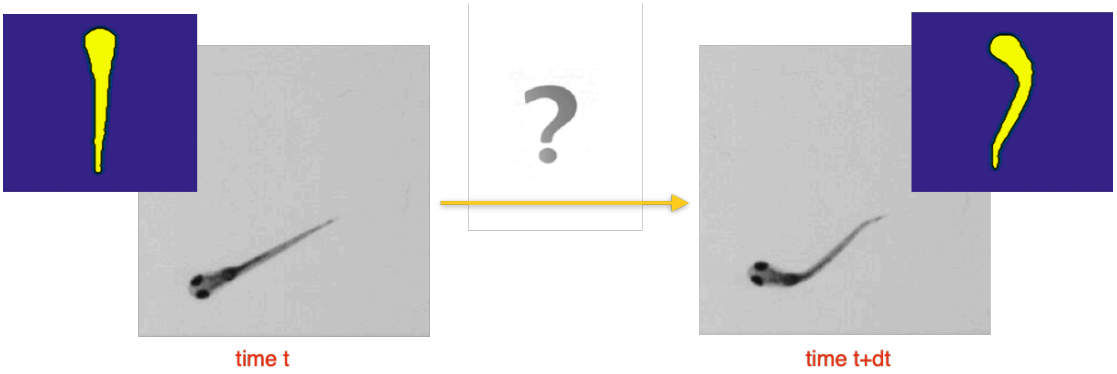


FIGURE 3.12: **Optimal transportation theory is used to estimate the deformation velocity between two consecutive snapshots ρ_0 and ρ_1 of zebrafish swimming.** The extraction of the deformation velocity field is based on the assumption all translation and rotation have been removed between the two snapshots. In addition, intermediate zebrafish deformations and velocity are reconstructed from experimental deformation velocity fields. This figure directly follows the previous computational sketch introduced in the previous chapter (Fig. 2.7).

of histological cross-sections. To that end, we especially used Wasserstein interpolations to generate the intermediate transverse shape, the complete procedure being presented in Appendix A. Essentially, the interpolation of cross-sections is based on efficient Sinkhorn iterations [109] which approximate a regularized solution *i.e.* a cross-sections with smoothed contours. In what follows, we will show that the impacts of regularization are minimal and eventually negligible.

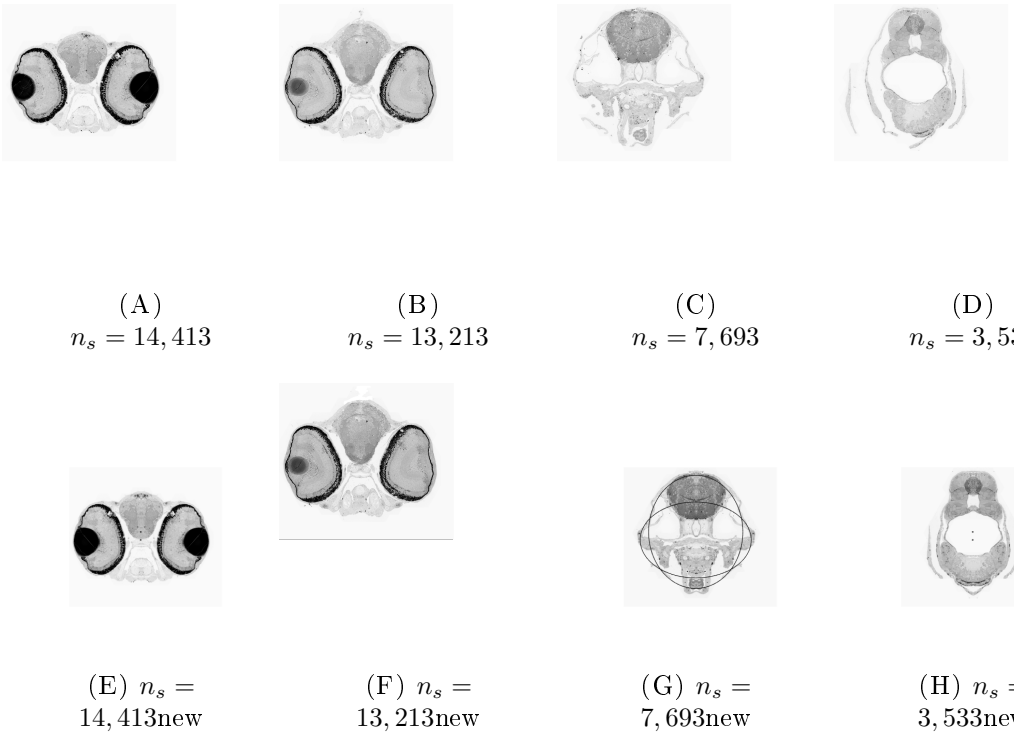
3.4 3D Reconstruction of the entire swimming motion

To run 3D simulations, we needed to reconstruct in 3D the 2D segmented video frames to provide to the CFD code, a set of 3D snapshots describing the zebrafish body kinematics. Two steps were fundamentals. First, we needed to generate a full 3D zebrafish model which describes accurately the morphology of actual zebrafish eleuthero-embryos. Secondly, we needed to reconstruct the zebrafish model for each deformed 2D silhouette corresponding to each video frame. The latter step was not straightforward and is presented with the whole methodology process: from the initial targeted implicit approach to the definitive Lagrangian reconstruction process. Note this crucial step could also be considered in 2D by reducing the initial 3D zebrafish modeling to a 2D zebrafish model. Besides, we introduced in the previous Chapter 2 the two body representations we used: either an implicit volume *via* a level-set distance function or an explicit description of the body surface *via* Lagrangian markers (see Fig. 2.5). The choice of body representation was essentially determined by the methodology we use to deform the zebrafish model. That is why in the first step, we initially reconstructed the 3D body volume using voxels which will be used to compute the 3D level-set function before constructing the Lagrangian markers as presented in the previous chapter (Fig. 2.5).

3.4.1 Initial 3D reconstruction

We first generated a full 3D zebrafish based on serial histological cross-sections to describe the characteristic shape of zebrafish eleuthero-embryo. The base idea was to obtain a zebrafish volume by concatenating multiple transverse slices from the snout to the tail. In particular, the head of a 5.5 days post-fertilization zebrafish was reconstructed up to the pectoral fins from high-resolution serial sections generated by [52]. Indeed, Hildebrand et al. [52] generated a massive amount of transverse slices for two zebrafish eleuthero-embryos aged of 5.5 and 7 dpf to analyze the brain sections and the larval neuronal system. Eventually, the authors obtained cutting-edge results by using advanced electronic-microscopy imagery. The high-resolution database is an open-access resource which was made available by *NeuroData* and hosted at `http://neurodata.io/data/hildebrand16` and `http://zebrafish.link/hildebrand16` [52]. We chose to base our 3D reconstruction on the generated database as no equivalent imagery exists to our knowledge and we desire to compute a real-like zebrafish eleuthero-embryo shape. The database is composed of $18,627$ high-resolution cross-section images uniformly spaced of 60 nm and widespread from the snout to the pectoral fins included which constitutes the most complex part, approximately one-third of the entire body of the zebrafish eleuthero-embryo morphology. By using high-resolution electronic microscopy, each section was composed of 1024×1024 pixels to represent a window of 758.8 nm $\times$ 1024 nm. For experimental reasons, certain cross-sections have been eliminated due to too weak quality which reduced the number of slices to $16,212$. As a result, some sections were regularly missing across all the dataset. Nevertheless, we obtained a satisfying enough precision of the description of the zebrafish morphology by using only 1 slice out of 80 slices. In definitive, we used 203 high-resolution sections, uniformly spaced of $\Delta z = 4.8\ \mu m$.

An image processing step was then required to segment all cross-sections into body silhouettes meaning binary masks as only the exterior contour needs to be captured. Indeed, the interior anatomy was superfluous and useless to reconstruct the body shape. Basically, an intensity threshold was found sufficient to generate proper binary masks of the transverse contours. However, specific corrections have been required for several cross-sections. First, the contour of multiple transverse images was not closed, especially regarding the natural gill openings. Large openings raised issues as morphological operations were not sufficient to complete the segmentation. In that case, we set up and superposed two ellipses to artificially close each body boundary adequately. Eventually, one side of the contour was slightly detached within a few slices located near the beginning of pectoral fins, which artificially deforms the body silhouette. The detachment, probably due to experimental manipulation, was much more significant on the left side of the body. As a result, we symmetrized the whole database to directly enforce a generic symmetry of the reconstructed eleuthero-embryo's geometry. The symmetry axis was found by using the so-called Hough algorithm designed to find any symmetry property.



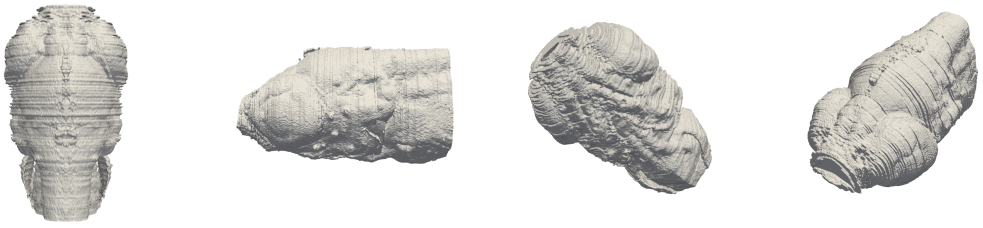
(A) $n_s = 14,413$ (B) $n_s = 13,213$ (C) $n_s = 7,693$ (D) $n_s = 3,533$

(E) $n_s = 14,413$new (F) $n_s = 13,213$new (G) $n_s = 7,693$new (H) $n_s = 3,533$new

FIGURE 3.13: **Few examples of image corrections applied to the original dense dataset.** First row refers to original data while second row denotes the corresponding transformed transverse slices. First column (Figs. 3.13 A, 3.13 E) shows the cross-section issued from the eleuthero-embryo's retina, with axial mirroring. Second column (Figs. 3.13 B, 3.13 F) represents a specific case with an artificial mass on the top of the head, removed before mirroring. Third column (Figs. 3.13 C, 3.13 G) illustrates the lack of contour information for segmenting the cross-section and the approximation used. Pectoral fins are visible in last column (Figs. 3.13 D, 3.13 H) alongside the large mass detachment on the left side, mirrored.

Obviously, only the proper half of each section was conserved and mirrored. Moreover, specific imperfections needed to be manually corrected on a few slices only. Note the pectoral fins have not been removed and have been untouched for each corresponding slice. Few examples of slice transformations are given in Fig. 3.13 and the resulting volume of the subset of enhanced transverse sections, is detailed in Fig. 3.14. Note the impact that a slight interfering mass can have without specific correction.

At this developmental stage, the morphology is characterized by the anterior part including the head and the anterior part of the trunk. As for the most posterior part of the trunk, from the pectoral fins to the tail tip, the general shape is much less complex and much more generic. We assumed the main differences from one cross-section to another one, were the position of the center of mass of the section and the vertical height of the body as well as the MFF *i.e.* the thin, transparent and typical membrane formed of few cells only which surrounds the posterior section and particularly the tail of zebrafish eleuthero-embryos. At 5 dpf, a zebrafish eleuthero-embryo sizes around 3.8 mm (total length). Considering the space step of the 203 anterior cross-sections, 595 additional slices were required on the posterior part to obtain equidistant spacing across



(A) top view. (B) lateral view. (C) perspective view of the bottom. (D) perspective view of the top.

FIGURE 3.14: **Detailed views of the original dataset after slice processing.** Operations essentially consisted of segmenting, mirroring, realigning, resizing, enhancing and correcting the contour. Concatenation of a binary transformed dataset. Note the artificial mass on the top of the head (emphasized in Figs. 3.13 B, 3.13 F) is well-visible if not removed.

the total length. To that end, we used a lateral view of a 8 dpf zebrafish larva (Fig. 3.15), as a precise view was available at that time including in-house histological transverse section dataset (data not shown). Based on that view, we manually constructed up to 34 transverse sections (Fig. 3.16) at key locations of the trunk and tail, regarding the variation of body morphology to automatically interpolate the remaining set of cross-sections. In fact, the posterior part of a 5 dpf zebrafish eleuthero-embryo can be well-characterized by the key transverse slices of a 8 dpf zebrafish larva, after scaling. The developmental difference was considered as not important as the larval posterior part is generic enough and an adequate scaling process was considered afterward. From the lateral view, we extracted two sets of information: the vertical position of the center of mass of each transverse slice, and the heights of MFF parts proportionally to the body size. As for the profile of the cross-sections, we identified four typical silhouettes: the body near the pectoral fins, the body surrounded by the thin membrane with eventually the digestive apparatus accordingly to the position, and the single thin membrane which forms the tail tip. The base shape of the body was based on the last transverse section of the database and two transformed histological slices. The MFF was modeled as a vertical line with a very thin width (few pixels). We chose the position of the 34 key slices along the body length depending on the significant changes of the body morphology *i.e.* heights variations. Furthermore, all slices including the anterior cross-sections have been resized (300×300 pixels corresponding to $\Delta x = 1.68622$ μm), realigned onto the respective center of mass, and vertically shifted according to the profile view.

As explained, the remaining slices then need to be interpolated from the key cross-sections. We chose to incorporate the anterior transverse slices to the key slices to recover the total zebrafish 3D volume by essentially computing any desired number of transverse slices uniformly spread across the body length. Obviously, the space step could not be greater than the current space step of anterior sections $\Delta z = 4.8$ μm

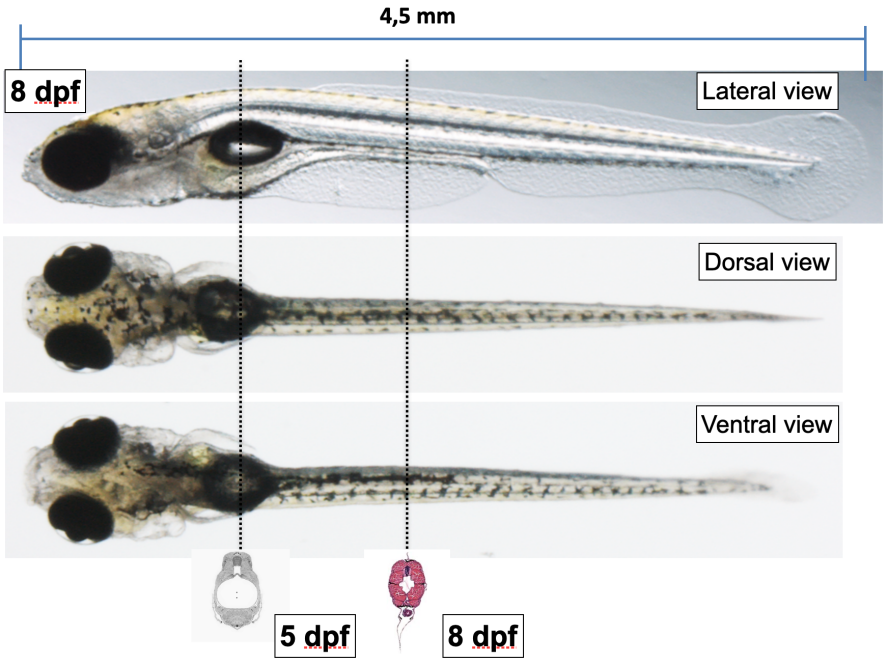


FIGURE 3.15: **Morphology overview of a 8 dpf zebrafish larva.** From the three original lateral (top), dorsal (center) and ventral (bottom) views of a 8 dpf zebrafish larva, we mainly used the lateral profile to dimension each key slice according to the vertical position, and the median finfold and body proportions. The shape of the body in key slices has been based on the histological slice represented on the posterior part of the body. Note the histological head sections from the snout to the position indicated near pectoral fins, were generated from an high-resolution dataset of cross-sections of a 5 dpf zebrafish eleuthero-embryo, as illustrated by the last available cross-section. Eventually, the length-height ratio was adjusted to fit a 5 dpf zebrafish morphology.

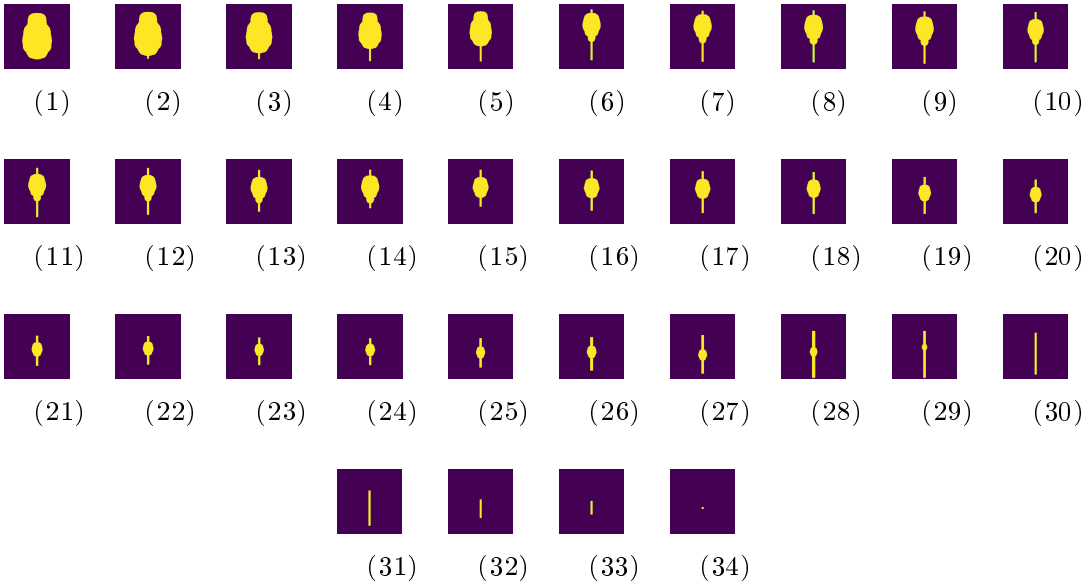
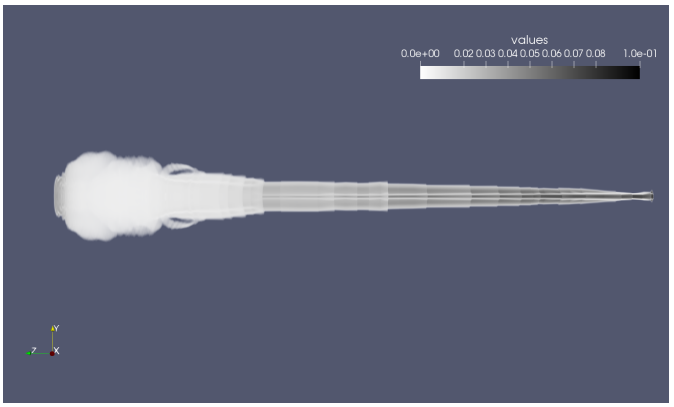


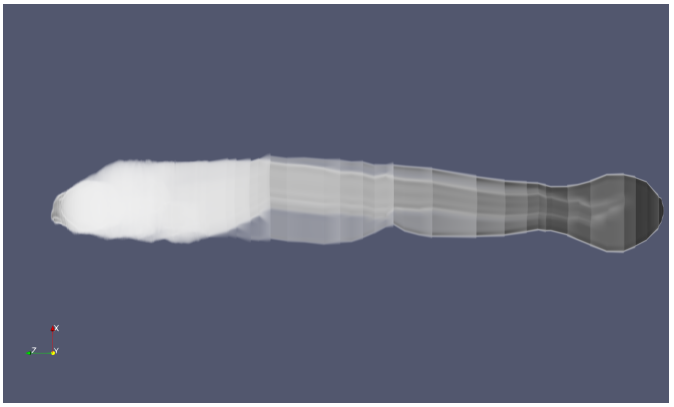
FIGURE 3.16: **Series of the 34 generated key cross-sections used to reconstruct the whole posterior part of the zebrafish eleuthero-embryo.** Note the presented transverse slices are well-aligned vertically but are not uniformly spaced along the eleuthero-embryo's length. Instead, each slice corresponds to a precise location. Notice the key slices enable the interpolation from the nearby pectoral fins (Fig. (1)) to the tail tip (Fig. (34)).

to conserve the original precision. Therefore, the final space step had to be smaller to show the current space step is not fixed and can be tuned. That is why, we chose a space step twice finer $\Delta z = 2.4\ \mu m$ which led to a stack of 1602 transverse slices. As for the interpolation, we used an innovative shape interpolation method based on the computation of regularized barycenters of Wasserstein distances *via* the Sinkhorn algorithm, described in Appendix A. Basically, we imposed the number of cross-sections to interpolate between each consecutive key slices. To that end, we especially used Wasserstein interpolations to generate the intermediate transverse shape. Therefore, each interpolated slice was considered as the barycenter of the consecutive slices in the sense of Wasserstein distance and we performed multiple Wasserstein interpolations (A.15) for each consecutive segments $[v_k; v_{k+1}]$. Unlike linear interpolation, optimal transportation results can produce natural interpolations, as explained in the previous section. The Sinkhorn algorithm is particularly well-designed to shape interpolations when no explicit description of the surface is available. Instead, the surface is implicitly generated *via* the regularization of mass density distributions *i.e.* the mass of the body in each cross-section. As a result (Fig. 3.17), we obtained smoothed cross-sections and the regularization differed for each interpolation segment due to geometric differences across the transverse slices (v_k). Notice the regularization parameter γ was fixed at $\gamma = 7.000 \times 10^{-4}$ for all interpolation segments. Cross-sections were found too smoothed with $\gamma = 1.000 \times 10^{-3}$ while convergence issues were raised at $\gamma = 5.000 \times 10^{-4}$. Indeed, numerical issues are raised in the Gaussian kernel in the limit of $\gamma \to 0$. In the literature, generalizations of the Sinkhorn algorithm have been developed to stabilize the Sinkhorn iterations [103] or accelerate the convergence rate [114]. However, the differences of regularization and mass variations were not a problem in our case although small discontinuities were visible (see Fig. 3.17). Indeed, the resulting output was then post-processed to normalize each cross-section density and segment with a simple threshold, the final zebrafish 3D reconstruction.

Overall, we obtained a 3D volume represented by $1602 \times 300 \times 300$ voxels in which the body surface was regularized according to each consecutive Wasserstein interpolation. The surface of the resulting 3D volume is then diffuse and somewhat discontinuous between near each key section, even after normalizing each transverse slice with the maximum density. Nevertheless, the interior mass was nearby one while the outer voxels were mostly zeros as the initial key slices had been segmented into binary images (masks). To conclude, we used a threshold to properly segment the resulting 3D volume describing the zebrafish eleuthero-embryo morphology. The discontinuities have been neglected regarding the massive amount of cross-sections used. Besides, the pectoral fins have been surprisingly well-reconstructed by optimal transportation. Notice the optimal transportation algorithm computed very satisfying results with non-compact density distributions (pectoral fins) especially near the transitional areas *i.e.* the apparition and vanishing of pectoral fins. The last step consisted of scaling the resulting 3D volume according to a 5 dpf zebrafish eleuthero-embryo based upon the ratio of height over



(A) top view.



(B) lateral view.

FIGURE 3.17: **Illustration of the Sinkhorn algorithm output in the case of a zebrafish shape interpolation.** Both top-view (Fig. 3.17A) and side-view (Fig. 3.17B) show the impact of the regularization of transverse slices. Moreover, we can see how variable the final density is within each cross-section. Besides, a slight and variable discontinuity can be observed between each consecutive interpolation segment. Notice the 3D surface is well-approximated by post-processing the represented output accordingly. Note we thresholded the maximal density to $value = 0.1$ for the illustration purpose.

length which resulted in a 5 dpf zebrafish volume (Fig. 3.18) with a 3.834 mm total length and formed by voxels sized of $2.2517^2 \times 2.4\ \mu m^3$. In definitive, we computed a stack of transverse silhouettes which constituted the first 3D reconstruction of a 5 dpf zebrafish eleuthero-embryo. Note the computed 3D representation is implicit as the precision on the surface entirely relies on the amount of voxel.

After the reconstruction of a first 3D volume, the deformed shapes corresponding to each video frame were required as input to the CFD code. The first initial way to deform the 3D zebrafish model is briefly presented although it has not been a success. Afterward, the final deformation process will be entirely described.

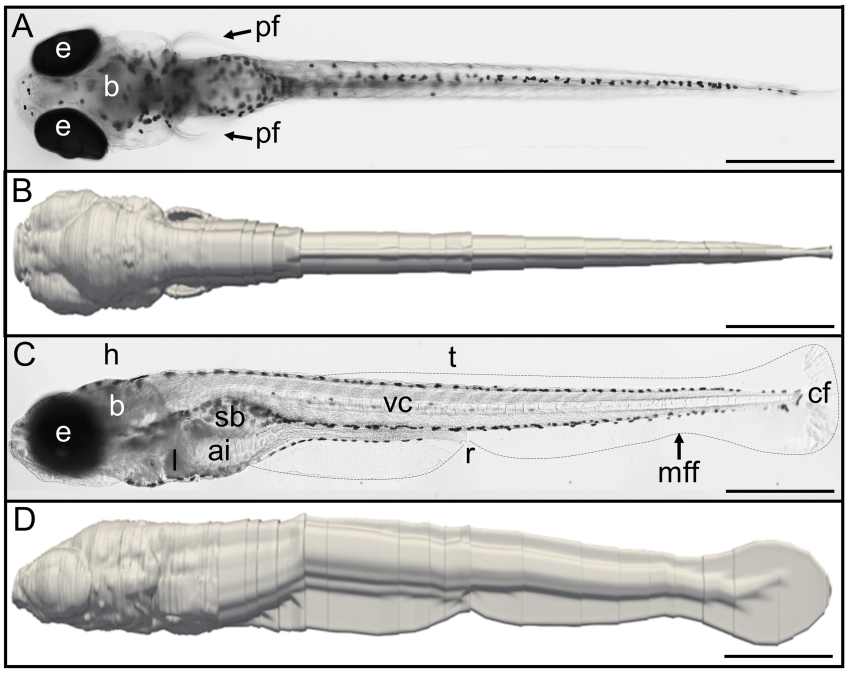


FIGURE 3.18: **Comparison between our 3D reconstruction (Figs. (B),(D)) and the photos of an actual 5 dpf zebrafish eleuthero-embryo morphology (Figs. (A),(C)).** In the dorsal views, we can notice the symmetry and the position of pectoral fins while in the lateral view, the median fin fold (mff) contour is well-reproduced. Besides, slight discontinuities are visible in both representations which denote the shape interpolation segments. Representative 5 dpf zebrafish eleuthero-embryo is shown in dorsal view (A) or in lateral view (C) with the anterior part to the left. The median fin fold (mff) is highlighted by a dashed line. Other abbreviations: ai, anterior intestine; b, brain; cf, caudal fin; e, eyes; h, head; l, liver; pf, pectoral fins; r, rectum; sb, swim bladder; t, trunk; vc, vertebral column. Scale bar, 500 μm.

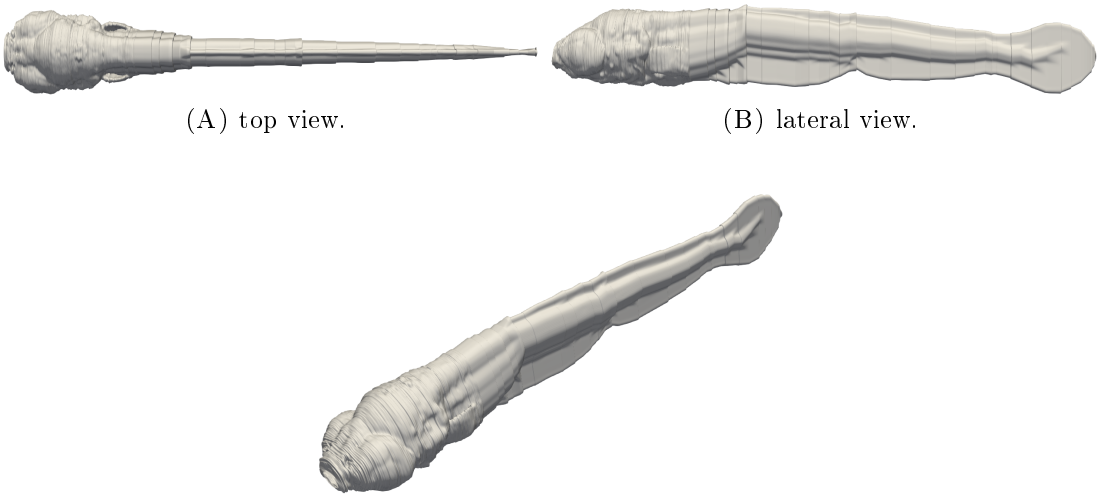
3.4.2 3D Deformations: a failed process

How can we deform a 3D volume according to experimental kinematics ? In the previous section, we already extracted the body kinematics from the top-view experiment videos. At the same time, we assumed and modeled the entire swimming motion as horizontal and planar, only meaning no depth motion is present. As a result, we could consider the deformation is actually identical within each horizontal plane across the vertical axis and well-represented by the 2D velocity field resulting from the optimal transportation computation, as we considered the zebrafish body entirely rigid. That is why, we initially desired to deform each horizontal plane of the 3D reconstructed model according to the 2D top-view deformation velocity field. The idea was *a priori* feasible but was essentially based on two assumptions. First, the deformation velocity had to be highly accurate and embed much more constraints than the previous usage: an input velocity field to the 2D CFD code. Secondly, the actual transportation scheme for deforming each horizontal section had also to be high-order accurate. In fact, none of these two key conditions was realized with the current algorithms. Let unfold the initial deformation process. First, the resolution and dimensions of the 3D computed volume formed by $1602 \times 300 \times 300$ voxels, had to be matched against the video frames from which we

extracted the deformation velocity field (200×200 image resolution). In that regard, we transformed the 3D computed volume accordingly while reducing the number of cells to 200^3 for data-reducing reasons. As for the deformation itself, we used the implicit technique introduced in the previous Chapter 2: the level-set method. We computed the level-set distance function from the segmented 3D volume with the *Russo and Smereka* algorithm [101] which we enhanced by using a classic WENO5 scheme to compute the first-order upwind derivatives and generalized to three dimensions. Basically, equation (2.10) is discretized as:

$$\phi_{i,j}^{n+1} = \phi_{i,j}^{n} - \Delta t S(\phi_{i,j}^{0}) G(\phi) \tag{3.21}$$

where $\phi_{i,j}(t=0) = \phi_{i,j}^{0}$, $S(\phi) = sgn(\phi)$, $G(\phi) = \|D^{upwind}\phi_{i,j} - 1\|$ and D^{upwind} denotes a combination of upwind (x, y)-first-order derivatives computed with the WENO5 scheme. In definitive, the *Russo et Smereka* algorithm computes the signed distance function ϕ such as $\nabla\phi = 1$ with an iterative algorithm, initialized with the 3D mask ϕ^0 describing the zebrafish body with binary voxels.



(A) top view. (B) lateral view.

(C) perspective view.

FIGURE 3.19: **Representations of the 3D reconstructed 5 dpf zebrafish eleuthero-embryo volume (level $\phi = 0$).** Notice slight discontinuities are visible in the top (Fig. 3.19 A) and side (Fig. 3.19 B) views as the volume is formed by 1602 cross-sections. With the perspective view (Fig. 3.19 C), we can see how detailed the final representation is.

As for transporting the level-set function, we considered the following advection equation:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \tag{3.22}$$

After the level-set computation, the actual contour is accessible *via* the level zero $\phi = 0$. We used the level-set function to only transport the interior points of the body. The goal was essentially to transport each horizontal plane of the level-set function according to equation (3.22). Indeed, we aim to transport the initial 3D level-set function to the final deformed image from the experimental video which represents more than

60 ms in real-time. When reaching each frame of the video, we use the corresponding deformation velocity previously computed. As for the advection scheme, we implemented the remeshing particle advection methods introduced in the previous Chapter 2 and used in section 3.3.2, which are designed in particular to solve Euler equations (see [132, 133]). In the case of incompressible flows *i.e.* $\nabla \mathbf{u} = 0$, equation (3.22) is equivalent to the transportation equation (2.7). Nevertheless, optimal transportation does not generally enforce incompressibility conditions. That is why, equation (2.7) could not be used *a priori* to transport the level-set function. However, we compared our remeshing particle algorithm with a more standard numerical scheme for a validation purpose. In particular, we first solved equation (2.7) on basic 2D test cases by using an incompressible velocity field and a WENO5 spatial scheme in order to minimize spatial inaccuracy, coupled with a 3^{rd}-order Runge-Kutta scheme (RK3) in time. As explained in the previous Chapter 2, the signed distance property tends to not be conserved over time. That is why, we regularly recomputed the distance function. Based on the few standard test cases used, both methods provided a global second-order accuracy in space which was sufficient to use the remeshing particle method with the velocity field issued from optimal transportation calculations.

In particular for remeshing, we used various interpolation kernels such as kernels M_4 or M_4' (see Eq. 3.20) to reach a second-order interpolation (M_4) and third-order interpolation (M_4'). Note M_4 is more diffusive and M_4' is well-popular for CFD particle methods [132, 133]. However, we could not obtain a satisfying deformed 3D volume as we rapidly observed unexpected behaviors: all the horizontal slices rapidly converged toward an identical silhouette, after hundreds of image steps (few tens of ms). We replicated the 3D advection issues in the 2D case by transporting the initial top-view to match the intermediate reference snapshot sequence used for computing the deformation velocity with optimal transportation (see Figs. 3.20, 3.21). First, we only transported the initial snapshot of the reference sequence (Fig. 3.20). Similarly, the body silhouette could not have been conserved after only hundreds of image steps $\Delta_i t$, when using the M_4'-kernel. Nevertheless, we were able to observe the effective motion of the interior particles of the body according to the deformation velocity field. In particular, we observed the interior body particles tended to aggregate near the body contours used for computing the velocity field. Concerning the M_4-kernel-based advection, we observed intermediate snapshots were in good accordance with reference data. Besides, no spurious interior displacements seemed to be produced. However, horizontal slices do not correspond to the initial top-view of the reference sequence. Therefore, we applied the same particle advection algorithms to an arbitrary initial horizontal slice (Fig. 3.21). As a result, the initial contour was clearly not conserved, regardless of the interpolation kernel used. In addition, the contours of intermediate reference snapshots seemed to always be recovered. Furthermore, we observed non-desired spurious behaviors from the deformation velocity field such as the transportation of mass particles between the head and the tail tip when in proximity. In definitive, any transportation of a compact

mass density non-corresponding to initial reference silhouette produced spurious effects due to the deformation velocity field. In fact, optimal transportation does not generally enforce incompressibility conditions. As explained in Appendix A, algorithms can be developed to generalize the original optimal transportation problem by including an incompressibility penalization for example. We investigated those algorithms to enhance the optimal transportation solution by adding incompressibility or rigidity properties but the adaptation to our current algorithm (described in the previous section) was not straightforward. Actually, those questions constitute the focus of a whole class of optimal transportation investigations, previously described in section A.1.

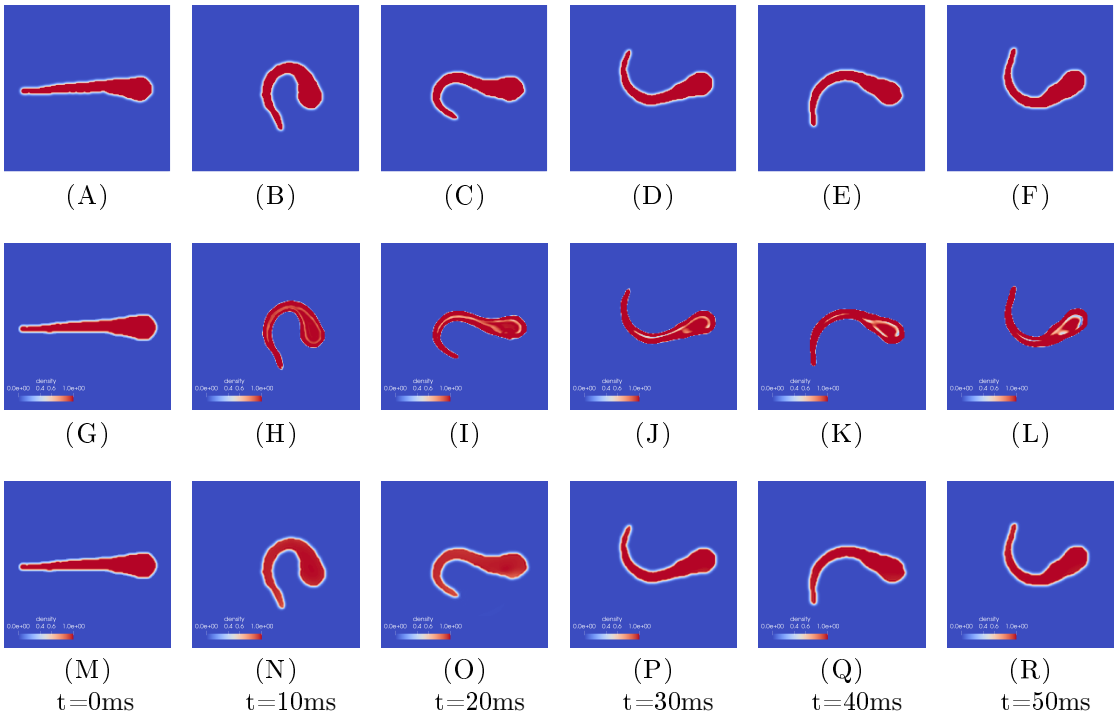


FIGURE 3.20: **Comparison in 2D of level-set transportation using remeshing particle methods.** The top row (Figs. 3.21 A-3.21 F) represents the reference sequence (15,000 fps) used for computing the deformation velocity with optimal transportation. The middle row (Figs. 3.21 G-3.21 L) illustrates the transportation of the initial distribution (Fig. 3.21 A) with the M'_4-interpolation kernel. As for the last row (Figs. 3.21 G-3.21 R), the M_4-interpolation kernel was used instead. To validate the deformation velocity and the advection algorithm, intermediate transported snapshots should match the reference data.

In definitive, the deformation velocity $\mathbf{u}_\epsilon$ has been computed between each consecutive 2D frames ρ_0 and ρ_1 *via* an optimal transportation algorithm. In 2D, the deformed silhouettes were directly provided by the segmented frames ρ. By construction, the deformation velocity was only used for the computational times between ρ_0 and ρ_1, in particular to interpolate the intermediate silhouettes. Instead, in 3D, several deformation velocity properties especially the interior velocity field, have been required to conserve the slice contour at each vertical plane across time. Moreover, the accumulation of numerical errors was inevitable regarding the number of total time steps required to transport from the initial shape to the final video time. As a consequence, we considered

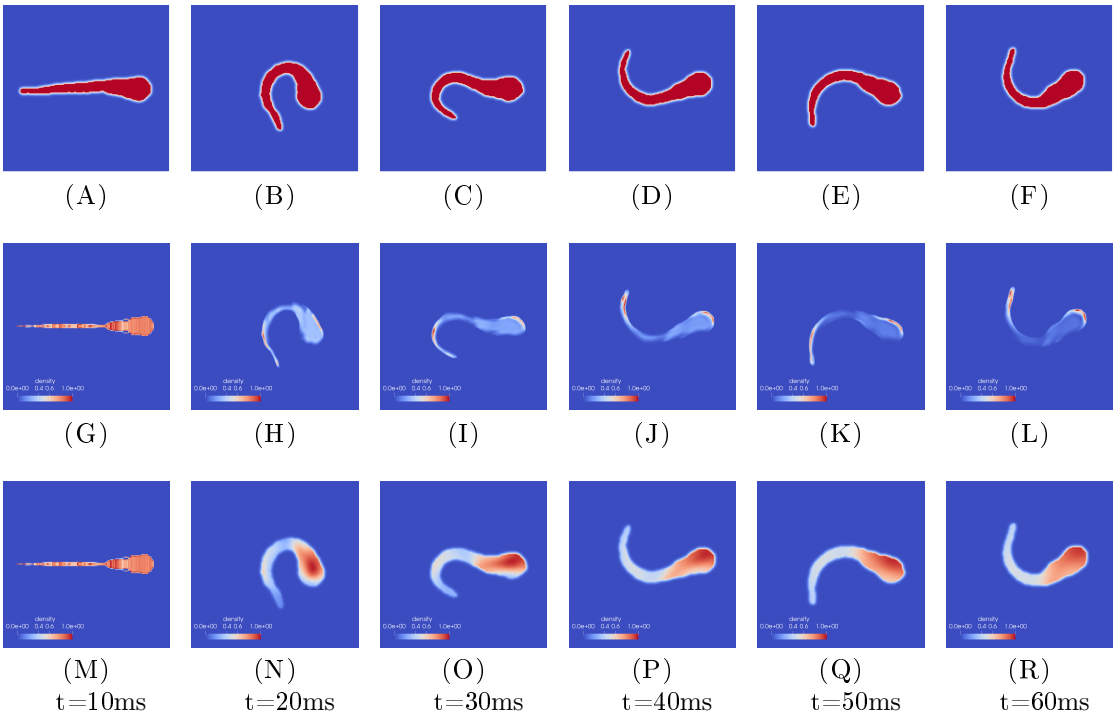


FIGURE 3.21: **Comparison in 2D of level-set transportation using remeshing particle methods, based on an arbitrary initial silhouette.** The top row (Figs. 3.21 A-3.21 F) represents the reference sequence (15,000 fps) used for computing the deformation velocity with optimal transportation. The middle row (Figs. 3.21 G-3.21 L) illustrates the transportation of the initial distribution (Fig. 3.21 A) with the M_4'-interpolation kernel. As for the last row (Figs. 3.21 G-3.21 R), the M_4-interpolation kernel was used instead. To validate the deformation velocity and the advection algorithm, intermediate transported snapshots should match the reference data.

an alternative approach to deform the 3D volume across time based on the experimental midline kinematics.

3.4.3 3D Deformations: a Lagrangian idea

The essence of the new approach was based on the construction and tracking of Lagrangian markers across the zebrafish surface. In the previous chapter 2, we briefly explained how to recover the signed distance function from Lagrangian markers. At this stage, we have already computed the 3D zebrafish level-set function from the initial volume. Besides, there is no need in resizing the number of voxels to 200^3, unlike the previous approach. Thus, the level-set function has been computed with a $2000 \times 300 \times 300$-sized domain to conserve the initial high precision. Note the length of the reconstructed volume is parallel to the axes such that the midline corresponds simply to a straight line from the snout to the tail tip, fitting the Cartesian mesh excepted at extremal points where the body boundary position are interpolated from the level-set information at the mesh nodes. As for the description of the body surface, we used

the structured representation detailed in the previous chapter 2, which consisted of discretizing the midline in N_s uniformly-spread transverse slices and discretizing the contour of each cross-section into N_θ points distributed across 360° with a constant angular frequency. For that purpose, we first interpolated the body contour ($\phi = 0$) of each transverse slice fitting the Cartesian grid using bilinear interpolations of the level-set function in the near-interface cells (see Fig. 3.22). For accuracy reasons, we combined the interpolation with a dichotomy process to find the body interface. In total, we needed $N_\theta - 1$ uniformly distributed surface points per cross-section.

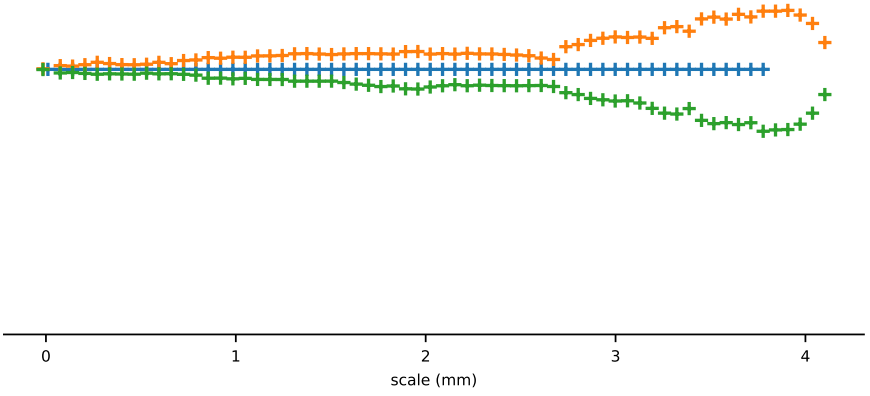


FIGURE 3.22: **Construction of control points using the Cartesian grid and the level-set function.** For illustration purpose, we only represented 1/25 control points including the surface and midline points. Note the *green*-line denotes the horizontal section $\theta_1 = \theta_{N_\theta} = 0$ while the *orange*-line almost represents the symmetric points as the sectional plane corresponds to $\theta_{N_\theta/2}$. In that way, we only represent two surface points per cross-sections. That is why, we do not particularly notice how symmetric the 3D reconstruction is.

Secondly, we interpolated the midline and the $N_\theta - 1$ lines from the snout to the tail tip, into N_s points using B-splines namely Bezier curves. To be more precise, B-splines are not interpolation curves but approximation methods. Bezier curve starting and ending points correspond to the extremal points of the set to approximate. The tangents at starting and ending points point toward the neighbor point. The $N_s - 2$ remaining points are computed from the so-called De Casteljau algorithm which consists of a recursive algorithm to compute $N_s - 2$ barycenters. Indeed, let assume we know N_c control points which define $N_c - 1$ segments and we desire to find the coordinate of the mid-point ($N_s/2$) in the Bezier curve. We can define the mid-point of each segment to define $N_c - 1$ new control points. After $N_c - 1$ iterations, we obtain the desired point of the Bezier curve.

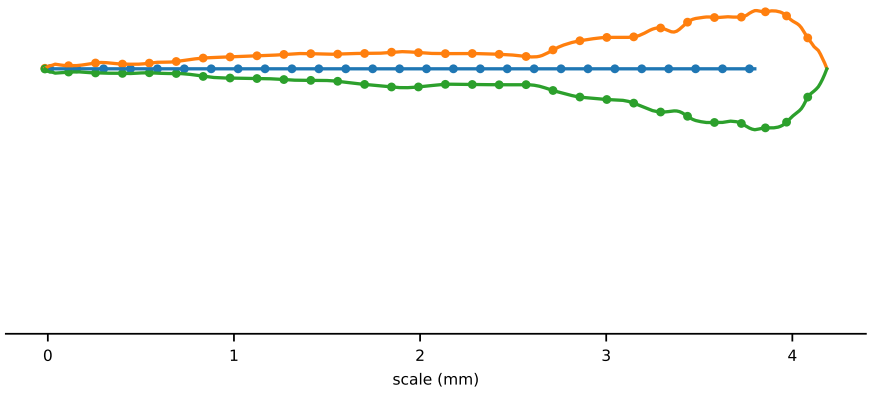


FIGURE 3.23: **Representation of the B-spline approximations for two surface longitudinal lines.** As in Fig. 3.22, we only represented two $\theta_i (i = 1, N_\theta/2)$ per cross-sections, which are almost opposed. Note the generated Lagrangian markers are not specifically inside each cross-section but regularly spaced along the surface geodesic to create uniform cells on the body surface. Note only 10% of all Lagrangian markers are represented, for indicative purpose.

After the B-splines computations, we obtain a surface mesh structured by the cross-sections. However, the resulting mesh was significantly non-uniform as the curvature was not taken into account. Indeed, let consider two different cross-sections: a circle and a vertical ellipse. For each case, we compute two joined Bezier curves. In the circular case, the barycenter at $t = 0.25$ is strictly located at $\theta = \pi/4$. However, the corresponding point in the elliptical case is located between $\theta = \pi/4$ and $\theta = \pi/2$, relatively to the cross-section curvature as the curvilinear abscissa travels more or less distance in the same time $t = 0.25$, to join $\theta = \pi$ from $\theta = 0$. We show an example with a transverse slice of the posterior part of the body (Fig. 3.24). Therefore, we enhanced the B-spline searching algorithm to obtain uniformly spread markers *in curvilinear distance* instead of the original constantly spread *in time* barycenters. The final reconstructed surface mesh was considered well-satisfying with $N_s \times N_\theta = 300 \times 180$ Lagrangian markers (see Fig. 3.23).

Note the resolution of the reconstructed 3D volume in terms of cross-sections (Fig. 3.25) *i.e.* the number of control points, is predominant in the resolution of the computed surface regarding a chose amount of slices (see Fig. 3.26). The main advantage of Bezier Curves resides in the approximation of 1602 control points instead of a direct interpolation which is not necessary. As a result, we obtain a well-smoothed surface explicitly described by Lagrangian markers (see Figs. 3.27, 3.28). Besides, note we only considered the most exterior body boundary to define the control points, essentially to

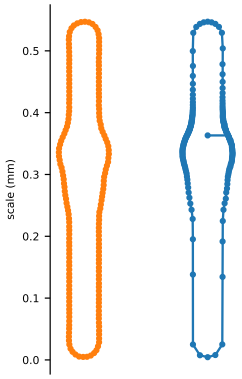
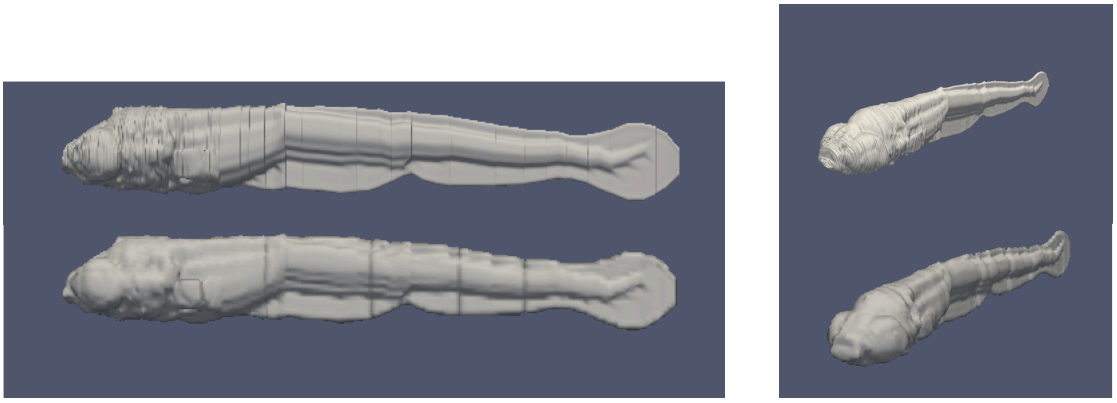


FIGURE 3.24: **Example of a cross-section in which each Lagrangian marker has been regularly spaced along the surface geodesic.** We can see how important a remeshing algorithm is, in particular for such transverse slices with large curvatures.



(A) (B)

FIGURE 3.25: **Comparison of two 3D reconstructions based on different spatial resolution.** We reduced the amount of cross-sections from the original 1602 transverse slices (top) to approximately 160 reconstructed slices (bottom). Note we compare the lateral view (Fig. 3.25 A) and the general perspective view (Fig. 3.25 B).

model the exterior of pectoral fins and then considering the pectoral fins as adducted to the body. In definitive, we constructed an explicit surface entirely described by a set of $N_s \times N_\theta$ uniformly-distributed Lagrangian markers, from the 3D level-set calculation. The resulting representation of the zebrafish eleuthero-embryo is horizontal and straight *i.e.* non-deformed, symmetric with a generic-like shape and sized of 3.834 mm length. The last step regarding every experimental zebrafish eleuthero-embryo consisted of adjusting the total length, including the median fin-fold represented herein, by scaling each dimension accordingly.

The second step was to reconstruct the 3D surface of each 2D video frame meaning deforming the initial 3D surface according to the observed 2D experimental deformations. More precisely, we considered each perpendicular cross-section stays perpendicular to

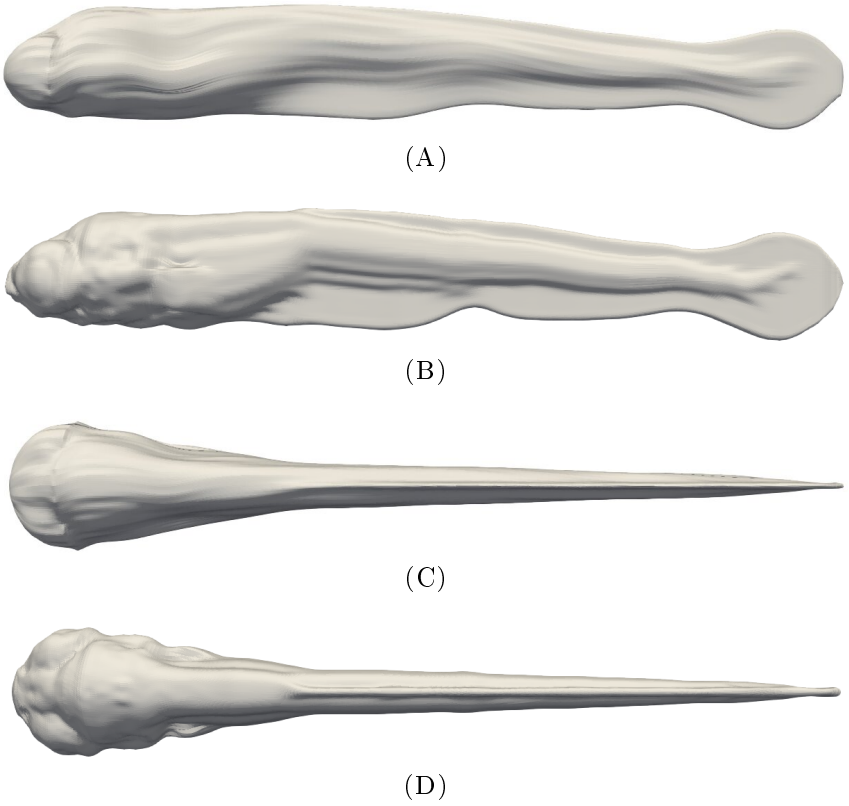


FIGURE 3.26: **Comparison of two 3D surface mesh based on different sampling size of control points.** We reduced the amount of control points form the original 1602 transverse slices (top) to approximately 160 reconstructed slices (bottom). Note we compare the lateral view (Figs. 3.26 A, 3.26 B) and the top view (Figs. 3.26 C, 3.26 D). For both 3D representations, we used the same surface mesh discretization: $N_s = 300$ slices and $N_\theta = 180$ angular directions for each cross-section.

the midline across the motion, similarly to Euler-Bernoulli beam deformation models. In that way, we consider each cross-section as entirely rigid meaning the body deformations could be reduced to the midline kinematics. Therefore, the complete deformation process performed for each video frame was straightforward: we re-positioned each rigid transverse slice perpendicularly to the deformed midline and regarding its curvilinear abscissa along the midline. Beforehand, the midline had to be computed from the top-view segmented experimental frames. Let detail the complete reconstruction, step by step. For each 2D deformed and segmented image, we first computed the corresponding level-set function (Fig. 3.29) using the *Russo and Smereka* algorithm from the 2D masks, as described previously. Afterward, we computed the midline from the gradient of the level-set function. Indeed, the gradient of the signed distance function (Fig. 3.30) is singular and discontinuous on the midline as the midline is located half-way from the body boundary, by definition. Nevertheless, the level-set function is only computed on a discrete and Cartesian grid formed by the image pixels. A first set of pixels have been computed *via* a threshold on the level-set gradient. Then, we performed a so-called thinning algorithm after eventually completing the set of thresholded pixels.

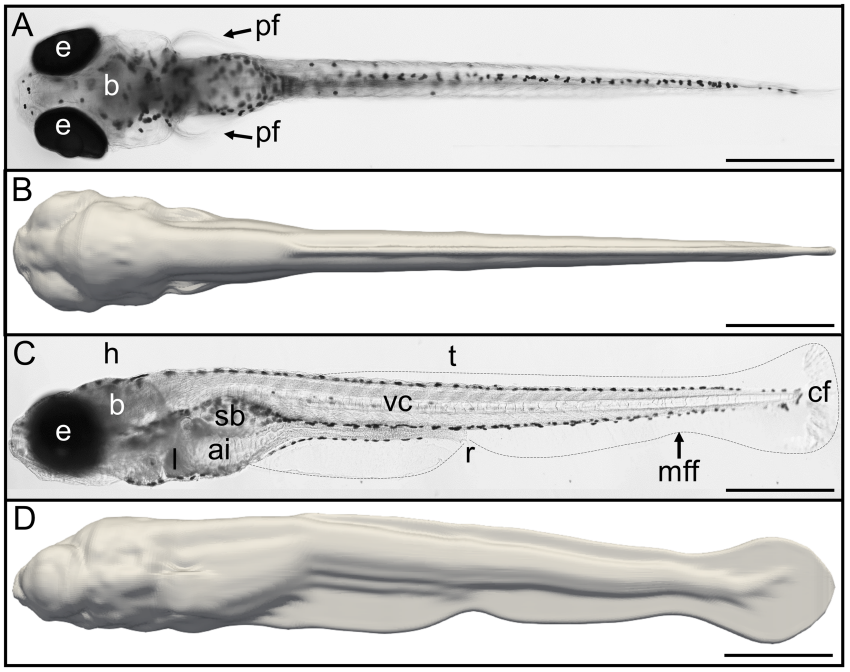
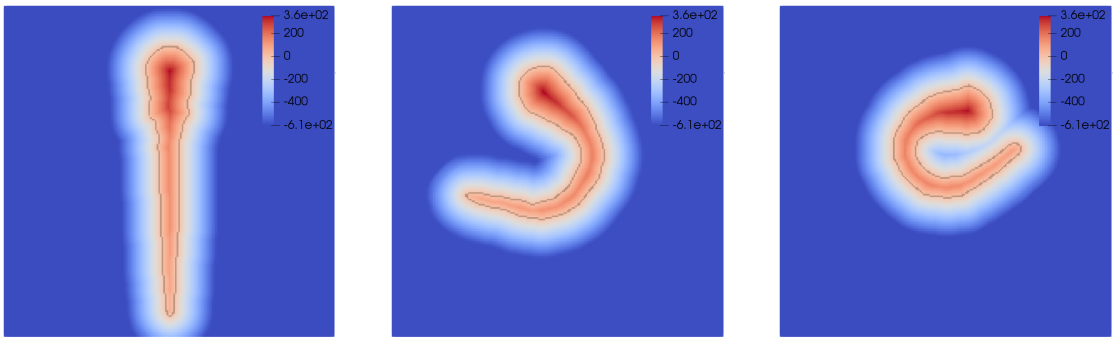
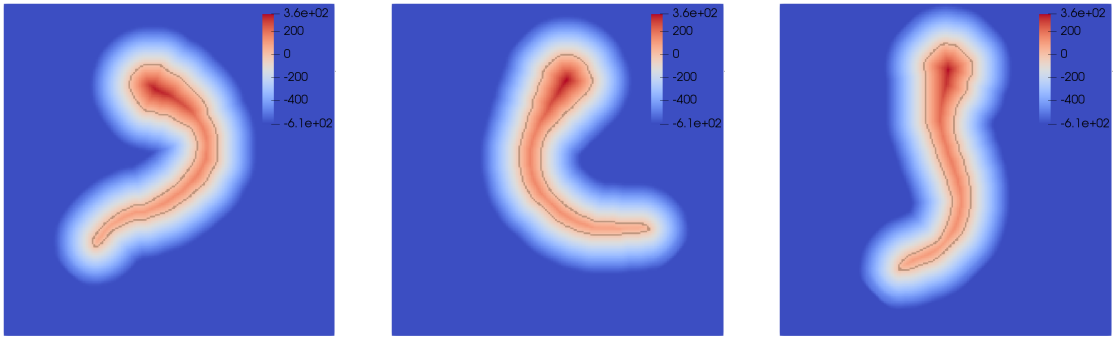


FIGURE 3.27: **Comparison between our 3D surface representation (Figs. (B),(D)) and the photos of an actual 5 dpf zebrafish eleuthero-embryo morphology (Figs. (A),(C)).** In the dorsal views, we can notice the symmetry and the position of pectoral fins while in the lateral view, the median fin fold (mff) contour is well-reproduced. Besides, notice how the surface is smoothed by the splines. Representative 5 dpf zebrafish eleuthero-embryo is shown in dorsal view (A) or in lateral view (C) with the anterior part to the left. The median fin fold (mff) is highlighted by a dashed line. Other abbreviations: ai, anterior intestine; b, brain; cf, caudal fin; e, eyes; h, head; l, liver; pf, pectoral fins; r, rectum; sb, swim bladder; t, trunk; vc, vertebral column. Scale bar, 500 μm.

FIGURE 3.28: **Perspective view of the reconstructed surface representation.** Surface mesh discretization: $N_s = 300$ slices and $N_\theta = 180$ angular directions for each cross-section.



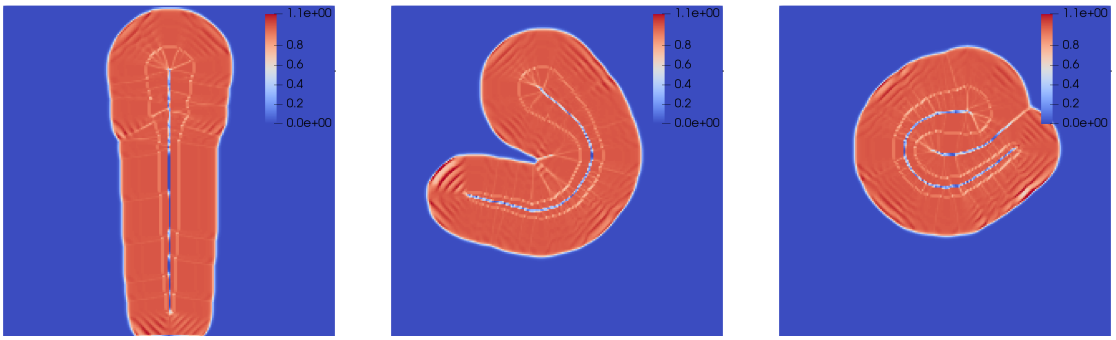
(A) $t = 0$ ms (B) $t = 10$ ms (C) $t = 20$ ms



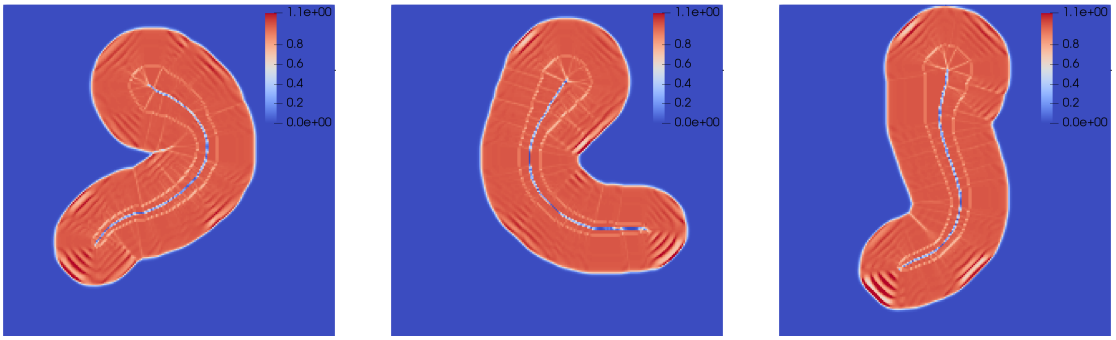
(D) $t = 30$ ms (E) $t = 40$ ms (F) $t = 50$ ms

FIGURE 3.29: **Computation of 2D level-set functions ϕ.** Outside of the body, $\phi < 0$ (blue) while $\phi > 0$ (red) defines the zebrafish mass density. The boundary is given by $\phi = 0$ (dark red).

To that end, we developed a custom algorithm which selects appropriate pixels according to the configuration of the midline and the distance to the body boundary, to reduce the midline to a 1-pixel wide line connected by the selected pixels (see Fig. 3.31). This step was important as the pixel-wise discrete midline was very sensible to the gradient threshold while the latter was fixed empirically - according to each experimental video. On the whole, the main concern which remains is the precision of extremal points. Indeed, only a subset of the actual midline could have been found with this method as the singular line of the level-set gradient cannot be defined straight from the tail tip to the snout of the head. However, the remaining parts of the midline could be described by raw segments to the extremal points of the body as we can legitimately model the head kinematics with a rigid motion. As for the tail, the rigid description is clearly acceptable considering the close proximity to the tail tip. Three operations remained to properly compute the midline. First, we normalized the total length by adjusting the length of the tail segment to match the non-deformed body length. Secondly, the head extremity of the pixel-wise discrete midline which roughly corresponds to the center of the head, required special attention. Indeed, the position of the head extremity was hardly accurate. Our whole deformation process is fundamentally based on the assumption the midline can be tracked across time. In particular, the extremity of the midline needed to be tracked accurately to avoid spurious phenomena. To that end, we developed an algorithm based on the level-set gradient (Fig. 3.32) to automatically track and correct the ending point



(A) $t = 0$ ms (B) $t = 10$ ms (C) $t = 20$ ms



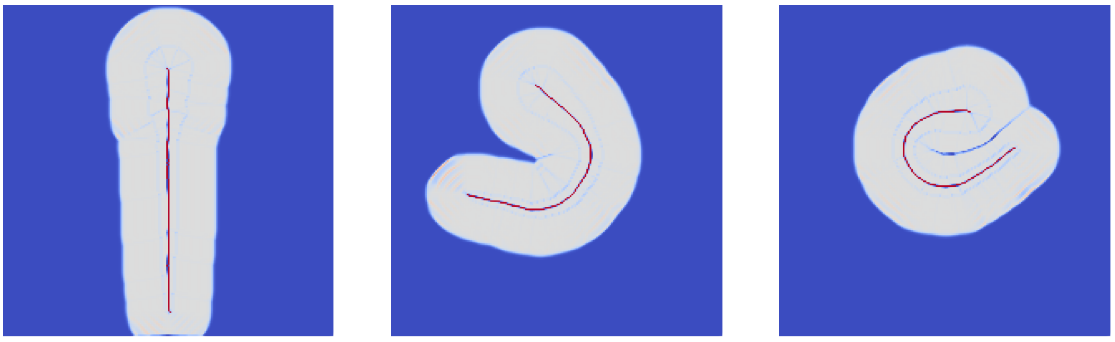
(D) $t = 30$ ms (E) $t = 40$ ms (F) $t = 50$ ms

FIGURE 3.30: **Illustration of the gradient magnitude of the level-set function.** Notice how the midline corresponds to the singular line of the gradient (low intensity pixels inside the body). By definition, the level-set gradient is 1 (red). However, we stopped the iterative process before reaching the domain boundary. That is why the gradient is not updated everywhere (blue).

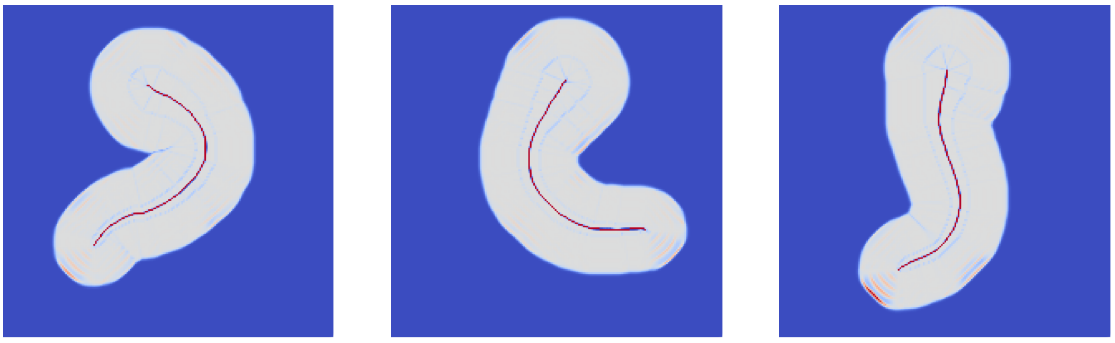
of the pixel-wise discrete midline. However, manual corrections could be needed at some points, according to the experimental data. Those corrections essentially consisted of monitoring the tracking the end-point of the discrete midline and manually correcting either the discrete midline or the discrete gradient or the thresholds, in the case of tracking issues. Figure 3.33 shows the final pixel-wise midlines after thinning, tracking, and eventual corrections.

Finally, each midline was smoothed and interpolated into N_s discrete points (Fig. 3.34) *via* Bezier curves exactly as presented previously. On the whole, the computation of the midline was not straightforward and required special attention at multiple key points and could be very time-consuming in reason of the desired accuracy considering the high-resolution data in time and space.

After recovering the N_s Lagrangian markers of each deformed midline, the last step simply consisted of putting back the corresponding transverse slices, perpendicularly to the discrete midline (Fig. 3.35). Again, we underline how all this whole reconstruction process is obviously to be considered for 2D simulations. Besides, we can emphasize that each step of this 3D methodology has been previously implemented and validated with the contour of a zebrafish surface in 2D.



(A) $t = 0$ ms (B) $t = 10$ ms (C) $t = 20$ ms



(D) $t = 30$ ms (E) $t = 40$ ms (F) $t = 50$ ms

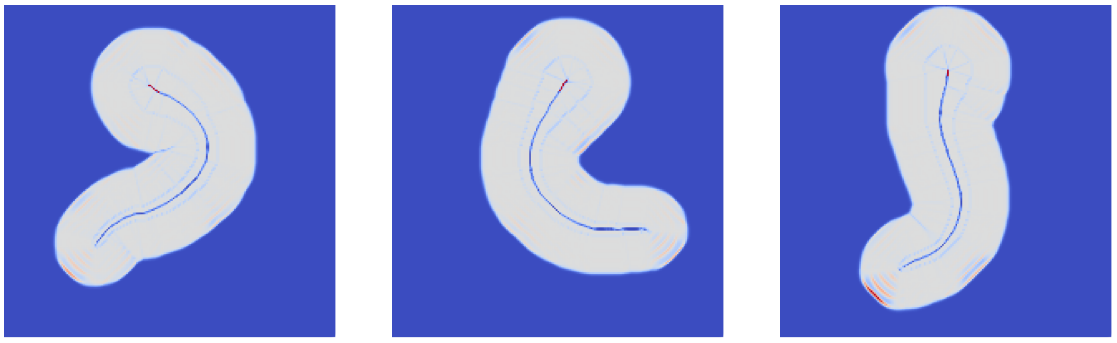
FIGURE 3.31: **Representation of the 1-pixel wide midline resulting from the custom thinning algorithm.** We can see the pixel-wise midline (red) on top of the level-set gradient magnitude (shades of blue). In addition, the zebrafish contour is also visible through the gradient magnitude.

In definitive, we reconstructed a 3D zebrafish eleuthero-embryo shape, tracked the midline kinematics from the experimental videos, and generated in 3D the body deformation motions (see Fig. 3.36). Two points are keys to construct the proper body kinematics. First, special attention was required to process the midline and effectively track the head extremity across time. It is crucial in our approach as we desire to generate a surface mesh with Lagrangian markers. Secondly, the *Procrustes Analysis* step is essential to ensure no rigid motion is present within the reconstructed body kinematics in 3D. In that regard, we perform an additional *Procrustes Analysis* step to take into account the position of the center of mass in 3D.

3.5 *Procrustes Analysis*

After the reconstruction in 3D of the body kinematics, a second step of *Procrustes Analysis* has to be performed by considering a 3D center of mass. On the one hand, the center of mass is computed regarding the 3D shape series instead of the top-view silhouettes. On the other hand, the tilt angle computation is also disturbed as we considered a barycentric calculation for determining the average tilt angle of the

(A) $t = 0$ ms (B) $t = 10$ ms (C) $t = 20$ ms



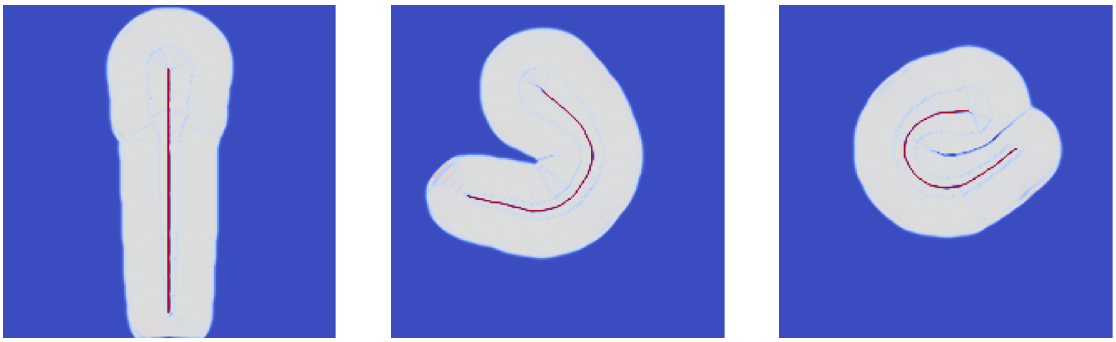
(D) $t = 30$ ms (E) $t = 40$ ms (F) $t = 50$ ms

FIGURE 3.32: **Illustration of our custom tracking procedure.** The red pixels delimit the searching area for tracking the center of the head. We represented the tracking area (red) on top of the level-set gradient magnitude (shades of blue) as the calculation was based on an adequate selection of low-gradient pixels.

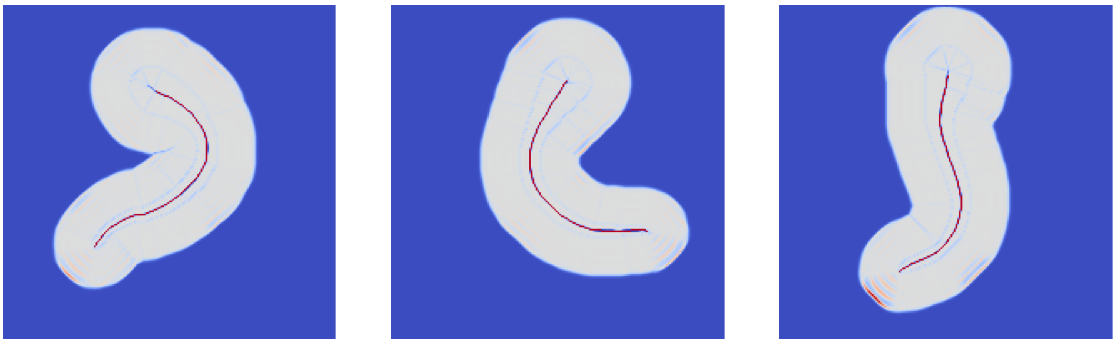
zebrafish body. In definitive, the first step detail in the previous section enabled the 3D reconstruction and a second step is required to correct the body kinematics afterward. For that purpose, we basically used the same approach as described in 2D. However, we found our methodology could lead to important inaccuracies during the *Procrustes Analysis* operation. That is why, we detail in this section, the key points to properly compute the rigid rotation of the zebrafish eleuthero-embryo, especially during the large C-bend and counter-bend motions. More highlights will be provided in the following chapter concerning the importance of a three-dimensional *Procrustes Analysis*.

3.5.1 Computing the rotating motion in 2D

Issues have been raised when analyzing attentively the angle output of the *Procrustes Analysis* performed during the image processing step. For instance, we can see in Fig. 3.6, a slight angular drop occurring between $t = 18$ ms and $t = 25$ ms. Actually, a slight angular shift as observed in Fig. 3.6, can produce very apparent and artificial motions. Indeed, if the computed rigid angle is not representative of the actual rotation then we introduce spurious rotating movements within the body deformation motions. That is why, special attention is required regarding the tilt angle of the body. In fact,



(A) $t = 0$ ms (B) $t = 10$ ms (C) $t = 20$ ms



(D) $t = 30$ ms (E) $t = 40$ ms (F) $t = 50$ ms

FIGURE 3.33: **Representation of the final pixel-wise midline computation.** The midline (red) is showed on top of the level-set gradient magnitude (shades of blue). Notice the differences near the head extremity with Fig. 3.31. Note the midline is incomplete and the extremal sections will be added when processing the pixel-wise midline.

the calculation is not straightforward as the eleuthero-embryo can bend its body into large deformations. The angle between two consecutive 3D shapes is well-defined even for C-shapes. However, we preferred to compare each 3D shape against a straight line, mainly for accuracy reasons. Indeed, the rigid angle between two consecutive shapes can be computed by searching for the optimal projection. The numerical error performed on the dichotomy for instance is therefore accumulated at each additional video frame. As we work with hundreds of video frames, the cumulative error was found rapidly too significant. Nevertheless, the angle computed relatively to a straight line is no more well-defined, especially for large deformations and C-shapes. That is why, we can observe non-natural rotation angles around the large body kinematics. To correct these phenomena accordingly, we first proposed to smooth the angle shape resulting from the *Procrustes Analysis*. We can see in Fig. 3.37 the global smoothing effects on the kinematics from Fig. 3.6.

This technique could be acceptable for a few test cases but we desire to build an automated process for experiment-based biological studies. In that regard, the smoothing parameters are not straightforward to generalize. Furthermore, notice the applied smoothing is global and not localized near the spurious effects. In terms of raw results, the spurious angle phenomenon vanished at least at first sight. Indeed, the main

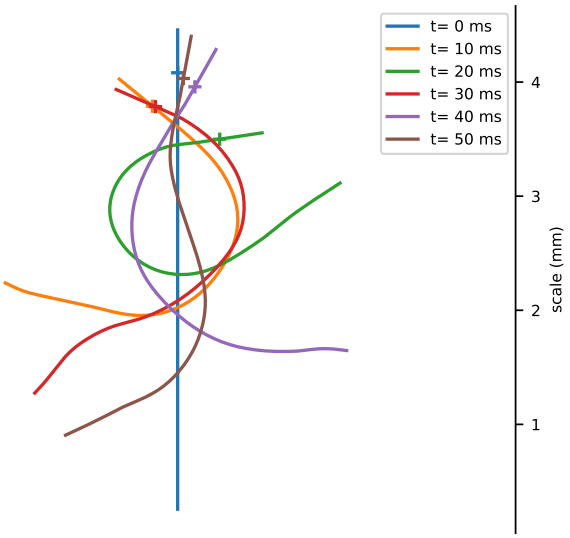


FIGURE 3.34: **Representation of six deformed midlines representing the body kinematics.** Notice only the two surface geodesics corresponding to $\theta_i(i = 1, N_\theta/2)$ are displayed. Besides, we added in the Figure, the rigid part of the midline corresponding to the head extremity. Note the six deformed shaped correspond to the experimental frames detailed previously in Fig. 3.7.

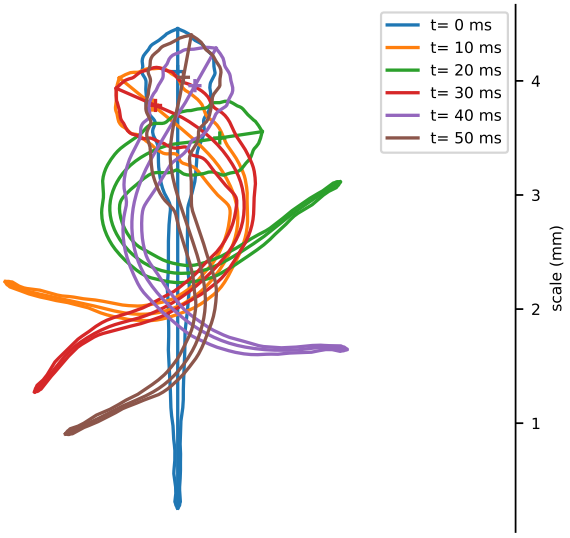
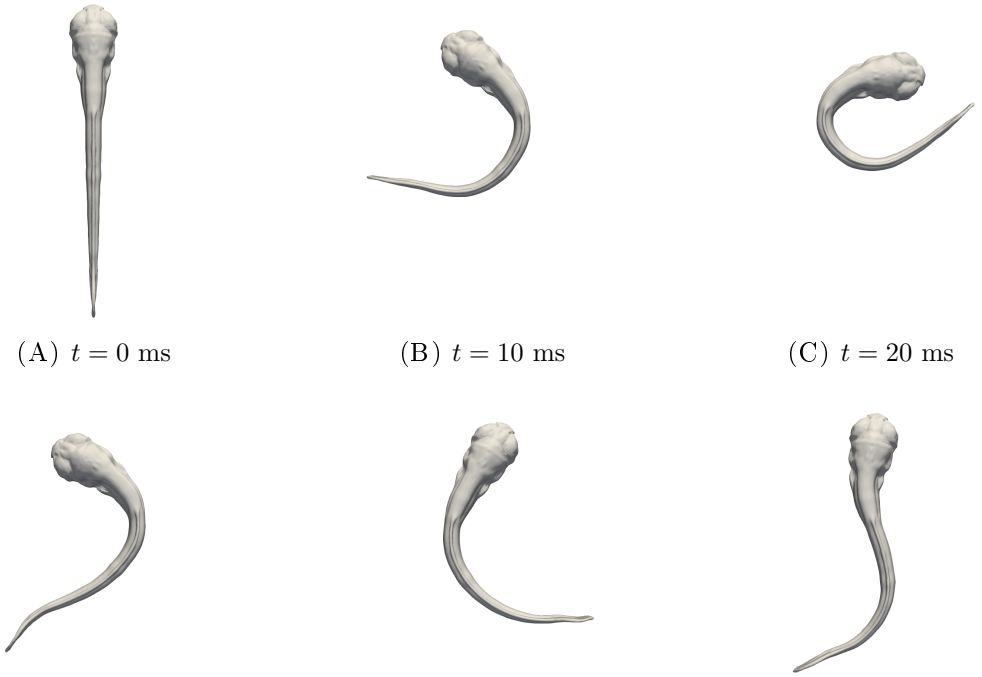


FIGURE 3.35: **Representation of six deformed midlines alongside the corresponding surface shapes.** Notice only the two surface geodesics corresponding to $\theta_i(i = 1, N_\theta/2)$ are displayed. Besides, we added in the Figure, the rigid part of the midline corresponding to the head extremity. Note the six deformed shaped correspond to the midline kinematics presented beforehand (Fig. 3.34) and to the experimental frames detailed previously in Fig. 3.7.



(A) $t = 0$ ms (B) $t = 10$ ms (C) $t = 20$ ms

(D) $t = 30$ ms (E) $t = 40$ ms (F) $t = 50$ ms

FIGURE 3.36: **Representations of the reconstructed and deformed 3D surfaces *via* Lagrangian markers.** Again, we represented the corresponding surfaces of 2D illustrations in Figs. 3.34, 3.35 and experimental images described in Figs. 3.3, 3.7.

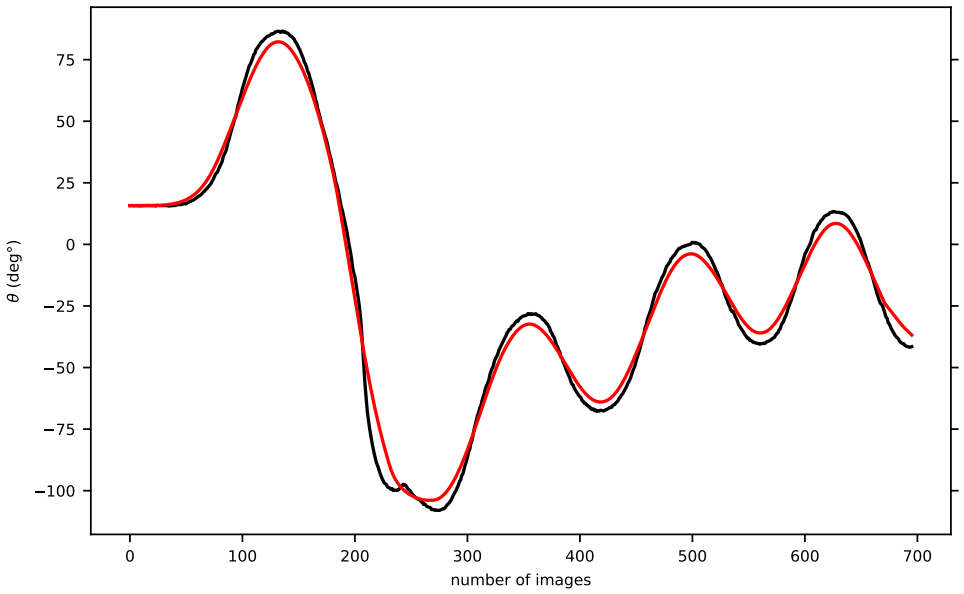
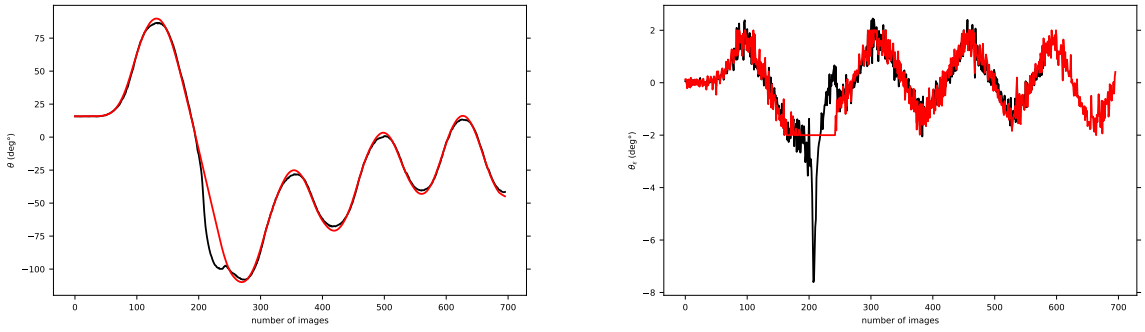


FIGURE 3.37: **Significant angle smoothing to improve the *Procrustes Analysis*.** We smoothed the original signal using a standard average filter to remove the spurious rotation occurring between $t = 18$ ms and $t = 25$ ms. The smoothing effects are well-visible on all rotation peaks, between the smoothed (red) and original (black) angular signals.

difficulty resided in the lack of criteria to predict the misbehavior of our *Procrustes Analysis* computation. Nevertheless, the rigid rotation was not subtracted directly between image n and the image at rest initially at time $t_n = 0$. As explained in the image processing section, the global subtraction was performed step by step: each image n is first reoriented according to the previous image $n-1$ before the calculation of the tilt angle. In consequences, we compute an intermediate angular step θ_ϵ between the two consecutive images n and $n-1$ which constitutes the final rotation angle θ when accumulated: $\theta_n = \sum_{1 \leq i < n+1} \theta_\epsilon^{i \to i+1}$. We reproduced in Fig. 3.38 A, the wrong approximation of the rigid angle, and we show the shape of the angular steps across video frames (Fig. 3.38 B). As we can see with the large spike, the angular step θ_ϵ parameter embedded the rotation error.

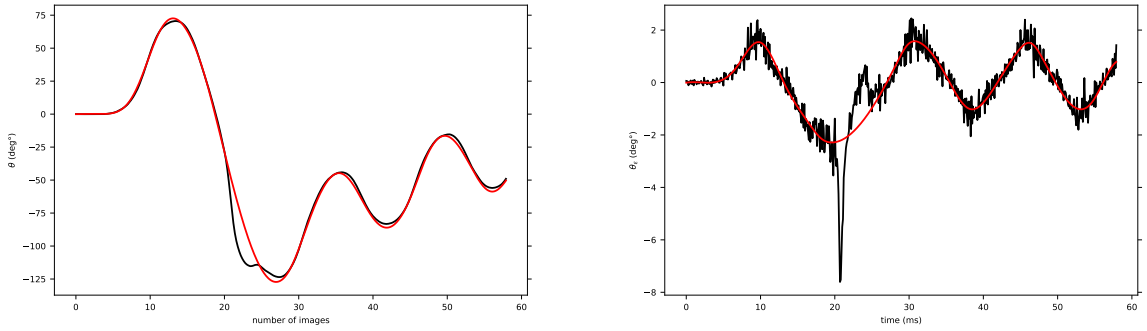


(A) Global tilt angle θ. (B) Intermediate tilt angle θ_ϵ.

FIGURE 3.38: **Enhancement of the *Procrustes Analysis* correction by using a threshold on θ_ϵ.** We can see on θ_ϵ (Fig. 3.38 B) the highly noisy signal and the limit of our angle calculation, approximately between times $t = 18$ ms and $t = 25$ ms. After applying a threshold (red) on the originally computed angles (black), we observe a better approximation of rotation peaks (Fig. 3.38 A).

Thus, we tend to correct the angular step θ_ϵ instead of the cumulative rotation θ. In Fig. 3.38, we fixed a threshold at $|\theta_\epsilon| = 2$. The resulting body rotation was acceptable. However, the approach seemed not automated and robust enough, again. Finally, we considered again smoothing methods, for the angular step θ_ϵ instead of the global output θ.

Figure 3.39 illustrates how we smoothed the θ_ϵ-signal while excluding the spurious part. The cumulative rotation θ obtained was found particularly accurate with the original rotation computation. All the *Procrustes Analysis* have been performed by integrating this θ_ϵ section. Note we significantly enhanced the *Procrustes Analysis* process although the characterization of the spurious parts still remained to some extent, arbitrary. Notice the 3D shapes are obtained based on the 2D *Procrustes Analysis*. Therefore, the potential angular steps between 3D shapes are expected to be minimal. Besides, the eventual remaining rigid angles will be added to the 2D rigid rotation.



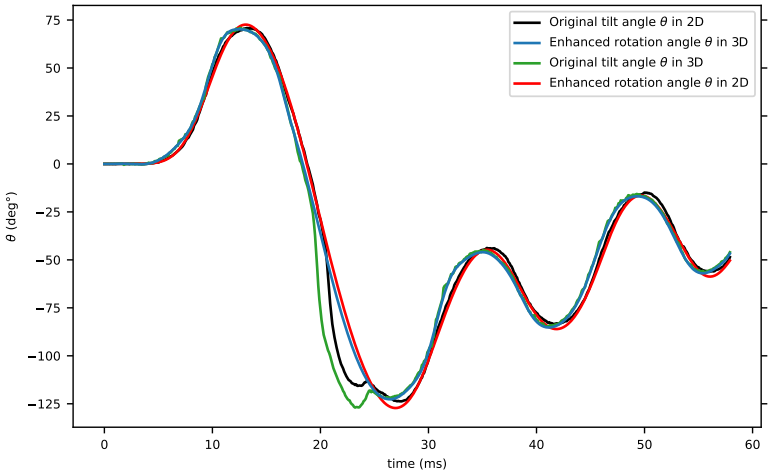
(A) Global tilt angle. (B) Intermediate tilt angle θ_ϵ.

FIGURE 3.39: **Enhancement of the *Procrustes Analysis* correction by smoothing the angular step θ_ϵ instead of θ.** After correction, the computed rotation angle (red) fits the original signal (black) and ignores the spurious calculations. Notice how slight differences on θ_ϵ (between Fig. 3.38 B and Fig. 3.39 B for example) can lead to a significant impact on the global rotation of the zebrafish eleuthero-embryo (Fig. 3.38 A against Fig. 3.39 A).

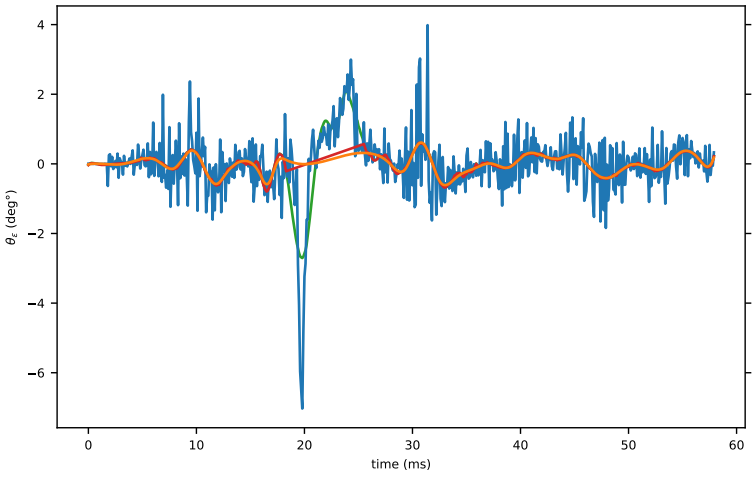
3.5.2 Computing the body rotation in 3D

When performing a similar process in 3D meaning the computation of the center of mass followed by the calculation of the intermediate tilt angle. An identical 2D-phenomenon has been raised regarding the global angle θ (Fig. 3.40 A) and the angular step θ_ϵ (Fig. 3.40 B, blue). Note we also reproduced in Fig. 3.40 A the 2D results on which the 3D outputs are computed. As for the angular step θ_ϵ, multiple filtering and smoothing methods have been required to eliminate the spurious noise. The main difficulty was due to the small variations and amplitudes of the signal, which we expected.

We focused our attention on performing an accurate representation of the experimental rigid kinematics. However, it is important to have a non-sensitive procedure as we want to obtain consistent results, regardless of the loss of precision during the *Procrustes Analysis* process. To that end, the next chapter presents the validation cases we considered to validate the entire workflow and more precisely the integration of the experimental kinematics within the CFD code.



(A) Global tilt angle θ.



(B) Intermediate tilt angle θ_ϵ.

FIGURE 3.40: ***Procrustes Analysis* in 3D: a similar enhancement.** All 2D angles (Fig. 3.40 A, black and red) are reproduced for comparison with the 3D outputs. Note in 3D, the computed angles (Fig. 3.40 A, blue and green) are added to the 2D angles, already subtracted to the motion. As for the angular step (Fig. 3.40 B), the proper approximation is not straightforward as multiple filtering and smoothing are required (green then red then orange) to filter the original spurious angular signal (blue).

Chapter 4

Validation of the global numerical approach

Previously, we presented our global numerical approach which combines a high-performance CFD code with experimental data. The CFD implementation has already been widely validated against various standard test cases in 2D and 3D [8–10, 12]. However, we strongly embedded the experimental vision within the numerical approach. As a consequence, the first important step was to validate our numerical process accordingly. Minor test cases regarding the major contributions of our work have primarily been detailed in Chapter 3. However, the validation of the global process was crucial in order to use it for applications in biology and experiments of zebrafish locomotion. In particular, we introduce the first results obtained in 2D which have primarily been based on the fully Eulerian point of view *via* optimal transportation, before using Lagrangian markers. For more realistic simulations, we present the first numerical validation case in 3D for which we based the choice of the main simulation parameters, sensitivity analyses of the global methodology such as data noise, or *Procrustes Analysis* impacts, and the analysis of the mathematical model relatively to the experiment-based approach.

4.1 2D numerical simulations

4.1.1 Fully Eulerian: optimal transportation

The first numerical results we present hereafter have been obtained by estimating the deformation velocity issued from 200×200 experimental snapshots. Afterward, deformation velocity and experimental snapshots have been used as input to the 2D CFD code (NaSCar) described in Chapter 2. Before performing considerable numerical simulations, we first adjusted specific numerical details such as image segmentation, actual fish length, or the study of convergence of the optimal transportation iterative

algorithm. Nevertheless, the resulting comparison between experimental and numerical kinematics showed slight discrepancies (Fig. 4.1), which seemed to be localized around specific time points such as C-bend and counter-bend movements which involve important rotation motions. Overall, these results suggested the global center-of-mass trajectory can be well-reproduced by numerical simulations.

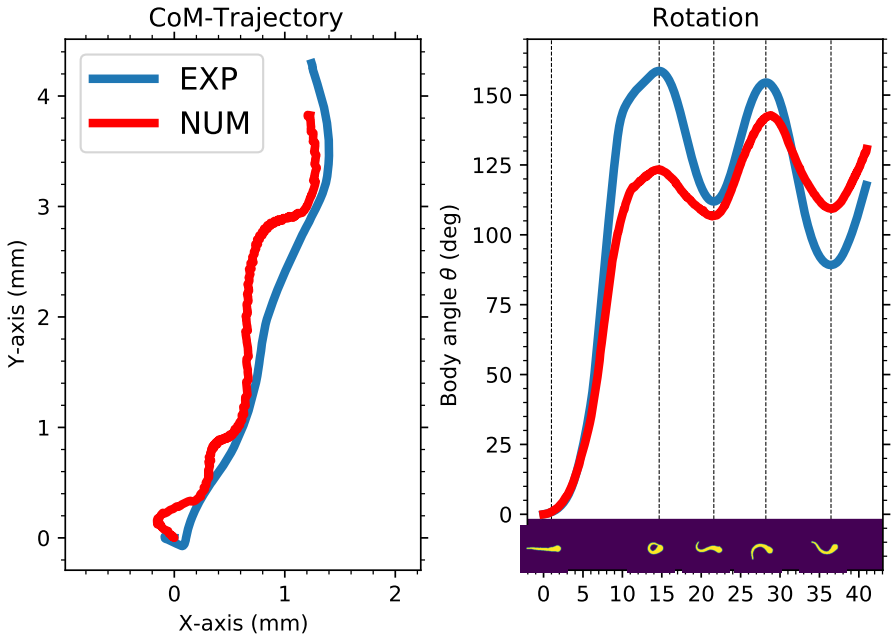


FIGURE 4.1: **Kinematic comparison between the experimental estimate (blue) and the numerical solution (red).** Note the black markers indicated along the center-of-mass trajectory (left) correspond to the maximal rotation timing denoted by vertical lines and associated zebrafish silhouettes in the rotation comparison (right). Notice the general aspect of the numerical trajectory does correspond to the initial result. Only the relative position of the CoM slightly changes. As for the rotation indicated by the body angle, we can see a larger numerical error near the C-bend (15 ms) while our solution slightly differs from the original one, especially after 30 ms. The optimal transportation algorithm has been performed on 200×200 segmented images. The computational domain was refined to 1024×1024 cells.

The direct comparison of experimental and numerical snapshots can also be seen in Fig. 4.2 hereafter. The numerical errors we observed by previously analyzing the zebrafish kinematics are clearly visible on the snapshot series, especially regarding the rotation motion. Besides, we can see the fluid vorticity which indicates the rotational movements present inside the fluid flow computed in the snapshots of the numerical solution. Eventually, the instantaneous velocity of the fluid could also be displayed as the CFD code computes the velocity of the whole computational domain at each time step.

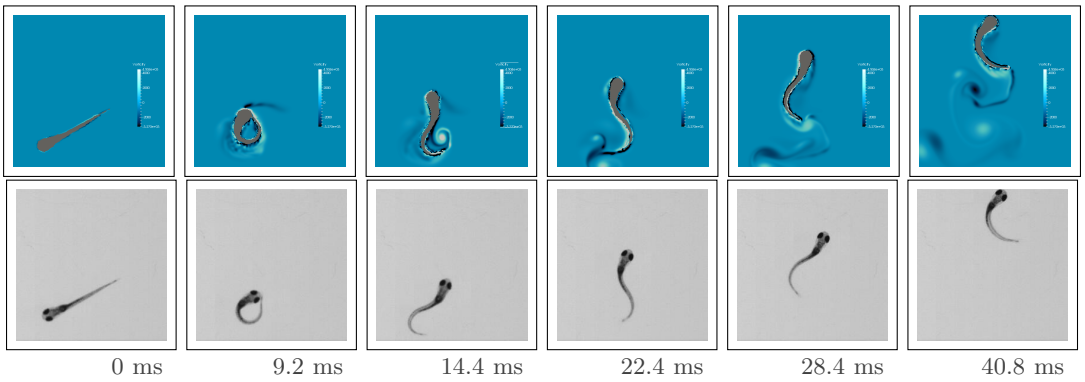


FIGURE 4.2: **Escape trajectory of a 5 dpf zebrafish eleuthero-embryo induced by an EFP stimulus.** The considered experimental escape response corresponds to Fig. 4.1. Notice how significant the numerical error produced on the body rotation between the experimental data (bottom) and the numerical solution (top) is.

In order to reduce the discrepancy between experimental and numerical center-of-mass trajectories and assess the consistency of our computational method, we performed a few numerical tests. In particular, we investigated the effects of the mesh size, boundary conditions, domain size, water viscosity, initial zebrafish orientation, and various impacts of the optimal transportation algorithm. The grid convergence is represented in Fig. 4.3 by comparing a 512×512 Cartesian mesh with a 1024×1024 computational grid. Despite slight differences in overall trajectories, we considered the numerical convergence as nearly attained with a 512×512 computational domain. However, coarser meshes significantly degraded the numerical solution as the level-set function and deformation velocity fields have to be computed with a minimum resolution. Moreover, the experimental data spatial step was fixed at approximately $\delta_i x = 2.760 \times 10^{-2}\,\text{mm}$ and thus, the resolution of the deformation velocity field was in the vicinity of the computational grid resolution *i.e.* $\delta x = 3.100 \times 10^{-2}\,\text{mm}$ for the 512×512 mesh.

Actually, we estimate the spatial resolution for which we have computed the deformation velocity field is important and should be linked to the choice of the computational grid size. When modifying the domain size, initial positioning, or optimal transportation parameters related to computational convergence, we observed slight variations of the numerical solutions while conserving the same trajectory shape (see Fig. 4.4, left).

Furthermore, we observed the choice of the initial frame could have an impact on the overall trajectory, essentially by modifying the escape direction (see Fig. 4.4, right). Usually, we chose the first frame such that no rigid movement is visible before. For instance, no significant trajectory difference was observed with additional time frames before the initial time of the red-simulation. However, non-significant silhouette variations could appear due to the segmentation noise which would generate spurious deformation velocity, especially when considering an acquisition frequency of $15,000$ fps. Therefore, our results suggest the choice of the initial frame could affect the final computed trajectory. As for the rotation, we did not observe significant

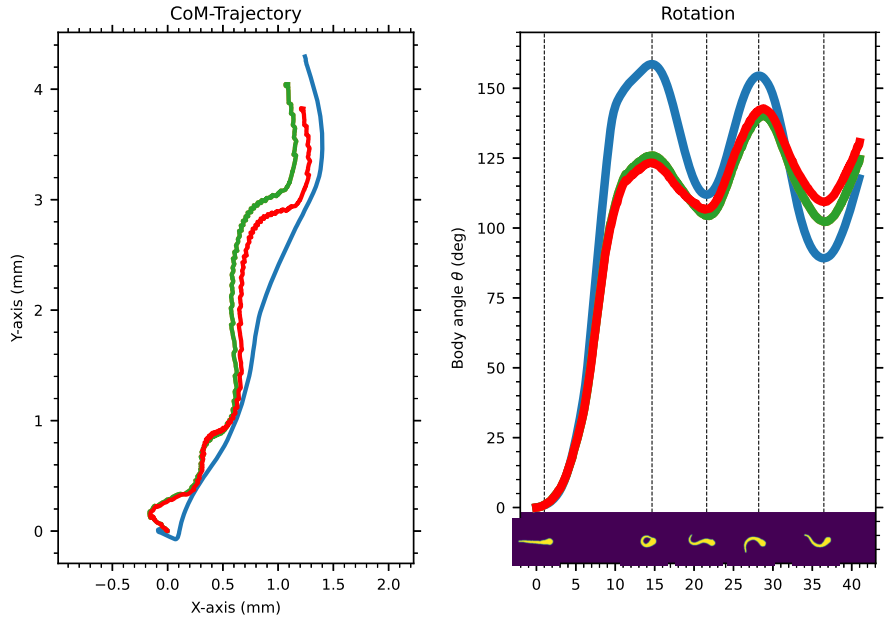


FIGURE 4.3: **Evaluation of grid convergence on escape kinematics for the initial test case presented in Figs. 4.1, 4.2.** Two different mesh size are represented: 512×512 (green) and 1024×1024 (red). The domain was fixed at $16.000\,\mathrm{mm}^2$. Note the red-kinematics corresponds to previous Fig. 4.1. The impact on the center-of-mass trajectory (left) and body rotation (right) are not important and the overall shape is identical.

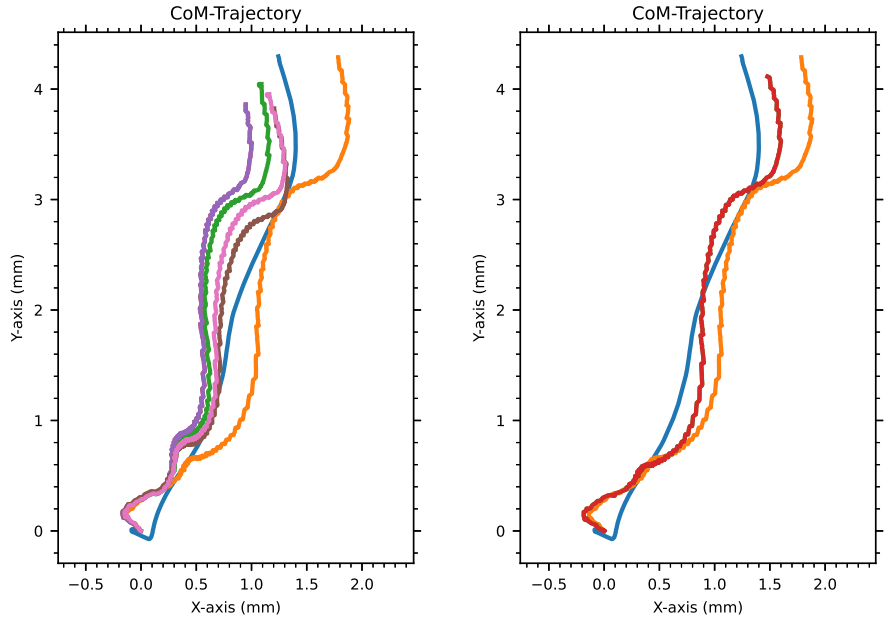
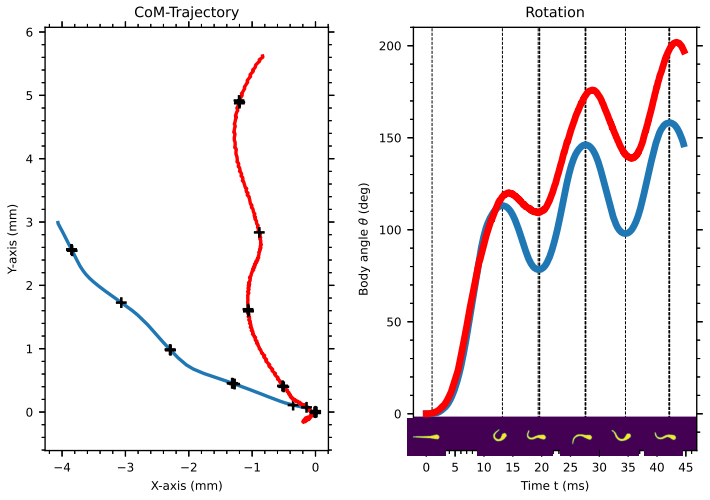


FIGURE 4.4: **Analysis of the numerical sensitivity of the center-of-mass trajectory.** Only few simulation parameters differed from each numerical simulation. On the left, we compare the previous 512×512 solution (red, Fig. 4.1) with reduced domain size (pink, brown) and different initial conditions (purple). Finally, the orange solution was computed before enhancing the numerical algorithm of optimal transportation. On the right, we only changed the initial time frame by integrating ten additional frames (red trajectory).

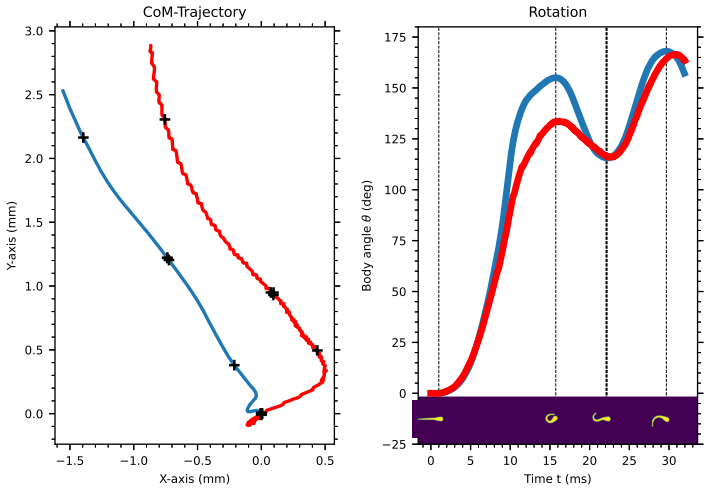
differences between each numerical simulation. In order to assess the variability of numerical errors without further analysis of data sensitivity, we conducted automated numerical simulations on additional test cases. Indeed, the objective is to develop a high-performance, robust, and consistent numerical tool which can be automated on large experimental samples with only simple experiment-based or segmentation-wise parameters to tune. To that end, three supplementary escape simulations shown in Fig. 4.5, have been performed with identical acquisition frame rate, optimal transportation algorithm, domain and mesh sizes as the 512×512 green-simulation presented in Figs. 4.3, 4.4.

As a result, we can see the numerical simulation was highly variable according to the experimental locomotion. In particular, the case in Fig. 4.5 A presents the proper swimming gait excepted the escape swimming direction does not correspond to the actual observations. In addition, the eleuthero-embryo seems to slightly accelerate, at the beginning of the fast-swimming module. In the second escape response (Fig. 4.5 B), the fast-swimming gait does correspond to the experimental speed, rotation, and escape direction. However, the C-bend is not well-reproduced as a large error can be seen in the body angle. That is why, the numerical trajectory is shifted during the C-start although the following escape modules are well-approximated. As for the third escape simulation, large errors are made at all stages of the C-start. Such result strongly suggested that more investigations are required into all the different modeling steps: image processing, *Procrustes Analysis*, optimal transportation, CFD choices and simulation parameters, in order to understand, limit and enhance the accumulation of data noise and numerical errors which can lead in some cases to non-significant numerical simulations.

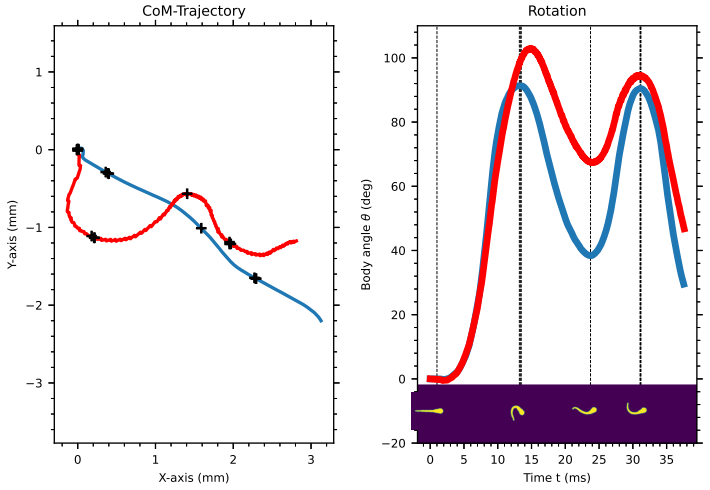
On the whole, optimal-transportation-derived numerical simulations raised several questions. On the one hand, the overall attained precision was not satisfying enough. For instance in the first initial escape response, the general trajectory or rotation shapes were found consistently similar despite multiple simulation modifications. That is why, we concluded that the numerical modeling required important enhancements. For example, the dimension of the problem should probably have an important role in the observed differences. Indeed, zebrafish eleuthero-embryos do swim in three dimensions although the general motion stays horizontal and planar. For 2D simulations, we considered both fluid flows and zebrafish morphology in 2D, which in consequence impacts significantly the resulting computed zebrafish locomotion. On the other hand, we observed negligible to large numerical errors according to the considered test case which implied our numerical approach is highly sensitive and dependent on the experimental video and its associated body deformations. Again, further investigations would be required to prospect the origin of numerical errors. As for optimal transportation results, we highlight the convergence rate and precision were also found highly dependent on the two considered experimental frames. For instance, two consecutive optimal transportation calculations could provide different numerical behaviors according to the experimental



(A)



(B)



(C)

FIGURE 4.5: **Kinematic comparisons between the experimental estimate (blue) and the numerical solution (red).** The three figures Figs. 4.5 A, 4.5 B, 4.5 C denote three additional and distinct experimental responses. Note the black markers indicated along the center-of-mass trajectory (left) correspond to the maximal rotation timing denoted by vertical lines and associated zebrafish silhouettes in the rotation comparison (right).

data. However, differences of convergence in optimal transportation calculation should be negligible provided a minimal amount of iterations are performed. Nevertheless, our results suggest a trade-off has to be reached between the numerical precision such as optimal transportation accuracy and convergence, and the data noise naturally generated during the segmentation (in space) and image processing algorithms (in time). Indeed, we work with high-resolution data in space and time which could produce spurious movements and deformations and thus significantly degrade the computed solution. Note the accumulation of data noise and the associated accumulation of optimal transportation numerical errors could also be considered as an explanation to computational misbehaviors or bad numerical CFD solutions relatively to the expected experimental trajectory. Finally, the core method we used to extract the deformation velocity field from the experimental images might not be well-suited for our application, although optimal transportation is designed *in theory* to minimize the energy required for swimming and more precisely, deforming the body kinematics. In fact, general optimal transportation algorithms are well-known to not capture all motions and physics. In particular, neither rotation motions nor tangential velocity cannot be well-estimated by optimal transportation due to the intrinsic translation property: the optimal path corresponds to straight lines (see Chapter 3). Therefore, minor errors might be introduced in the computed deformation velocity at certain locations on the silhouette, according to the actual body kinematics. Specific generalizations of a few optimal transportation algorithms have been developed to tackle those issues (see Appendix A).

In definitive, the optimal transportation computation was based on a validated and efficient Eulerian algorithm, with spatial-and-temporal high-resolution experimental images, after the segmentation and image-processing step. Numerical simulations have been performed across four experimental escape responses, with various segmentation, transportation, or simulation parameters. On the whole, we obtained satisfying results in some specific cases and large numerical errors in other cases. In particular, certain center-of-mass trajectories have been entirely or partly well-reproduced while the body rotation was poorly computed, especially during C-bend motions. In addition, the numerical errors have been found highly-variable across different experimental test cases which limits the reproducibility of an automated process. We see four limitations in the use of optimal transportation in such real-world applications. First, a fast and robust computational method is required to handle large datasets. The robustness and precision of the Eulerian algorithm we used might be improved. However, this is a minor consideration given a large amount of data in time. Secondly, the output deformation velocity field is strongly correlated to the image segmentation, image *Procrustes Analysis*, and optimal transportation accuracy. If the first two steps are not well-approximated, optimal transportation cannot compute the expected velocity field. Above all, a numerical noise is present in the experimental and segmented data which could transfer into spurious oscillations in the computed velocity field. An adequate

trade-off might be needed between the data resolution in time and space, and the optimal transportation overall precision. Third, the Eulerian algorithm we considered computes the original Monge optimal solution without imposing specific and additional physical constraints such as incompressibility or rigidity (see Appendix A). Therefore, spurious mass displacements might occur during optimal transportation, which do not represent faithfully the actual deformations. We estimate further investigations would be needed to understand and enhance our optimal-transportation approach. Finally, we compared 2D simulations with 3D animals and experiments. Thus, 3D numerical simulations would be an important step towards real-like simulations and valid comparisons against experiments.

4.1.2 Fully Eulerian vs Lagrangian markers

In the previous Chapters 2 and 3, we introduced two main approaches to embed the experimental kinematics into the CFD solver which use either optimal transportation and a deformation velocity field, or Lagrangian markers and midline deformations. The representation of the zebrafish body differs drastically between the two methods: either fully Eulerian *via* a level-set approach, or Eulerian-Lagrangian by directly tracking the Lagrangian markers. Although fully Eulerian results have previously been presented, additional simulations have been performed in 2D to compare the Eulerian and Lagrangian approaches with an identical computational mesh (Fig. 4.6). Note this present escape response was recorded with the Mini WX100 camera and at a different frame rate as previously: $10,000$ fps instead of $15,000$ fps. Basically, we considered those settings to obtain a higher video quality and above all, to track in a simpler way the zebrafish eleuthero-embryo such that a minimum amount of tail-beats were recorded. The present test case will be described in more details in the next section.

Both methodology have been widely described in Chapter 3. On the one hand, the optimal-transportation-based approach might be less suitable than previously due to the frame rate decrease. Indeed, the considered optimal transportation algorithm was particularly designed toward very close frames. Nevertheless, we implemented the correction algorithm within the *Procrustes Analysis* step in the present test case, unlike in the previous escape locomotions. Afterward, the imaging data were slightly regularized for the computation of optimal transportation. On the other hand, the 2D process used for generating the Lagrangian markers required for the Eulerian-Lagrangian approach, and reconstructing the body deformations of the contour, is identical to the presented 3D implementation (Chapter 3). Note Lagrangian markers have been found to be more difficult to generate in case of ultra-high or lower acquisition frequency, especially for tracking the midline with a minimum amount of data noise. Indeed, we highlighted the major bottleneck of this approach may reside in the tracking algorithm which would require enhancements to be robust across a wide range of space and time resolutions.

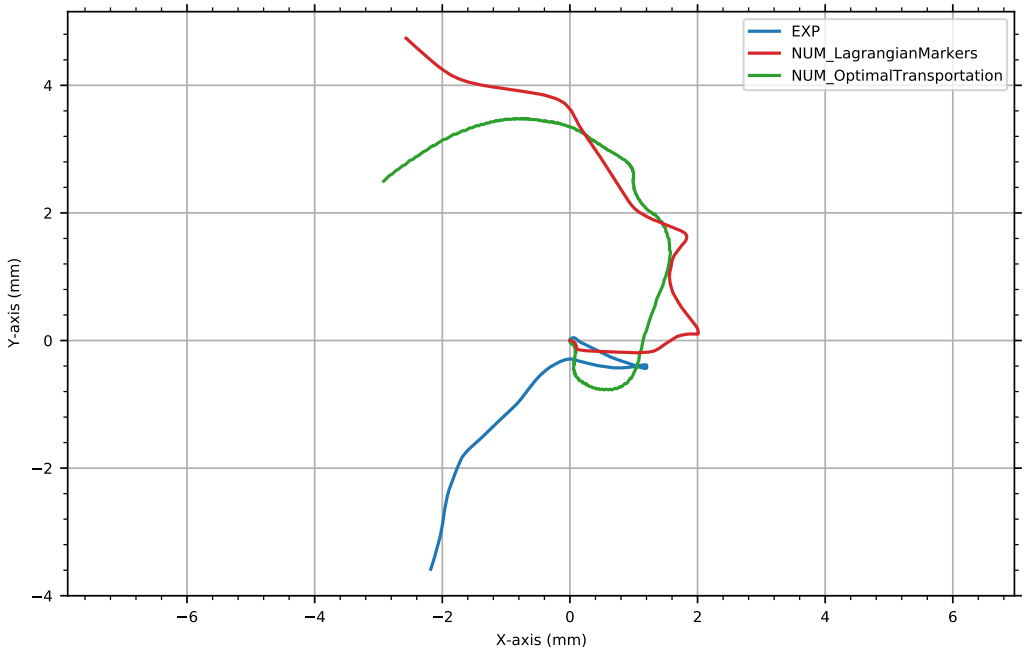


FIGURE 4.6: **Comparison between two approaches based on optimal transportation and Lagrangian markers, by considering the same test case.** The fully Eulerian simulation based on optimal transportation (green) was performed with a 512^2-grid as with previous results (Chap. 3). As for the Eulerian-Lagrangian method (red), we used 300 Lagrangian markers to describe the whole contour, which we combined with a 512^2 Cartesian grid. Both numerical simulations have been compared against the experimental kinematics (blue).

Besides, important spatial and time filtering were required in 2D to smooth the input data *i.e.* contours of Lagrangian markers. Otherwise, spurious effects were observed in the level-set computation from Lagrangian markers.

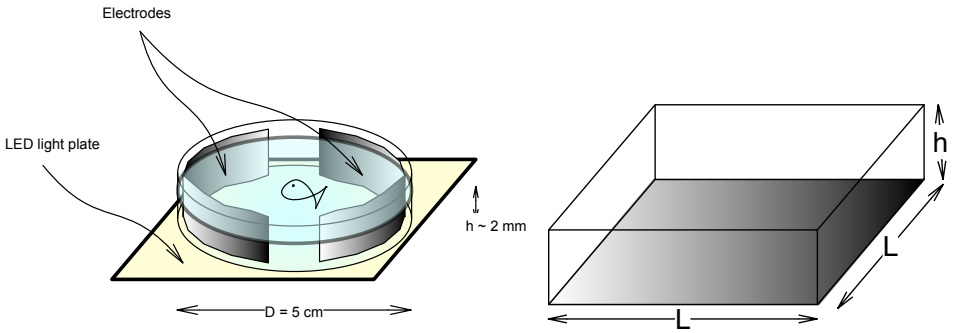
As a result, both numerical simulations presented the same kinematic trend and could not reproduce the global experimental kinematics. Nevertheless, the Eulerian and Lagrangian simulations presented a similar trajectory, based on identical simulation parameters. The input data processing constituted the main differences in the CFD code. In fact, the construction of the input data might be prevalent to compare the two approaches. First, the optimal transportation algorithm can handle very strong C-bend motions where the tail would overlap with the head of the zebrafish eleuthero-embryo from the top-view at least, while the level-set function cannot be reconstructed from the Lagrangian markers in that case. Secondly, the whole deformation velocity field is computed through the optimal transportation algorithm in the first approach and thus cannot be easily validated due to the complex motions involved. Instead, the boundary velocity is simply recovered from the reconstruction of Lagrangian markers, which is entirely monitored step-by-step in the latter approach. Therefore, it seems optimal transportation algorithms directly compute the deformation velocity field at the expense of implicit and potentially spurious mass transportation which is not straightforward to validate. In contrast, the Eulerian-Lagrangian approach can easily

be enhanced but requires more data processing and above all, is based on an important modeling assumption: the Euler-Bernoulli beam deformation theory which states each cross-section remains orthogonal to the midline throughout the entire escape motion. Actually, complex escape responses might also involve different body deformations although the Euler-Bernoulli hypothesis seems reasonable. Eventually, it is important to note the presented numerical simulations may not be fully converged. However, the large discrepancy observed between numerical simulations and experimental kinematics strongly suggests 3D modeling is required to perform more accurate and realistic numerical simulations. That is why, 3D numerical simulations are considered and validated in what follows, based on the construction and tracking of Lagrangian markers. That is why, 3D numerical simulations are considered and validated in what follows, based on the construction and tracking of Lagrangian markers.

4.2 A first experimental 3D test case

The first 3D numerical simulations were performed on the initial test case presented in the previous chapter (Figs. 3.34, 3.35, 3.3, 3.7, 3.36).

4.2.1 Preamble



(A) Experimental cell. (B) Computational domain.

FIGURE 4.7: **Illustration of the experimental cell and the computational 3D domain.**

Fluid dynamic viscosity and density were chosen to $\mu = 0.83$ mPa.s and $\rho = 1000$ kg/m^3. As for the zebrafish body, we assumed the body mass density is homogeneous and constant, equal to the fluid density. As discussed in [131], mass variations are naturally present in the zebrafish larval body especially for the MFF and the yolk sac. However, we can reasonably assume no significant variations of mass density are present. As introduced before, the 3D computational domain is constituted of a rectangular Cartesian mesh and represents a sub-domain of the experimental cell (see Fig. 4.7). Naturally, the zebrafish eleuthero-embryo is free to explore the circular cell. However, we selected the

experimental videos in which the eleuthero-embryo stays in the focal field of the camera. Thus, we chose the physical dimensions to be of the same magnitude order ($L \approx 12$ mm), except we initially centered the zebrafish eleuthero-embryo's reconstruction to limit boundary effects. As for the boundary conditions, we imposed free conditions appropriately on tangential velocities $u_{\parallel}$ and orthogonal velocity $u_{\perp}$ as:

$$u_{\perp} = 0, \frac{\partial u_{\parallel}}{\partial x_{\parallel}} = 0 \tag{4.1}$$

Moreover, we fixed the height of the domain at 2 mm which approximately corresponds to the experimental water depth: the actual water height being higher due to the presence of electrodes. As for the bottom boundary, we imposed a wall condition with a homogeneous Dirichlet condition: $\mathbf{u}_{bot} = 0$. The mesh size was chosen such that: $\delta x = 1.429 \times 10^{-5}$ mm. Eventually, the domain size was increased while keeping the same spatial step, particularly when the domain was found too small. Indeed, at the end of the fast-swimming stage, the eleuthero-embryo might reach the domain boundaries. That is why, boundary conditions cannot interfere with interior velocity and pressure. No inlet or outlet conditions are then imposed. Initially, we assumed no fluid or body motion is present and the zebrafish eleuthero-embryo is well-balanced such that no 3D motion is modeled.

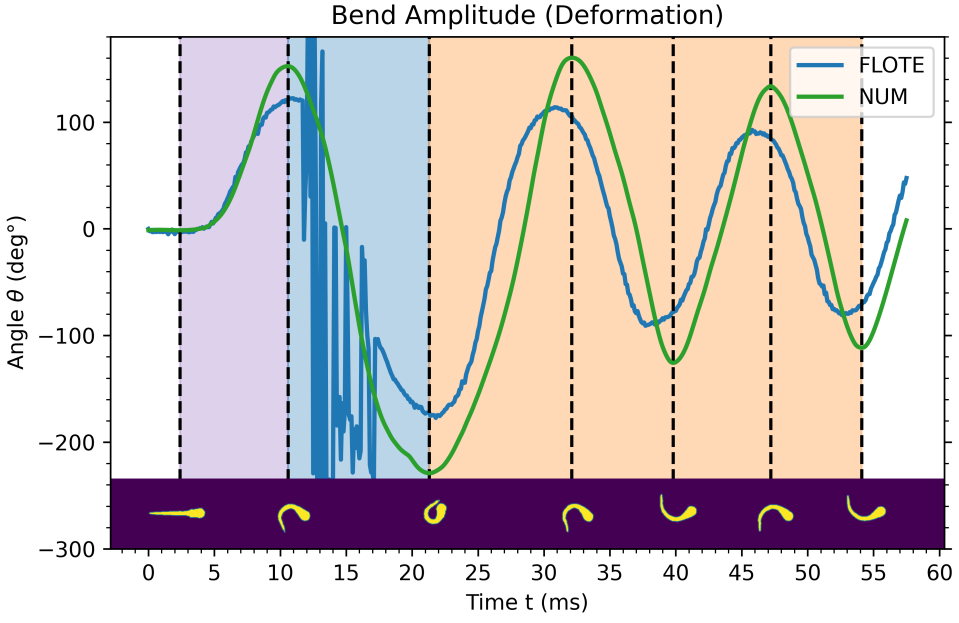
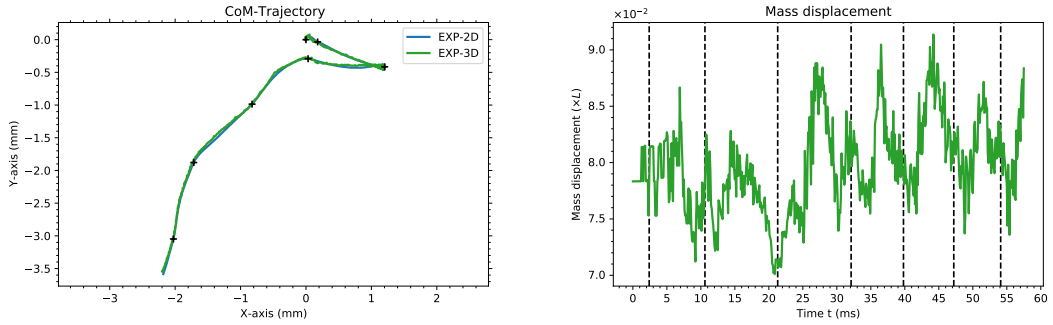


FIGURE 4.8: **Body deformations of the initial test case presented in Chapter 3.** The new calculation of the bend amplitude (green) is more accurate than the current software *FLOTE* bend amplitude (blue). The bottom pictures illustrate the body kinematics corresponding to the extrema of the bend amplitude. Besides, the three colored stages define the C-bend, counter-bend and fast-swimming phases, respectively.

Before conducting the numerical simulation, we computed the global bend amplitude from the midline discretization by using the original definition (Fig. 1.16). The resulting bend amplitude represents body deformations. As we used more than 300 midline segments, our calculation provided a better approximation of body kinematics than the actual tracking software namely *FLOTE* (see Fig. 4.8). In the first considered test case, we can notice both the C-bend and counter-bend motions are strongly deformed (see Fig. 4.8).

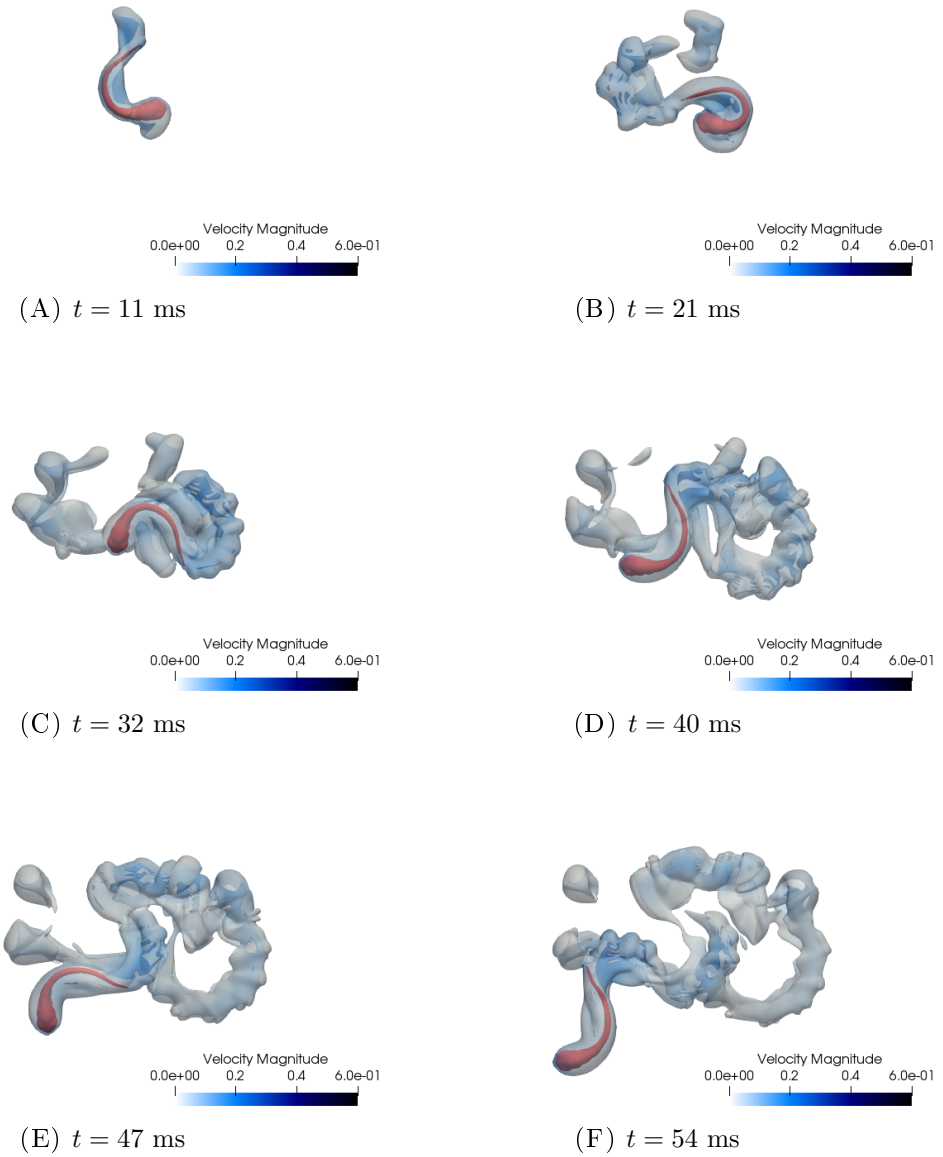


(A) (B)

FIGURE 4.9: **Impacts of the 2D and 3D calculation of the center of mass.** Only minor differences can be seen between the 2D and 3D trajectories (Fig. 4.9 A) although the overall displacement can be considerable according to the considered body deformation (Fig. 4.9 B). Note vertical lines denote the timing of deformation extrema (Fig. 4.8). The corresponding occurrence are also represented by markers '+' on the trajectory (Fig. 4.9 A).

All numerical simulations have essentially been compared with the experimental trajectory as it constitutes in general, a good and reliable indicator of the overall escape kinematics. Indeed, we preferred to emphasize the center-of-mass trajectory comparison as the rotation calculation is considerably based on the *Procrustes Analysis* which introduces slight numerical errors. Instead, the experimental trajectory was directly obtained from the segmentation and 2D *Procrustes Analysis* processes. Note differences between 2D and 3D center-of-mass calculations have only a minor impact on the resulting trajectory (see Fig. 4.9). In what follows, we will highlight the 3D computation is particularly prevalent to the *Procrustes Analysis* and the computation of body rotation as the center of mass varies considerably for high deformations and high rotation motions (Fig. 4.9 B). Kinematic quantities such as trajectory and velocities are very useful to validate the numerical simulation against the experimental data as swimming kinematics is generally accessible *via* standard tracking software. Notice we reconstructed in 3D the whole escape swimming motion by deforming the midline kinematics. In consequence, other measurements could be conducted with our model, such as the snout or tail tracking to provide the associated velocity and thus, information about tail propulsion. Given the deformation velocity and simple modeling of forces and torques, inverse-kinematics methods have recovered the overall swimming power [124]. Instead, the solution computed by the numerical simulation is complete, very accurate, minimizes the number of assumptions, and provides more information about the fluid flow as we

describe not only the body dynamics but the whole fluid dynamics and its interactions. For instance, for the current numerical simulation, we represented the fluid flow (Fig. 4.10) in addition to the eleuthero-embryo's displacements for the deformation extrema denoted by the vertical lines in Fig. 4.8. An example of a 3D wake is also provided (Fig. 4.11) to give supplementary insights.



(A) $t = 11$ ms (B) $t = 21$ ms

(C) $t = 32$ ms (D) $t = 40$ ms

(E) $t = 47$ ms (F) $t = 54$ ms

FIGURE 4.10: **Overview of the numerical solution corresponding to the original video in Fig. 3.3 and deformation pictures in Fig. 4.8.** The zebrafish escape motion is illustrated by six time snapshots through Figs. 4.10 A-4.10 F. We represented in red the zebrafish eleuthero-embryo while the fluid flow is described by the so-called fluid vorticity which highlights the rotation of the fluid rotation. We colored an iso-contour of the vorticity with the corresponding fluid velocity magnitude.

Therefore, we can observe through the numerical simulation all the fluid movements occurring during the escape response of a zebrafish eleuthero-embryo. In particular, we

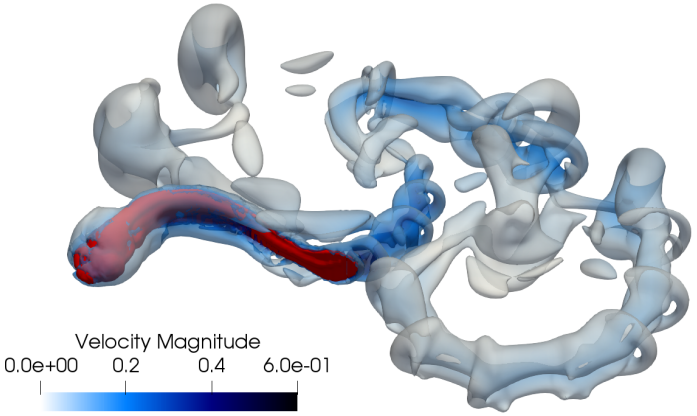


FIGURE 4.11: **Illustration of a 3D wake generated by a zebrafish escape response.** The test case corresponds to the original video (Figs. 3.3, 4.8, 4.10) at $t = 47$ ms. We represented in red the zebrafish eleuthero-embryo while the fluid flow is described by the Q-criterion, which highlights the fluid rotation of the 3D vortices similarly to vorticity (see Chapter 2). We colored an iso-contour of the Q-criterion with the corresponding fluid velocity magnitude.

can see how the eleuthero-embryo pushes away a whole arc of fluid for each tail-beat. Afterward, the circular fluid jet is forming rings which eventually dissipates later on. Each vorticity ring effectively describes a pair of fluid vortices with two opposite rotation velocity. Notice how important the arc of fluid produced by the counter-bend motion is (Fig. 4.10 E) compared to the C-bend beat (Fig. 4.10 B). In the last illustration (Fig. 4.10 F), we can see two rings on the left from the tail-beat occurring after the C-bend, the jet of fluid due to the tail-beat occurring after the counter-bend, and the premises of a double row of vortices generated during the fast-swimming tail-beats.

Furthermore, the numerical code computes the forces and torques applied on the body surface, resulting from the interaction with the fluid flow and the body dynamics. In consequence, global power consumption can be visualized across all time steps (see Fig. 4.12). The power consumption corresponds to the power expended at each time step by the zebrafish body to counter the fluid forces and move its body accordingly. Therefore, we only calculated the mechanical power of the zebrafish body and we assumed the mechanical power represents the biological power produced by muscles. Note that not all muscle power is not necessary for leisure locomotion but in the case of an escape response *via* an EFP, all the available power is expected to be used by muscles to maximize the escape motion.

Given the noisy numerical signal, we post-processed the power output by averaging the power consumption over a time window of one millisecond. Afterward, we smoothed the averaged power to obtain a smooth and accurate estimation of the expended power across time. The smoothing step is particularly important as the actual power consumption is continuous while the numerical power output is largely discontinuous due to time discretization for computing the numerical solution. As a result, we can see the power shape is also formed by multiple peaks occurring during each tail-beat. Indeed, the

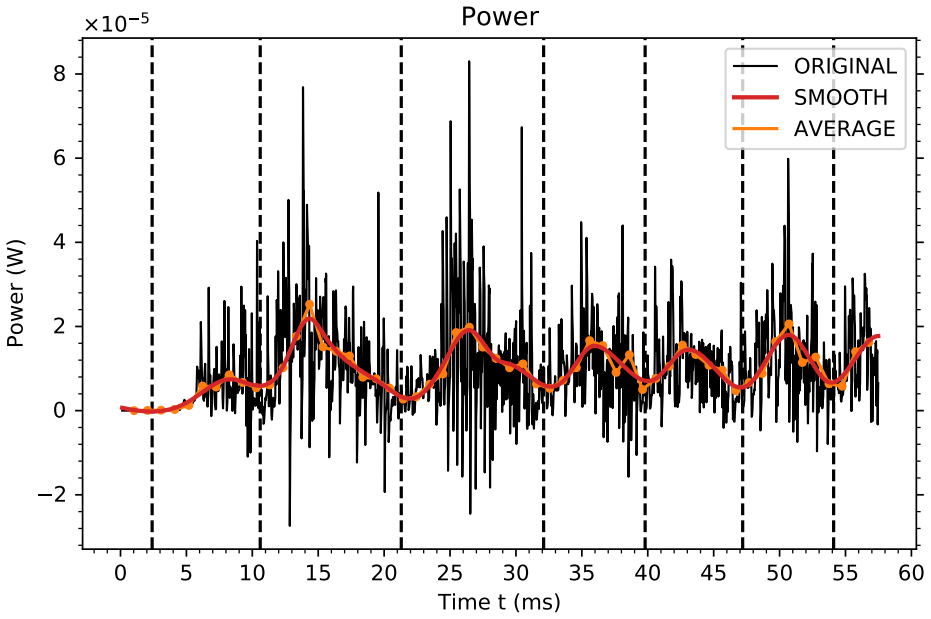


FIGURE 4.12: **Power consumption of a zebrafish eleuthero-embryo for performing a fast-start and two cycles of fast-swimming.** The numerical power output (black) is very noisy due to the time discretization of the solver. Thus, the power signal is averaged every millisecond (orange) before applying a smoothing filter (red). Note vertical lines denote the timing of deformation extrema (Fig. 4.8).

minima of the power are well-synchronized with deformation kinematics. Regarding the maxima, they do not seem correlated with the bend amplitude. Indeed, the power reaches its maximum in the first tail-beat *i.e.* when deforming into a C-shape from rest while the power required for deforming the eleuthero-embryo's body during the counter-bend motion, is less important.

From the power output, we computed four quantities: the maximal power (P_{max}) (in mW), the mean power (P_{mean}) (in mW), the total expended energy (E_t) (in mJ) and the CoT (in J/m) of the zebrafish escape swimming (Fig. 4.13). P_{max} was not considered as important as the other quantities as it is not correlated to a specific part of the escape motion. However, P_{mean}, E_t and CoT were found very interesting to study. Notice P_{mean} is based on the averaged and discontinuous power output but we estimate all mean powers are identical, in the limit of an accurate enough smoothing step. Indeed, in the case of escape response movements and C-starts, no standard efficiency such as the Froude efficiency could be defined by extracting the useful work from the total work, unlike steady and periodic swimming. As a result, we defined an additional performance quantity called CoT, as the ratio of total energy over the total traveled distance. In the literature, the mass of the fish is generally also taken into account. However, we used 5 dpf zebrafish eleuthero-embryos, which are considered of the same mass as the total length is approximately of the same magnitude order at this age. Therefore, the CoT is decreased either by a smaller amount of energy or a larger traveled distance.

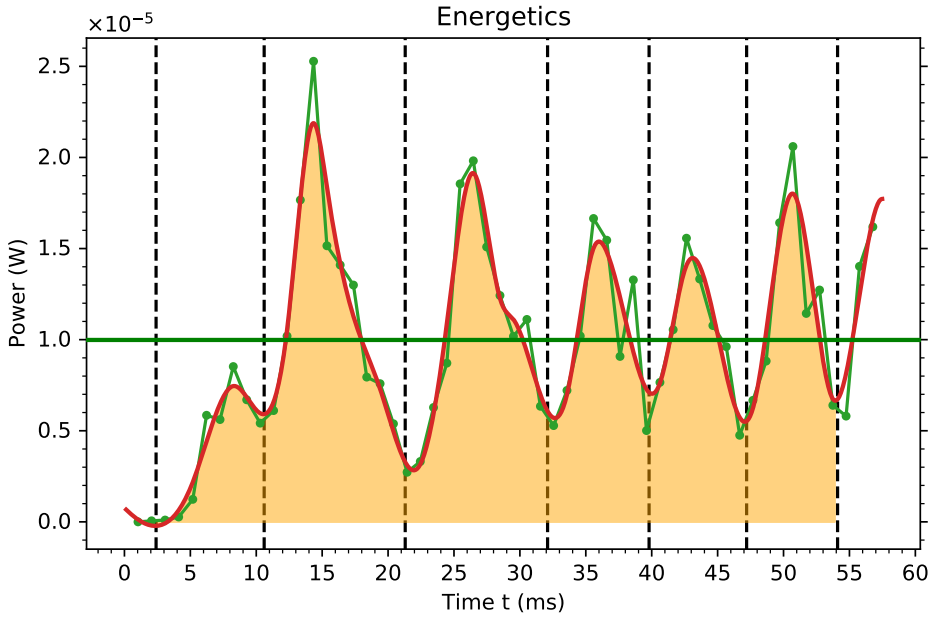


FIGURE 4.13: **$\mathbf{P_{mean}}$ and $\mathbf{E_t}$ from the power output.** The P_{mean} value (horizontal green line) is computed by averaging the smoothed power. E_t is computed by integrating the smoothed power over time. As a result, the expended energy corresponds to the area (orange area) delimited by the power curve.

TABLE 4.1: Energetic performances of the test case presented in Fig. 4.13.

P_{max} (mW)	P_{mean} (mW)	E_t (mJ)	CoT (J/m)
2.181×10^{-2}	0.998×10^{-2}	5.118×10^{-4}	7.140×10^{-5}

It is an important quantity as the ratio compares the energetics with the kinematic quantity which the eleuthero-embryo *a priori* maximizes during the escape response. As an example, the energetic performances of the test case depicted in Fig. 4.13 are presented in Table 4.1.

Before performing numerical simulations, we needed to fix the global simulation process which includes the choice of simulation parameters, computational domain, mesh size, *Procrustes Analysis* and penalization method, by using the current test case.

4.2.2 Simulation methodology

4.2.2.1 Grid convergence

Beforehand, we evaluated the optimal computational mesh size. Numerical schemes are constructed such that the computed and approximated numerical solution converges towards the exact solution in the limit of $\delta x \to 0$. However, a trade-off needs to be found between numerical accuracy and the mesh size which impacts the computational time. That is why, we compared in Fig. 4.14, three mesh size: $\Delta x = 3.810 \times 10^{-2}$ mm,

TABLE 4.2: Evaluation of the impact of grid convergence in energetics.

N_x	P_{max} (mW)	$\Delta^r_{P_{max}}$	P_{mean} (mW)	$\Delta^r_{P_{mean}}$
420	2.174×10^{-2}	-	1.016×10^{-2}	-
315	2.008×10^{-2}	7.617×10^{-2}	0.924×10^{-2}	9.012×10^{-2}
525	2.308×10^{-2}	6.182×10^{-2}	1.077×10^{-2}	6.041×10^{-2}
N_x	E_t (mJ)	$\Delta^r_{E_t}$	CoT (J/m)	Δ^r_{CoT}
420	5.206×10^{-4}	-	7.302×10^{-5}	-
315	4.714×10^{-4}	9.463×10^{-2}	6.910×10^{-5}	5.361×10^{-2}
525	5.514×10^{-4}	5.910×10^{-2}	7.562×10^{-5}	3.566×10^{-2}

For each quantity q, the relative numerical error Δ^r_q denotes the relative difference with the chosen size $N_x = 420$.

$\Delta x = 2.857 \times 10^{-2}$ mm, $\Delta x = 2.286 \times 10^{-2}$ mm which represents approximately a regular increase of 30% in mesh size: the respective mesh size being $N_x = 315$, $N_x = 420$, $N_x = 525$.

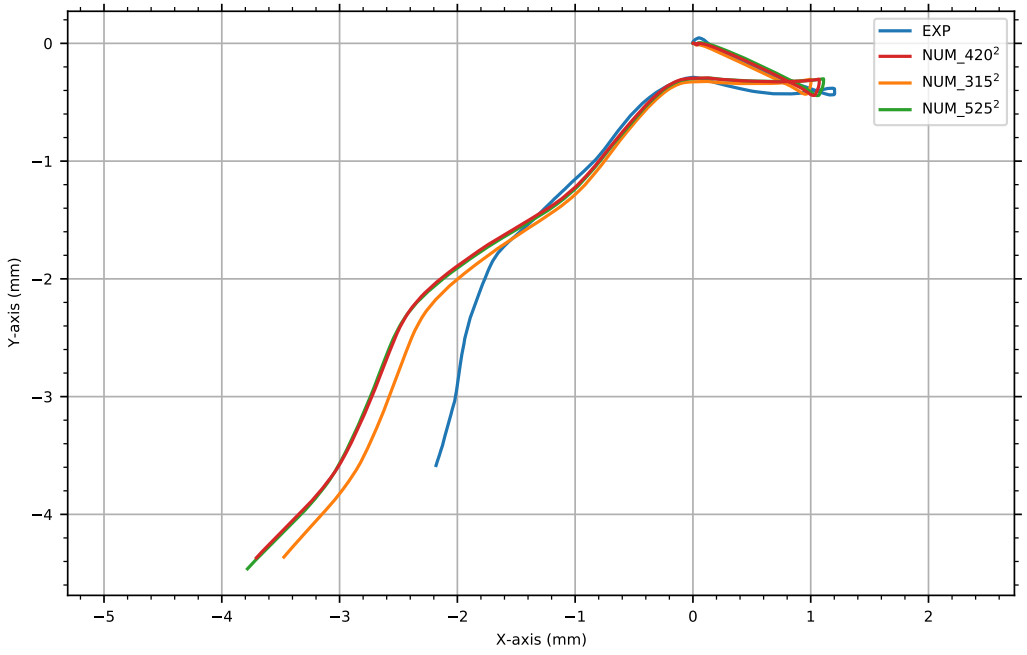


FIGURE 4.14: **Convergence of the numerical solution according to the computational mesh size.** All three resolutions with $N_x = 315$ (orange), $N_x = 420$ (red), $N_x = 525$ (green), present the same global trend regarding the experimental trajectory. The solutions obtained by the two most refined meshes are nearly identical.

Only with three different grid resolution, we can conclude the solution is well-converged from $\Delta x = 2.857 \times 10^{-2}$ mm *i.e.* $N_x = 420$ as a finer grid does not seem to significantly improve the numerical approximation. However, a coarser grid slightly differs in the end of the fast-swimming stage. As a result, we mainly used $\Delta x = 2.857 \times 10^{-2}$ mm for all the following numerical simulations.

The impacts of the computational grid size on the resulting energetics have been recapitulated in Table 4.2. Overall, no significant effect is visible on the energetic quantities although E_t and CoT are slightly better in the 525-case. In consequence, the center-of-mass trajectory is more sensitive to grid size relatively to energetics.

TABLE 4.3: Evaluation of the impact of the penalization method.

Pen	P_{max} (mW)	$\Delta^r_{P_{max}}$	P_{mean} (mW)	$\Delta^r_{P_{mean}}$
order1	2.181×10^{-2}	-	0.998×10^{-2}	-
order2	2.288×10^{-2}	4.924×10^{-2}	1.024×10^{-2}	2.616×10^{-2}
Pen	E_t (mJ)	$\Delta^r_{E_t}$	CoT (J/m)	Δ^r_{CoT}
order1	5.118×10^{-4}	-	7.255×10^{-5}	-
order2	5.259×10^{-4}	2.754×10^{-2}	7.391×10^{-5}	3.507×10^{-2}

For each quantity q, the relative numerical error Δ^r_q denotes the relative difference with the chosen size $N_x = 420$.

4.2.2.2 Order of the penalization method

Based on the previously fixed mesh size, we evaluated the effects of a second-order accuracy penalization method (Fig. 4.15). The corresponding comparison of energetic performances can be found in Table 4.3.

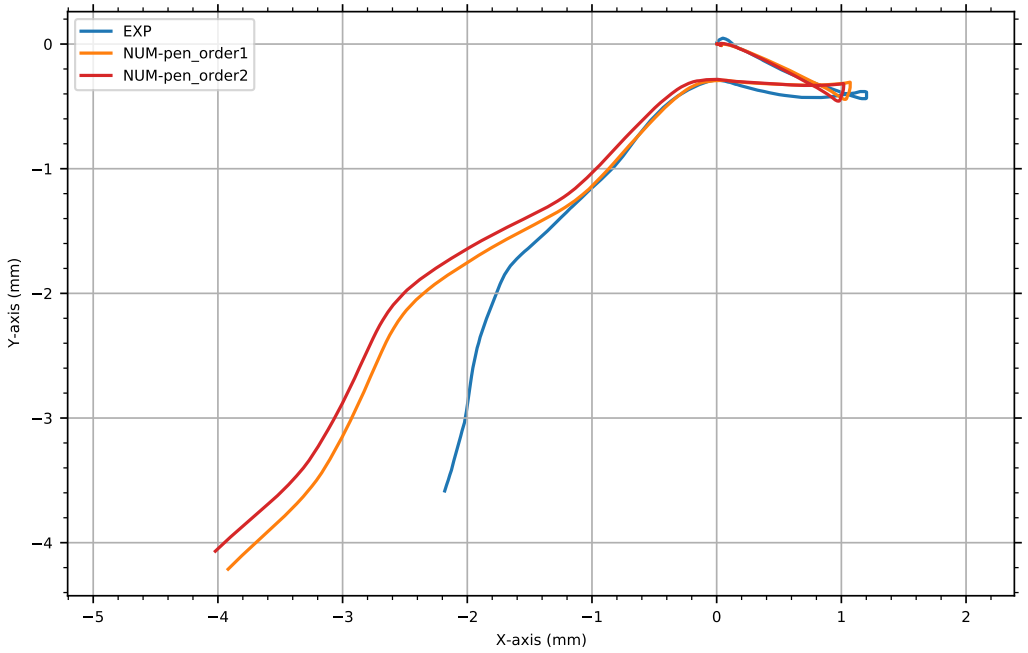


FIGURE 4.15: **Trajectory comparison between a first-order and a second-order accuracy penalization method.** The second-order penalization method (red) and the standard first-order penalization method (orange) have both been described in chapter 2.

Only a slight difference is noticeable in kinematics when using a higher-order penalization scheme near the interface. As for the energetic performance, the variation is not significant enough relatively to the cost in computational time. Indeed, the second-order penalization accuracy was found approximately 30% longer which is considerable regarding the overall gain in precision. As a result, we considered the first-order accuracy for the penalization scheme provided the space step of the grid was fine enough meaning $\Delta x = 2.857 \times 10^{-2}$ mm.

4.2.2.3 A 3D *Procrustes Analysis*

In the previous chapter, we detailed the *Procrustes Analysis* process we used for obtaining the body deformations without rigid movements. In particular, the calculation of body rotation was not straightforward. That is why, we first assessed the importance of a 3D *Procrustes Analysis* (Fig. 4.16). The impact is significant and particularly noticeable during the C-start stage: the direction at the end of the counter-bend corresponds to the experimental one. Instead, the body rotation is not important enough with no 3D calculation of the rotation.

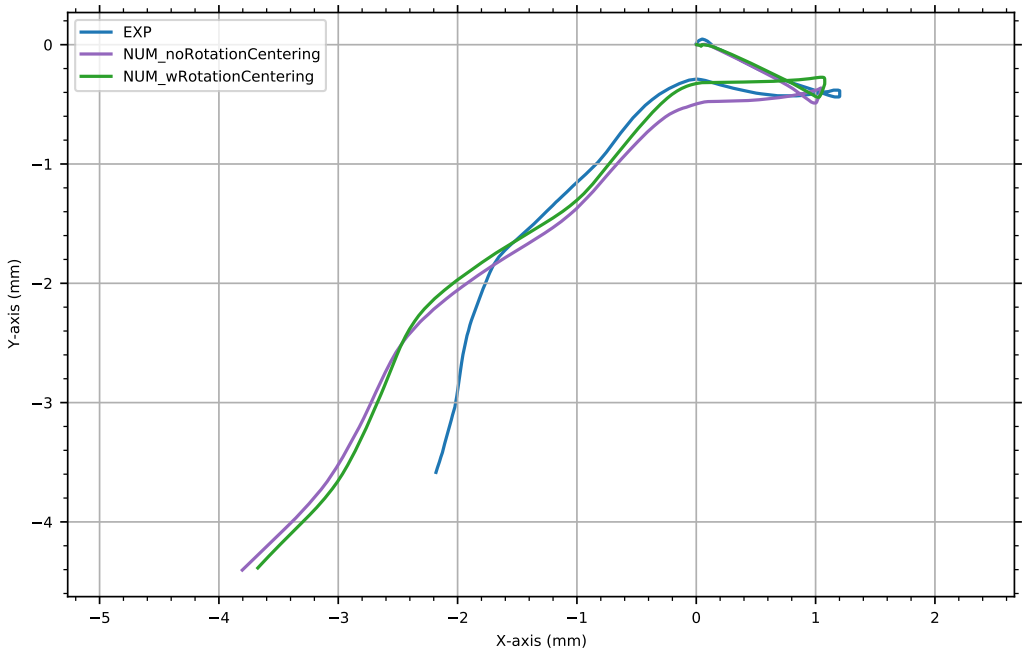


FIGURE 4.16: **Numerical trajectory based on a 3D *Procrustes Analysis*.** In the original simulation (violet), a 3D center of mass is considered but the body rotation is untouched. In the improved simulation (green), both the center of mass and the body rotation are computed in 3D. Note all *Procrustes Analysis* processes have been conducted without the correction step.

Then we compared the actual consequences of our enhancement step in the *Procrustes Analysis* step (Fig. 4.17). As previously, the numerical trajectory fits much more with the experimental data when we use proper body deformations meaning a consistent *Procrustes Analsysis* process.

All the previous numerical simulations have been quantified by considering the energetic performances (Table 4.4). Overall, relative numerical errors are of the order of a few percent and the improvements in body rotation do not impact significantly the energetic performance, unlike the swimming kinematics.

Afterward, we also considered the 3D volume mass to compute adequately the 3D body rotation during the 3D *Procrustes Analysis* step. The comparison of the associated numerical simulations (Fig. 4.18) raised no significant difference between the two

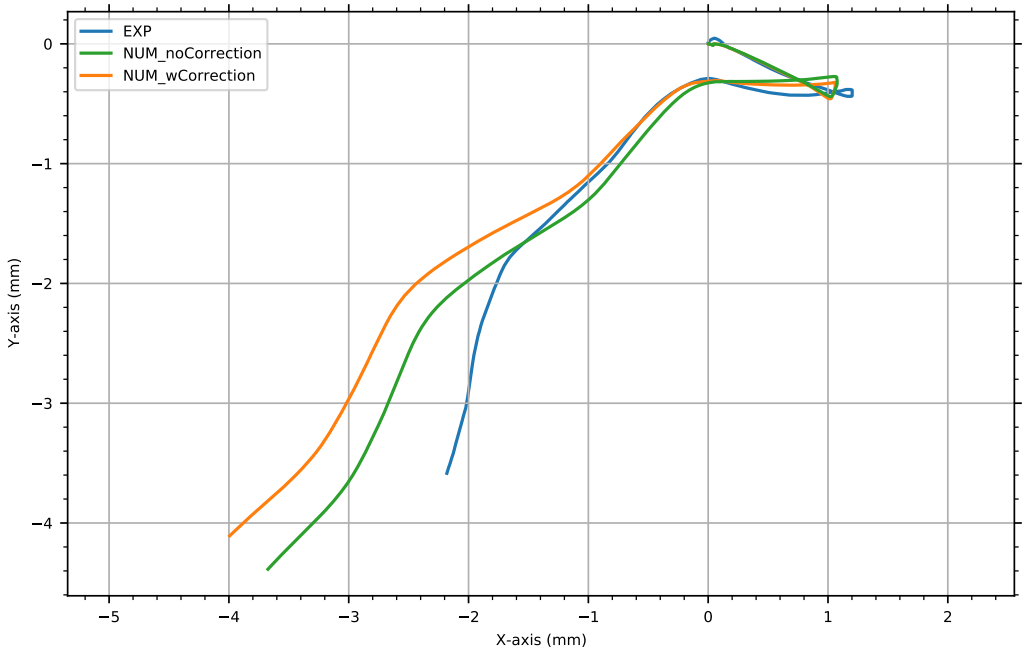


FIGURE 4.17: **Illustration of the impacts of the *Procrustes Analysis* correction on the center-of-mass trajectory.** The improved *Procrustes Analysis* results in a slightly better simulation (orange) compared to a basic *Procrustes Analysis*-based simulation (green). Note the latter is identical to Fig. 4.16.

TABLE 4.4: Evaluation of the impact of the 2D *Procrustes Analysis*.

PA	P_{max} (mW)	$\Delta^r_{P_{max}}$	P_{mean} (mW)	$\Delta^r_{P_{mean}}$
2D-like	2.153×10^{-2}	-	0.989×10^{-2}	-
noCor	2.253×10^{-2}	4.649×10^{-2}	1.018×10^{-2}	2.996×10^{-2}
noRot	2.297×10^{-2}	6.666×10^{-2}	0.970×10^{-2}	1.866×10^{-2}
PA	E_t (mJ)	$\Delta^r_{E_t}$	CoT (J/m)	Δ^r_{CoT}
2D-like	5.070×10^{-4}	-	7.112×10^{-5}	-
noCor	5.244×10^{-4}	3.440×10^{-2}	7.356×10^{-5}	3.426×10^{-2}
noRot	4.987×10^{-4}	1.627×10^{-2}	7.061×10^{-5}	7.180×10^{-3}

For each quantity q, the relative numerical error Δ^r_q denotes the relative difference with the numerical simulation performed with a "2D-like" and corrected rotation computation in the *Procrustes Analysis* step.

Procrustes Analysis methods. As a result, we only considered "3D-like" *Procrustes Analysis* in the following simulations as the computation of the 3D tilt angle is more realistic.

All the previous numerical simulations have been quantified by considering the energetic performances (Table 4.5). Overall, the relative numerical errors are inferior to 1% regardless of the *Procrustes Analysis* method. As previously, these results suggest the *Procrustes Analysis* step affects especially the swimming kinematics with no considerable change in energetics.

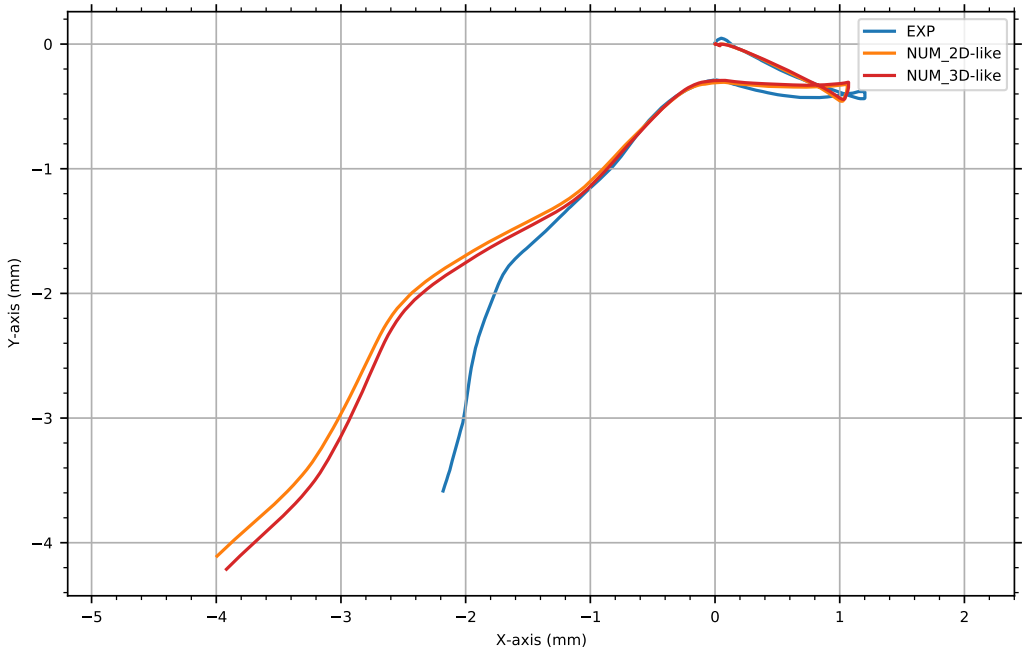


FIGURE 4.18: **Impact of the mass volume in the *Procrustes Analysis* step, on the numerical simulation.** The 3D-like simulation (red) used a *Procrustes Analysis* which takes into account the actual 3D mass for computing the rotation. Instead, the 2D-like simulation (orange) has been based on a *Procrustes Analysis* which considered the dorsal projection of the 3D volume for weighting the 3D rotation calculation. Note the latter is identical to Fig. 4.17.

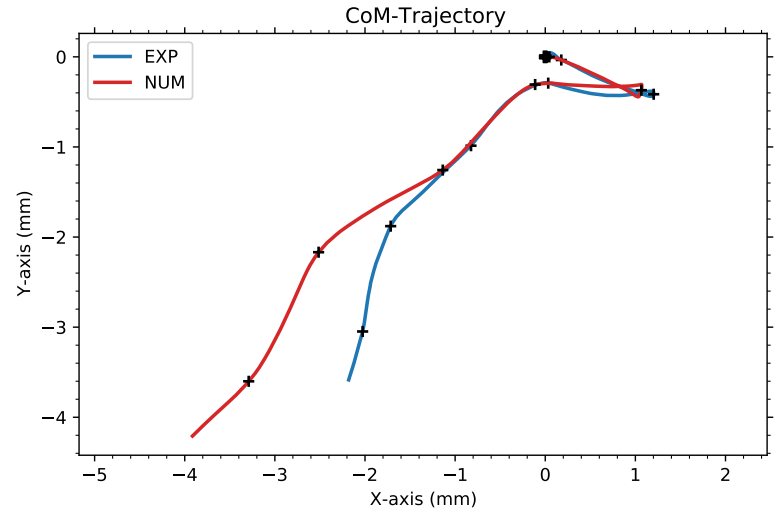
TABLE 4.5: Evaluation of the impact of the 3D *Procrustes Analysis*.

PA	P_{max} (mW)	$\Delta^r_{P_{max}}$	P_{mean} (mW)	$\Delta^r_{P_{mean}}$
3D-like	2.181×10^{-2}	-	0.998×10^{-2}	-
2D-like	2.153×10^{-2}	1.259×10^{-2}	0.989×10^{-2}	9.699×10^{-3}
PA	E_t (mJ)	$\Delta^r_{E_t}$	CoT (J/m)	Δ^r_{CoT}
3D-like	5.118×10^{-4}	-	7.140×10^{-5}	-
2D-like	5.070×10^{-4}	9.378×10^{-3}	7.112×10^{-5}	3.923×10^{-3}

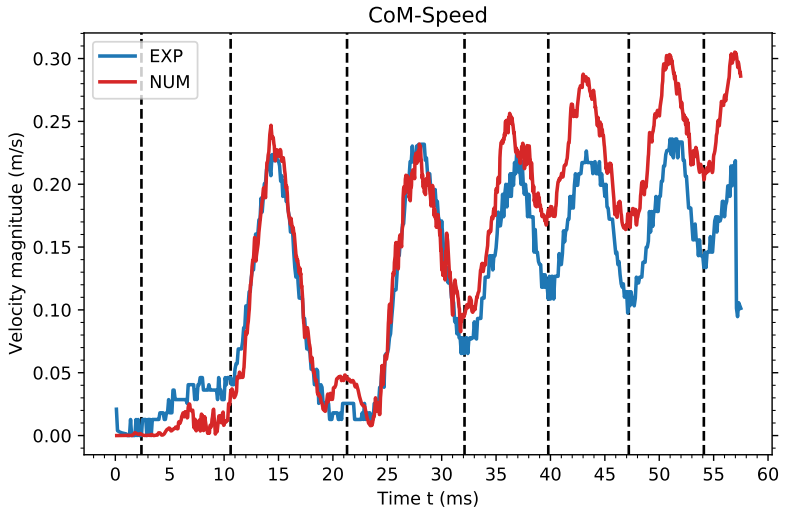
For each quantity q, the relative numerical error Δ^r_q denotes the relative difference with the numerical simulation performed with a "3D-like" and well-smoothed rotation computation (smooth1.3) in the *Procrustes Analysis* step.

4.2.3 Numerical validation

In this particular escape motion, the two first modules namely the C-bend and counter-bend motions *i.e.* times between 0 and 32 ms, are very well-reproduced regarding the direction of the center-of-mass trajectory or the center-of-mass velocity (Fig. 4.19). This result is particularly remarkable considering how strong the C-bend and counter-bend deformations are. Nevertheless, the numerical simulation overestimates the velocity produced during the fast-swimming stage. Indeed, the zebrafish body accelerates as soon as it engages the first tail-beat of fast-swimming. In parallel, the computed rotation (Fig. 4.20) is less accurate from time $t = 32$ ms, especially in the last tail-beat around time $t = 54$ ms, and in consequence, the resulting escape swimming direction does differ slightly from the experimental one (Fig. 4.19). Moreover, notice the center-of-mass velocity reached during the counter-bend motion is identical to the C-bend translation

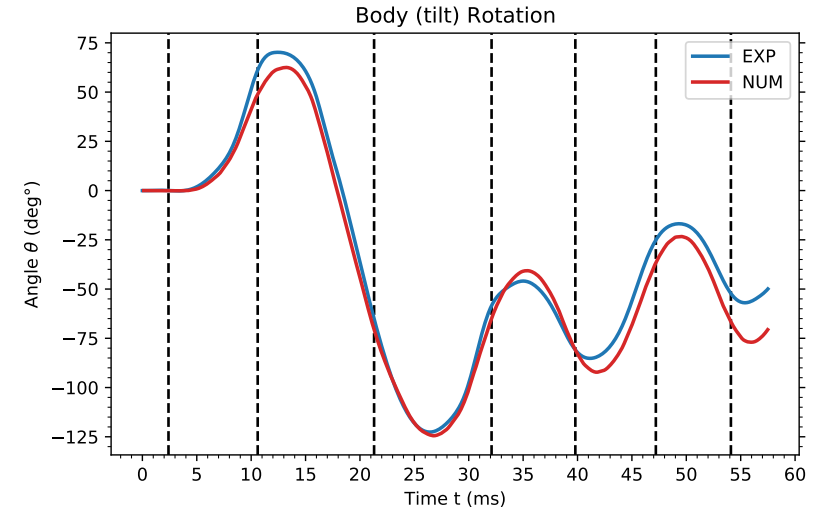


(A)

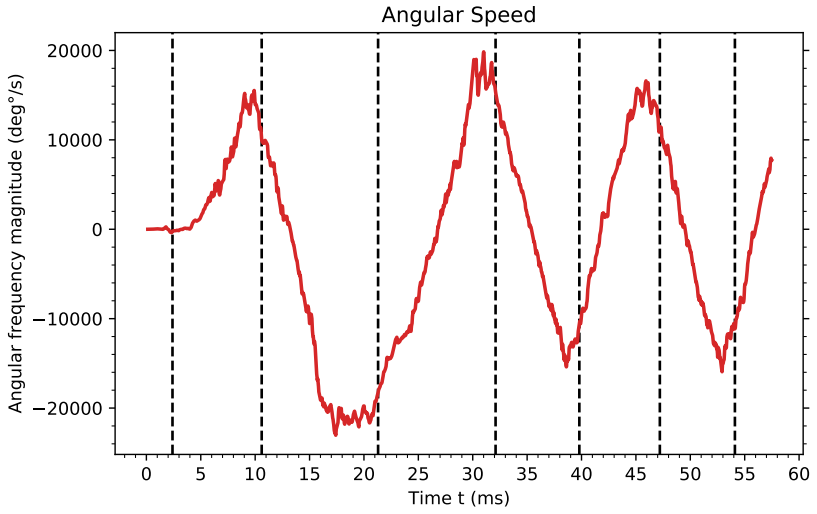


(B)

FIGURE 4.19: **Comparison of the center-of-mass kinematics between experimental measures (blue) and the numerical solution (red).** Overall, the trajectory (Fig. 4.19 A) and velocity magnitude (Fig. 4.19 B) are well-reproduced despite the slight acceleration in the beginning of the fast-swimming stage (t 32 ms). Note vertical lines denote the timing of deformation extrema (Fig. 4.8). The corresponding occurrence are also represented by the black markers on the trajectory.



(A)



(B)

FIGURE 4.20: **Visualization of the rotation kinematics.** On the global body rotation (Fig. 4.20 A), we can see the comparison with the experimental rotation issued from the *Procrustes Analysis*. In addition, we represented the rotation velocity (Fig. 4.20 B). Note vertical lines denote the timing of deformation extrema (Fig. 4.8).

speed despite stronger deformation kinematics. In fact, the bend amplitude seems to only affect the body rotation as both rotation angle and rotation velocity are higher for the counter-bend motion than for the C-bend.

4.2.4 A sensitivity analysis

In this section, we present the various test cases we conducted to assess the limits of our numerical approach. In particular, we intended to reduce the numerical error due to the fast-swimming acceleration.

TABLE 4.6: Evaluation of the impact of the data-smoothing filter.

filt	P_{max} (mW)	$\Delta^r_{P_{max}}$	P_{mean} (mW)	$\Delta^r_{P_{mean}}$
5	2.181×10^{-2}	-	0.998×10^{-2}	-
no	0.119	4.470	5.838×10^{-2}	4.848
11	1.839×10^{-2}	1.568×10^{-1}	0.831×10^{-2}	1.672×10^{-1}
filt	E_t (mJ)	$\Delta^r_{E_t}$	CoT (J/m)	Δ^r_{CoT}
5	5.118×10^{-4}	-	7.140×10^{-5}	-
no	29.978×10^{-4}	4.858	38.967×10^{-5}	4.457
11	4.276×10^{-4}	1.645×10^{-1}	6.059×10^{-5}	1.514×10^{-1}

For each quantity q, the relative numerical error Δ^r_q denotes the relative difference with the chosen size $N_x = 420$.

4.2.4.1 Data noise

As the 3D snapshots are reconstructed directly from the experimental video frames and are above all based on the midline movements, a numerical noise is naturally present. This data noise is particularly emphasized when the acquisition frequency is important meaning experimental video frames are very close due to the numerical error introduced in the midline approximation step.

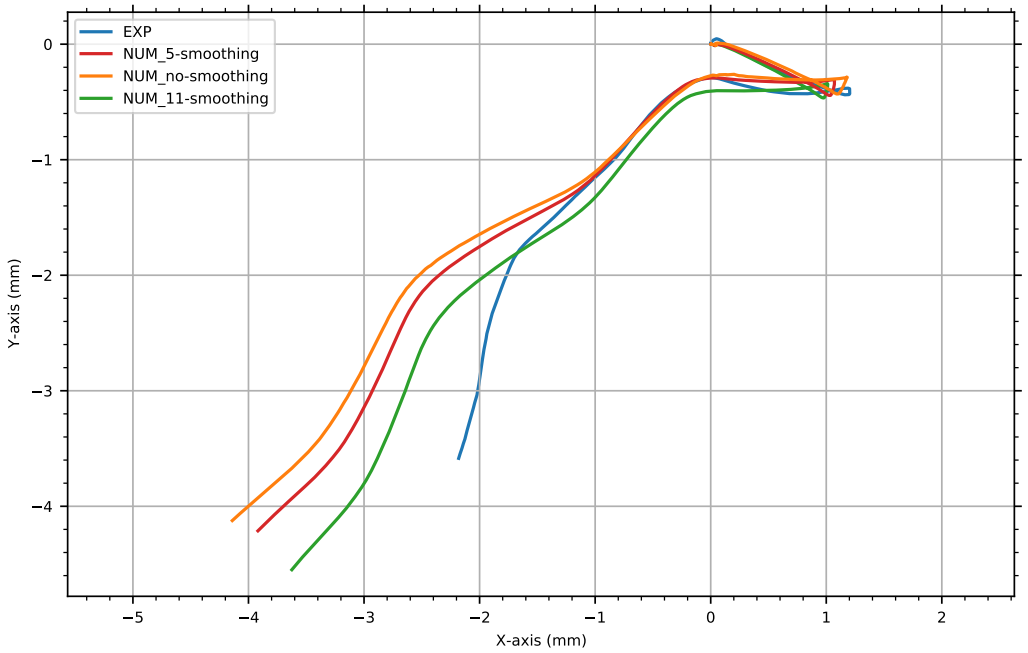


FIGURE 4.21: **Impact of the smoothing of experimental data on the trajectory.** Initially, numerical simulations have been performed by using the raw experimental data (orange). Afterward, we applied a mean filter to smooth in space the data noise, over a range of a 5-node (red) and a 11-point (green) window.

The comparison of different smoothing values (Fig. 4.21) showed a reasonable data smoothing namely the 5-smoothing, is sufficient to preserve the global trend of the center-of-mass trajectory. More important values such as the 11-smoothing should be avoided as the center of mass and the Lagrangian markers might be too much impacted. Note the energetics has been recapitulated in Table 4.6. Similarly, we did not smooth in time the Lagrangian markers as we observed significant deviations in the center-of-mass

TABLE 4.7: Evaluation of the impact of the 3D *Procrustes Analysis*.

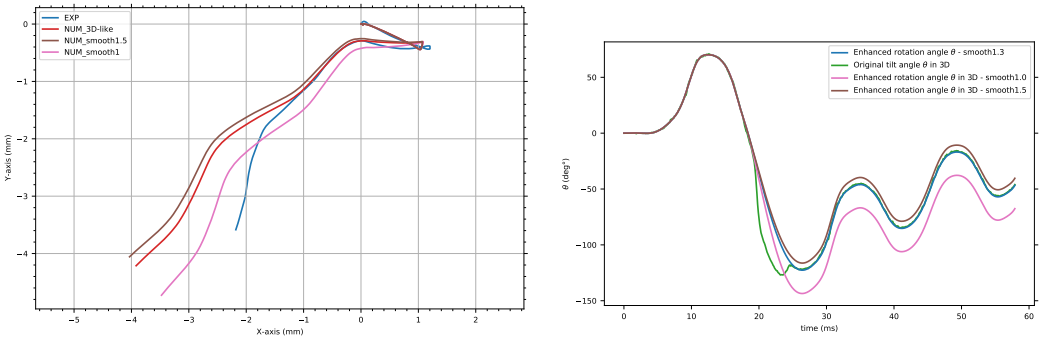
PA	P_{max} (mW)	$\Delta^r_{P_{max}}$	P_{mean} (mW)	$\Delta^r_{P_{mean}}$
3D-like	2.181×10^{-2}	-	0.998×10^{-2}	-
smooth1	2.183×10^{-2}	1.277×10^{-3}	1.000×10^{-2}	1.408×10^{-3}
smooth1.5	2.139×10^{-2}	1.889×10^{-2}	0.995×10^{-2}	2.772×10^{-3}
PA	E_t (mJ)	$\Delta^r_{E_t}$	CoT (J/m)	Δ^r_{CoT}
3D-like	5.118×10^{-4}	-	7.140×10^{-5}	-
smooth1	5.142×10^{-4}	4.701×10^{-3}	7.144×10^{-5}	5.541×10^{-4}
smooth1.5	5.119×10^{-4}	3.541×10^{-4}	7.110×10^{-5}	4.210×10^{-3}

For each quantity q, the relative numerical error Δ^r_q denotes the relative difference with the numerical simulation performed with a "3D-like" and well-smoothed rotation computation (smooth1.3) in the *Procrustes Analysis* step.

calculation in 3D. As a result, we only smoothed the experimental data using a 5-window mean filter to represent the actual body kinematics without affecting the numerical results.

4.2.4.2 *Procrustes Analysis* variations

We need to assess the impact of the final smoothing step on the rotation. Indeed, we used a mean filter after a spline-interpolation step to reconstruct the experimental body rotation. However, the reconstructed body rotation eventually slightly differed from the original experimental data due to the smoothing parameter. Thus, we evaluated the impact of the *Procrustes Analysis* step on the body rotation and the resulting numerical simulation, by assessing the numerical error introduced in the smoothing step of the improved *Procrustes Analysis* (Fig. 4.22).



(A) Numerical simulations: trajectory.

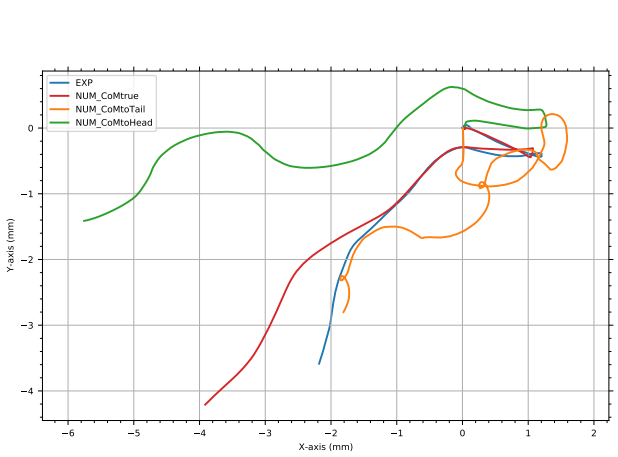
(B) *Procrustes Analysis*: body rotation.

FIGURE 4.22: **Impacts of slight differences in the body rotation computed by the *Procrustes Analysis* on the numerical trajectory of the center of mass.**

All the previous numerical simulations have been quantified by considering the energetic performances (Table 4.7). Overall, the relative numerical errors are inferior to 1% regardless of the *Procrustes Analysis* smoothing parameter. As previously, these results suggest the *Procrustes Analysis* step affects especially the swimming kinematics with no considerable change in energetics.

4.2.4.3 Mass variations

Similarly, we also evaluated the impacts of a different center-of-mass position. Due to several factors such as segmentation errors, geometry approximation, or 2D imaging, the position of the center of mass might not correspond to the actual center-of-mass movements. That is why, we artificially moved the center of mass either near the head or toward the posterior to analyze its influence on the body rotation and the resulting swimming trajectory (Fig. 4.23).



(A) Numerical simulations: trajectory.

(B) Artificial CoM-displacement.

FIGURE 4.23: **Artificial variations of the center of mass.** The center of mass has been moved towards the snout (green) and towards the tail (orange). Note colors correspond between subfigures 4.23 A and 4.23 B. All three representations are aligned on the center-of-mass position in Fig. 4.23 B.

As expected, both test runs do not seem reasonable, especially for rotation motions. Nevertheless, these numerical tests suggest the center of mass is well-approximated during the C-bend and counter-bend motions while numerical errors seem to be introduced during the fast-swimming stage. Indeed, we can see the direction of the fast-swimming phase might fit better with the experiment if the center of mass were located slightly back towards the tail. Therefore, it seems the mathematical model is lacking from supplementary information about biomechanics phenomena occurring during fast swimming.

4.2.4.4 Bi-fluid

Finally, we verified no boundary effects were affecting the numerical solution. For that purpose, we first increased the domain size, especially for the last fast-swimming movements where the zebrafish is close to the domain boundary. Secondly, we also varied the vertical position relatively to the bottom surface or alternatively, modified

the boundary conditions. In conclusion, the computational domain boundary does not impact the overall numerical simulation. In particular, we emphasize an important test case: the modeling of the top boundary as a free-surface in the case where the zebrafish is close to the boundary. To that end, we performed bi-fluid simulations (Fig. 4.24) by representing both air and water mediums. Indeed, we can assume the experimental zebrafish eleuthero-embryo is slowed during the fast-swimming stage due to the proximity of the free-surface as the experimental fluid height could eventually be inferior to 2 mm.

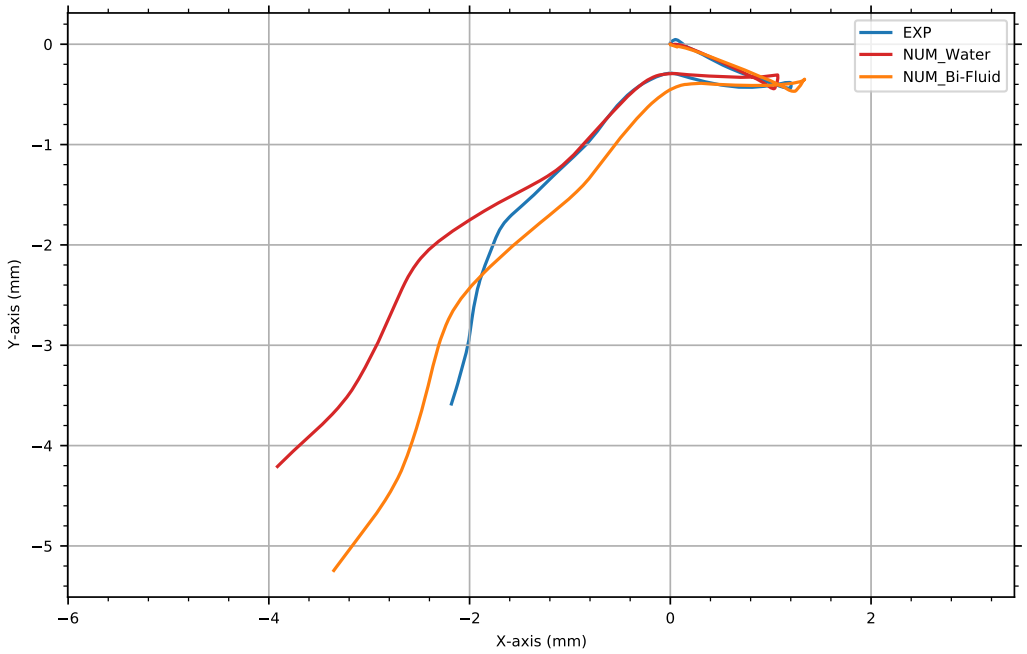


FIGURE 4.24: **Changing boundary conditions: a bi-fluid simulation.**

The trajectory seems to be slightly degraded by conducting a bi-fluid simulation while energetic quantities do not significantly differ (Table 4.8). Most importantly, we can observe an identical acceleration in the fast-swimming stage. This result suggests the zebrafish acceleration is not caused by boundary effects. In definitive, we tested various modeling changes to reduce the fast-swimming stage acceleration with no considerable results. On the one hand, we observed an important deviation between the numerical and experimental solutions which persisted with simple simulation changes. On the other hand, the kinematic difference might be due to modeling issues specific to the considered escape experiment. In order to assess our numerical model accuracy and isolate particular experimental differences, we conducted experiment-driven numerical simulations on a larger experimental sample.

4.3 Reproducibility

To study the reproducibility of our numerical and experiment-driven approach, we compared the numerical solution against the experimental trajectory over a large

TABLE 4.8: Evaluation of the impact of the free-surface modeling.

Phase	P_{max}	$\Delta_{P_{max}}$	P_{mean}	$\Delta_{P_{mean}}$
Water	2.181×10^{-2}	-	0.998×10^{-2}	-
Bi-Fluid	2.167×10^{-2}	6.456×10^{-3}	1.062×10^{-2}	6.367×10^{-2}
Phase	E_t	Δ_{E_t}	CoT	Δ_{CoT}
Water	5.118×10^{-4}	-	7.140×10^{-5}	-
Bi-Fluid	5.509×10^{-4}	7.651×10^{-2}	6.965×10^{-5}	2.451×10^{-2}

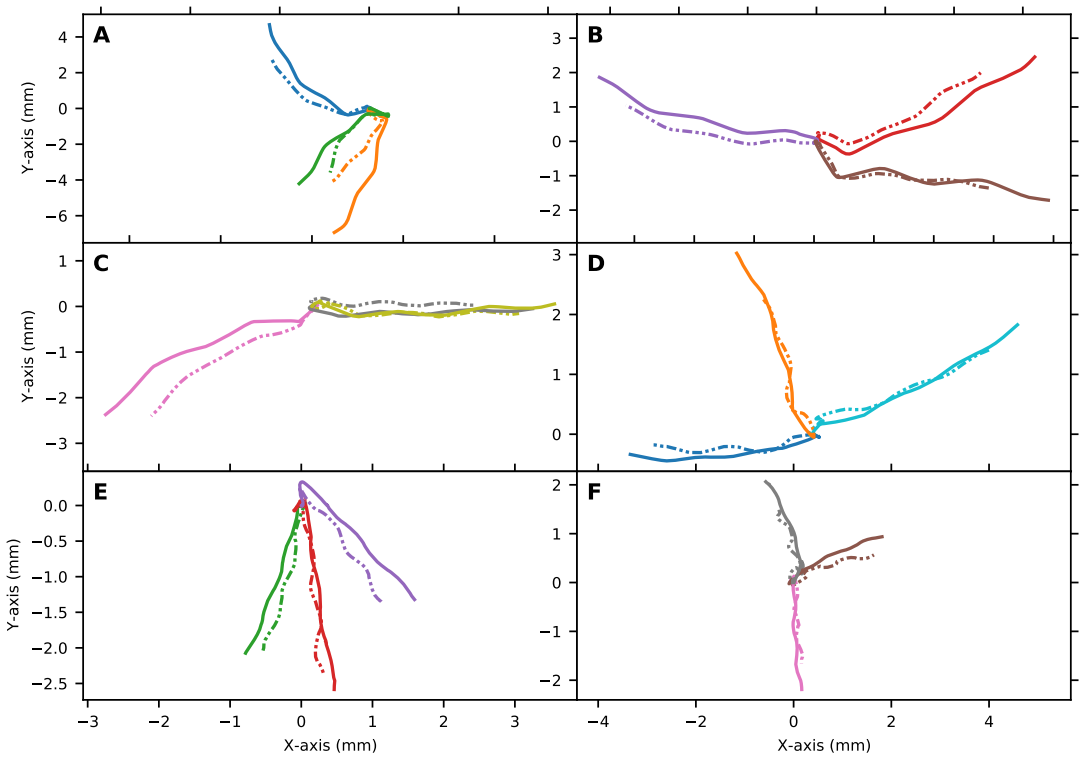
For each quantity q, the relative numerical error Δ_q^r denotes the relative difference with the numerical simulation performed with only a single water phase.

experimental dataset. The description of the dataset will be provided in Chapter 5. In this section, we only focus our attention on the numerical validation of the methodology on various body kinematics (see Fig. 4.25). To be more precise, we altered the body kinematics either by a fluid viscosity change (Fig. 4.25 A) or by exposure to chemical compounds (Fig. 4.25 B).

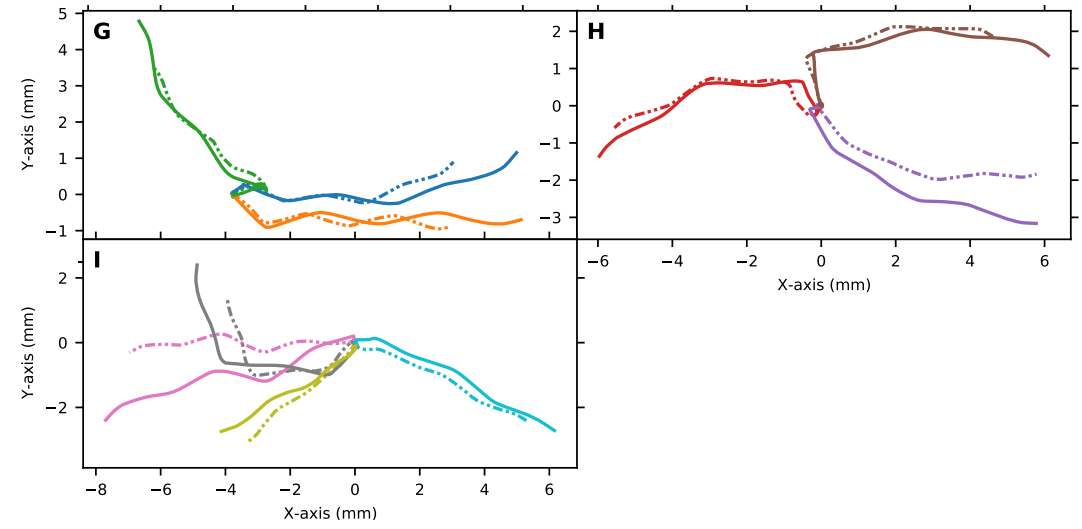
In Fig. 4.25 A-**(A)**, we represented again the initial test case in green. In the two other escape responses, the numerical acceleration starts just after having performed the C-bend motion. In Fig. 4.25 A-**(B)**, only both the red and brown-experiments provide a slight increase in the speed of the numerical solution after the C-bend motion. All kinematic quantities including body rotation and velocity are well-reproduced for the two other experiments. In Fig. 4.25 A-**(C)**, only the green-experiment kinematics is well-reproduced while the pink-experiment provides a satisfying simulation where the zebrafish eleuthero-embryo produces a minor speed increase. The grey-experiment simulation accelerates after the C-bend motion. In Fig. 4.25 A-**(D)**, only the orange-experiment simulation performs a minor acceleration relatively to the experimental speed. The other two numerical simulations reproduce the experimental kinematics with satisfying precision. In Fig. 4.25 A-**(E)**, the green-experiment kinematics is very well-reproduced by the numerical simulation. As for the other two experiments, the numerical simulations are very satisfying despite minor differences against the experimental kinematics. In Fig. 4.25 A-**(F)**, a minor acceleration is visible in the grey-experiment numerical simulation while only minor differences can be seen in other numerical simulation kinematics.

Similarly to water simulations (Fig. 4.25 A-**(A)**), a numerical acceleration can be observed for all three experiments of Fig. 4.25 B-**(G)**, clearly visible after the C-bend motion. In Figs. 4.25 B-**(H)**,(I) only minor differences are visible in kinematics such as velocity, especially after the C-bend motion. Only the green-experiment is particularly well-reproduced.

All relative numerical errors regarding the main kinematic variables: raw displacement, total traveled distance, fast-swimming traveled distance, and fast-swimming average velocity, have been recapitulated in Table 4.9, for each experiment. We highlight numerical errors could reach up to 30%, even in the case of overall satisfying results. At first view, the best numerical simulations seem to provide smaller displacement



(A) Viscosity cases.



(B) Neuro-toxicant-based cases.

FIGURE 4.25: **Reproducibility of experiment-driven numerical simulations.** For the viscous test cases (Fig. 4.25 A), we immersed zebrafish eleuthero-embryos into water **(A)**, and viscous fluids with a ratio of 1.1 **(B)**, 2.3 **(C)**, 5 **(D)**, 10 **(E)**, 15 **(F)** against water viscosity. For the toxicological test cases (Fig. 4.25 B), we simulated up to two drug concentrations (Figs. **(H)**, **(I)**) alongside the control group **(A)**. Three experiments have been simulated for each experimental condition, corresponding to each color. For each experiment, we represented the experimental trajectory (dashed line) and its associated numerical simulation (solid line).

TABLE 4.9: Numerical validation: relative numerical errors of translation kinematics. Note we quantified the trajectory comparisons recapitulated in Fig. 4.25. Bold values denote a relative error inferior to 30%.

	run	Displacement (%)	Total traveled distance (%)	Fast-swimming traveled distance (%)	Fast-swimming average velocity (%)
A	blue	**14.523**	16.652	26.335	27.332
	orange	44.253	35.422	54.154	55.634
	green	**22.048**	9.561	25.632	24.821
B	red	**24.969**	22.536	39.819	40.824
	violet	**8.623**	8.701	20.675	18.159
	brown	**29.434**	31.244	45.130	45.130
C	pink	**15.449**	12.274	28.470	25.660
	grey	32.096	23.101	40.526	38.699
	yellow	**11.148**	9.734	24.535	25.277
D	cyan	**16.190**	8.665	19.687	18.638
	blue	**11.929**	3.647	11.177	11.190
	orange	33.438	24.779	37.071	37.114
E	green	**1.707**	4.996	4.043	4.611
	red	**6.787**	1.250	5.916	6.180
	violet	**12.867**	8.983	10.609	11.237
F	brown	**14.303**	6.692	10.890	11.595
	pink	31.810	19.009	25.285	24.950
	grey	40.324	15.925	21.759	22.320
G	blue	**21.908**	16.729	32.703	31.894
	orange	**25.837**	25.410	37.195	36.119
	green	**21.563**	18.358	35.004	31.518
H	red	**6.010**	9.436	21.361	20.978
	violet	**4.655**	4.359	11.011	11.008
	brown	**20.093**	15.197	23.929	20.961
I	pink	**11.700**	14.484	20.544	21.527
	grey	**27.041**	19.924	29.940	27.460
	yellow	**10.740**	11.850	11.468	12.460
	cyan	**11.381**	12.110	24.336	18.927

For each quantity q, the relative numerical error Δ_q^r denotes the relative difference with the experimental value.

errors. For example, both all escape responses in Fig. 4.25 B-**(H)** are well-reproduced but numerical errors for the brown simulation seem important by reaching up to 20% while both displacement and total traveled distance errors seem reasonable for the red and violet simulations. Instead, small errors can be obtained for the total traveled distance with no significance as the zebrafish is clearly accelerating (see the green simulation in Fig. 4.25 A-**(A)** or the brown simulation in Fig. 4.25 A-**(F)**). These important numerical errors might be due to relatively long escape times combined with the presence of small modeling issues which we highlighted previously. In addition, note multiple sources of errors may accumulate regarding either the numerical solution or the experimental approximations especially in the case of experiment-based numerical simulations. However, multiple statistical comparisons Tukey's tests were performed for both 4.25 A and 4.25 B datasets, and showed all relative numerical errors were not dependent on the experimental conditions (Table 4.10).

TABLE 4.10: A one-way ANOVA and multiple comparison Tukey's test: Evaluation of numerical errors across all experimental conditions.

	Traveled distance	Displacement	Fast-swimming Distance	Fast-swimming Velocity
μ_w vs. μ_1	ns (>0.9999)	ns (0.9871)	ns (>0.9999)	ns (>0.9999)
μ_w vs. μ_2	ns (0.9765)	ns (0.967)	ns (0.9967)	ns (0.9849)
μ_w vs. μ_5	ns (0.8879)	ns (0.9817)	ns (0.7254)	ns (0.6981)
μ_w vs. μ_{10}	ns (0.3934)	ns (0.358)	ns (0.0717)	ns (0.0838)
μ_w vs. μ_{15}	ns (0.9482)	ns (>0.9999)	ns (0.5169)	ns (0.5343)
μ_1 vs. μ_2	ns (0.9710)	ns (>0.9999)	ns (0.9972)	ns (0.996)
μ_1 vs. μ_5	ns (0.8737)	ns (>0.9999)	ns (0.7354)	ns (0.7862)
μ_1 vs. μ_{10}	ns (0.3758)	ns (0.6954)	ns (0.0739)	ns (0.1093)
μ_1 vs. μ_{15}	ns (0.9390)	ns (0.9584)	ns (0.5271)	ns (0.628)
μ_2 vs. μ_5	ns (0.9992)	ns (>0.9999)	ns (0.9281)	ns (0.9608)
μ_2 vs. μ_{10}	ns (0.7848)	ns (0.7773)	ns (0.1509)	ns (0.2291)
μ_2 vs. μ_{15}	ns (>0.9999)	ns (0.919)	ns (0.7761)	ns (0.8749)
μ_5 vs. μ_{10}	ns (0.0.9269)	ns (0.7239)	ns (0.5338)	ns (0.6157)
μ_5 vs. μ_{15}	ns (>0.9999)	ns (0.9469)	ns (0.9989)	ns (0.9997)
μ_{10} vs. μ_{15}	ns (0.8556)	ns (0.2749)	ns (0.7418)	ns (0.7752)

	Traveled distance	Displacement	Fast-swimming Distance	Fast-swimming Velocity
H_2O vs. $DMSO_{0.1}$	ns (>0.9999)	ns (0.9589)	ns (>0.9999)	ns (0.9828)
H_2O vs. CPO_{100}	ns (0.3389)	ns (0.2124)	ns (0.2122)	ns (0.1456)
H_2O vs. CPO_{150}	ns (0.7289)	ns (0.4219)	ns (0.2929)	ns (0.1879)
$DMSO_{0.1}$ vs. CPO_{100}	ns (0.3662)	ns (0.4023)	ns (0.2281)	ns (0.2425)
$DMSO_{0.1}$ vs. CPO_{150}	ns (0.7648)	ns (0.7093)	ns (0.3149)	ns (0.317)
CPO_{100} vs. CPO_{150}	ns (0.8224)	ns (0.9022)	ns (0.9792)	ns (0.9851)

For each kinematic variable, we show the experimental and numerical impacts between each viscosity condition (N=3) by summarizing the p-value. Statistics from a 1-way ANOVA Tukey's test. non-significant (ns): $p \geq 0.05$, *: $p<0.05$, **: $p<0.01$, ***: $p<0.001$, ****: $p<0.0001$.

In definitive, all numerical simulations are overall very satisfying as experimental kinematics is well-reproduced. Nevertheless, in specific experiments, the numerical solution seems to overestimate the center-of-mass velocity which may increase the traveled distance, displacements, or a significant change in the final escape direction. Particular direction changes might be raised by *Procrustes Analysis* errors, especially in the estimation of body rotation. Furthermore, in several experiments, the numerical simulation was found extremely accurate relatively to the experimental swimming kinematics regarding all kinematic variables: trajectory, velocity, or rotation. Finally, note only few numerical simulation escape directions such as green-and-orange-experiments (Fig. 4.25 A-**(A)**) or the pink-experiment (Fig. 4.25 B-(I)) differ considerably from the experimental escape response after two cycles of fast swimming. Similarly, it seems a few amount of numerical simulations as in Figs. 4.25 A-**(D)**,**(E)**,**(F)** present smaller swimming amplitudes regarding the center-of-mass trajectory.

4.4 Additional considerations

In consequence, we also examined attentively the presence of 3D motions such as pitch and roll rotations. Indeed, we remind the reader the zebrafish eleuthero-embryo is inside a 3D medium whereas our 3-DoF mathematical model constrains the movements to a horizontal planar motion. Therefore, significant numerical errors could be raised in case of the apparition of experimental pitch and roll motions. However, no considerable, if

any, 3D movements were visible on the experimental video. Furthermore, the eleuthero-embryo zebrafish morphology is constituted of an anterior mass section and a posterior mass body surrounded by a thin layer of cells namely the MFF. Thus, the mass density might not be homogeneous due to the differences in texture. Actually, it is one of the reasons we artificially modified the center-of-mass position. Another reason could be the difference in geometry between our 3D reconstruction and the actual biological and unique shape. For instance, the thickness of the MFF might be important to zebrafish propulsion. As an additional remark, notice our numerical model considers the zebrafish eleuthero-embryo as entirely rigid meaning we did not implement any elastic model in addition to the experimentally observed body kinematics. Nevertheless, elastic modeling could be considered for certain parts of the zebrafish body such as the MFF, and might enhance the numerical error. Similarly, note both the head and tail tip have been modeled with a rigid segment during the tracking and deformation reconstruction step. First, the head section can be considered rigid as a strong stiffness can experimentally be measured near the head region [89]. Secondly, the tail tip might be very important for generating the zebrafish propulsion and a significant error might be generated although the length of the segment is negligible. On the one hand, we analyzed and proposed various solutions to understand and reduce the numerical error to the experiment, for one specific test case. On the other hand, the numerical error might be highly variable according to the considered body kinematics. Therefore, we next analyzed the numerical error performed on the center-of-mass trajectory on a wide range of experimental test cases.

To conclude the numerical validation, we initially found numerical errors could be important according to the body kinematics. Consequently, we strongly interacted with the mathematical model in order to reduce the numerical error of the initial experimental case *i.e.* the acceleration occurring during the fast-swimming stage. The presented results suggest our numerical approach may overestimate the zebrafish propulsion mainly produced by the posterior part and the tail. Despite the possible presence of 3D movements in experiments or a non-homogeneous zebrafish eleuthero-embryo mass, the numerical acceleration might essentially be explained by elastic parts of the body not taken into account in this initial mathematical modeling. Finally, we then used our numerical simulations for studying the escape swimming performances of the validation cases, applied towards biological applications. The next chapter will focus on the description of these biological experiments, the corresponding swimming performance, and the interests of such a powerful numerical tool.

Chapter 5

Applications to biology

This chapter presents how our numerical process and CFD simulations can be embedded within a full experiment-driven biological study. Biologists of the MRGM lab use zebrafish eleuthero-embryos as an animal model for human pathologies and more precisely for studying the locomotion processes and the underlying biological mechanisms. In the previous chapters, we detailed the computational implementation and the full process of swimming reconstruction and CFD simulations. After controlling and verifying that the numerical solution obtained from a few experimental videos, was consistent enough, we first aimed to deeply analyze the original outputs provided by the CFD, in terms of energetics, performances, and wake structure while challenging the reproducibility of our numerical model on a wider basis of experimental escape responses. Ideally, computational modeling should enable the differentiation of locomotor misbehavior from stereotyped escape responses, by using the supplementary and original information about swimming performance. To that end, we focused on two experimental studies. First, a complete study of swimming performances in viscous flows has been conducted. Afterward, we considered a specific chemical compound to alter the zebrafish escape response and initiate a first step toward toxicological experiments.

5.1 A Viscosity study

The primary locomotion experiments consisted of modifying the eleuthero-embryo's surroundings and more precisely the dynamic fluid viscosity. Note the dynamic viscosity is one of the fluid parameters which characterizes the flow regime and can be directly tuned in the CFD code. However, one of the main questions resides in the body kinematics. Indeed, we do not know *a priori* what is the impact of a hydrodynamic regime change on the zebrafish body and more precisely on its escape deformation. Above all, the body deformations include passive motion generated by the surrounding fluid flow and the active part created by the zebrafish eleuthero-embryo. In this regard,

the interaction between passive and active swimming remains largely unknown. That is why, numerical simulations could help in understanding the zebrafish escape behaviors. In definitive, the chapter introduces the experimental study before developing the main computational results from the experiment-based simulations and additional explorations on viscosity change. This section constitutes the preview of a larger computational study for the toxicological and pharmacological investigations which will be introduced in the next section.

5.1.1 Experimental protocol

We have conducted all the experiments within the MRGM biology research team. Basically, all experiments consisted of recording escape swimming movements as described in Chapter 3. Nevertheless, we explained the general recording procedure with no specific experimental protocol. In this section, we focus on particular experiments in which we modified the fluid dynamic viscosity in order to challenge the zebrafish escape response. Eventually, numerical simulations will be performed afterward, to reproduce the experimental results and complete the collected data to analyze the escape performances under important viscous forces. We chose to first challenge zebrafish eleuthero-embryos with highly viscous fluids as dynamic viscosity is a direct parameter of the governing Navier-Stokes equations (see Chapter 2). Thus, viscosity is one of the few simulation parameters which can be tuned both in the computational and experimental model. To that end, we first prepared viscous solutions by dissolving solid powder of Dextran 500 (Pharmacosmos A/S, Holbaek, Denmark) into highly-filtered fish water. Dextran has been well-used in the literature across experimental biology [28, 30, 55, 79] mainly for its physical properties. First, Dextran aqueous solutions are Newtonian fluids [28, 30] which means the shear stress is proportional to the shear rate. Moreover, Dextran aqueous solution modifies the water dynamic viscosity while the fluid density remains untouched [1]. Note the so-called Dextran 500 denotes dextran molecules with a molecular weight of $500,000$ Da. The direct relationship between the concentration of Dextran 500 and the fluid dynamic viscosity has been extrapolated from the literature [28].

We considered Dextran concentrations of 0%, 1%, 3%, 5%, 7%, 9% corresponding to a dynamic viscosity of 0.83 mPa.s, 1.1 mPa.s, 2.3 mPa.s, 5 mPa.s, 10 mPa.s, 15 mPa.s at 28 °C, respectively (see Fig. 5.1). In what follows, we refer to each respective viscous condition as: μ_w, μ_1, μ_2, μ_5, μ_{10}, μ_{15}. To conduct this biological study, we used eleuthero-embryos aged of 5 dpf. For the experimental protocol, we essentially individually immersed eleuthero-embryo into the prepared viscous solutions, after rinsing it before the transit to limit the potential introduction of water inside the viscosity experimental cell. For each individual, we recorded between three and five escape responses. After the experiment, videos were reviewed in order to conserve

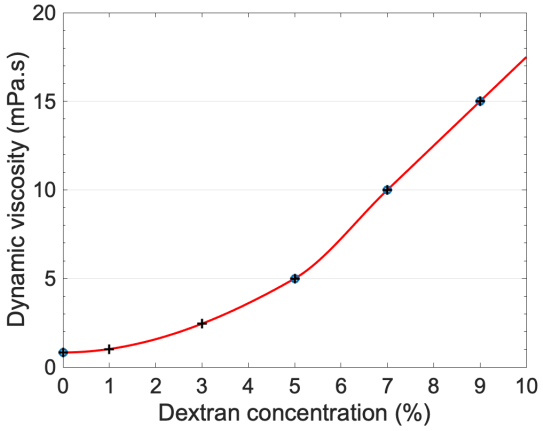


FIGURE 5.1: **Relationship between Dextran concentration and fluid dynamic viscosity μ.** Note the concentration denotes the weight/volume ratio (w/v). The circles denote the known estimated values from [28] for 5%, 7% and 9% of Dextran concentration. The viscosity value for 1% and 3% of Dextran solution is extrapolated from a cubic spline interpolation.

the best escape video corresponding to each eleuthero-embryo. In total, we obtained approximately 85 escape locomotions in viscous flows and more than 30 escape swimming in water, at $10,000$ fps. Again, we eliminated all low-quality videos in terms of imaging and motion capture duration, which resulted in a set of 70 plus 14 experimental videos, including between 10 and 15 videos per viscous condition. Each fast-start video has then been post-processed to analyze the swimming kinematics while segmenting the first 90 ms of the escape locomotion, according to the custom Matlab algorithm detailed in Chapter 3. Notice we only analyzed the swimming kinematics of the C-bend and counter-bend motions as well as two cycles of fast swimming *i.e.* 4 complete tail-beats, based on the calculation of rotation motion *via* the *Procrustes Analysis* (Fig. 5.2). However, particular videos have not been able to capture all the fast-swimming tail-beats with a highly stereotyped pattern. In the end, we selected $N = 3$ escape videos per viscosity condition, based on both experimental and modeling constraints. Indeed, we first chose escape responses which were the most stereotyped while performing four fast-swimming tail-beats, at least. Additionally, we avoided at maximum near-boundary and non-responsive zebrafish. Afterward, we selected the best experimental motion regarding modeling constraints which include good image quality for the segmentation step *i.e.* satisfying focus and clear fluid, eleuthero-embryo at rest initially *i.e.* straight body and above all, planar escape response which is the strongest hypothesis of the model. Finally, we currently cannot model with Lagrangian markers the strongest C-bend motions where the tail of the eleuthero-embryo touches or crosses over its head. Given the number of constraints, we chose to drastically reduce the number of experimental escape videos to run our numerical modeling with high-quality experimental imaging.

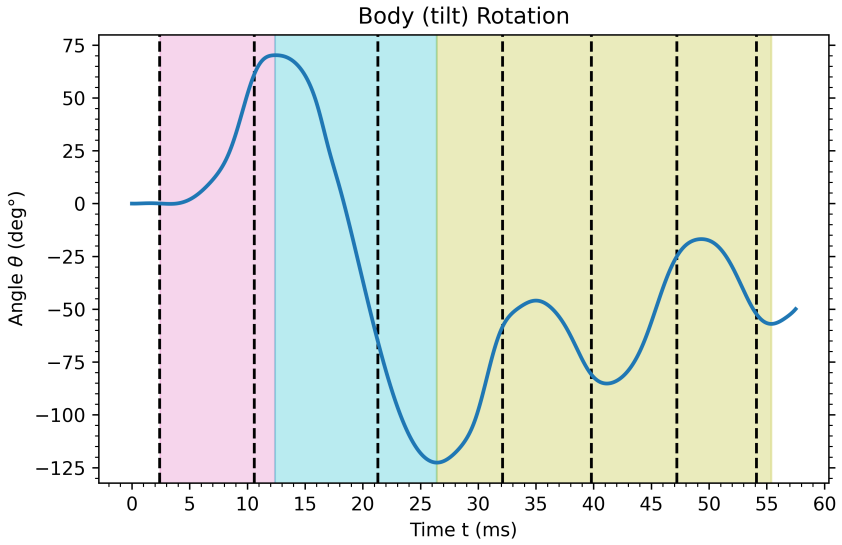
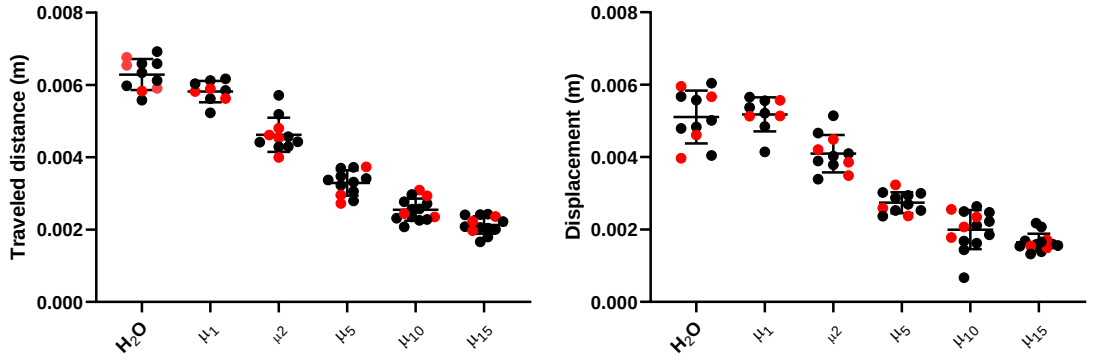


FIGURE 5.2: **The body rotation computed from the *Procrustes Analysis* was used to characterize the different stages.** The total escape time was considered as the sum of the C-bend-driven (pink) and counter-bend-driven (blue) rotation phases plus two cycles of fast-swimming rotation (green). Note the example is the initial test case of Chapter 4 and dashed lines correspond to extrema of body deformations (Fig. 4.8).

5.1.2 Experimental kinematic performance

From the 2D *Procrustes Analysis*, the kinematic analysis consisted of computing various quantities between the three phases defined by the tilt angle meaning the body rotation (Fig. 5.2): the first rotation phase (pink) is due to C-bend motion while the second rotation (blue phase) is driven by the counter-bend. The last four rotation motions (green) are caused by the fast-swimming cycles. Notice the tilt angle does not necessarily correspond to the bend amplitude which represents the deformation amplitude and at this stage, the body deformation has not been quantified yet. For our specific recording conditions, the in-house tracking *FLOTE* software available at the MRGM lab could not compute properly and accurately the angular deformations on such high-resolution datasets in space and time (see Fig. 4.8). Unlike the bend amplitude which is calculated after extracting the midline kinematics, the body rotation is directly accessible through the *Procrustes Analysis*. That is why, all kinematic quantities were defined based on body rotation and not the body deformation, in a first stage. In particular, we distinguished the fast-swimming rotation stage from the C-start-driven motions which can be more variable according to the escape response. Similarly, the initial time was based on the apparition of the first rotation movement as the rotation was more sensitive to deformations than translation kinematics, and that we observed a highly variable latency time, up to 20 ms *i.e.* 200 video frames. First, we computed the total traveled distance and the global raw displacement associated (Fig. 5.3). The first distance corresponds to the exact distance traveled by the 2D center of mass while the latter is the length of the straight line connecting the initial and final center-of-mass positions. Biologists

use displacements in addition to distances in order to measure the escape performance. In the case of misbehavior, the center of mass might travel a lot without covering an important displacement from the starting point. Obviously, both distances would present the same trend in case of a normal escape behavior *i.e.* moving away from the danger, the stimulated position. We can see both distances are strongly impacted by fluid viscosity as we can notice a rapid decay (Fig. 5.3). Above all, notice how reproducible the results are: each viscous condition can easily be represented by a few points only as kinematics were highly stereotyped and impacted by viscosity. Note we only computed the distances when the escape response included the four tail-beats of fast swimming.

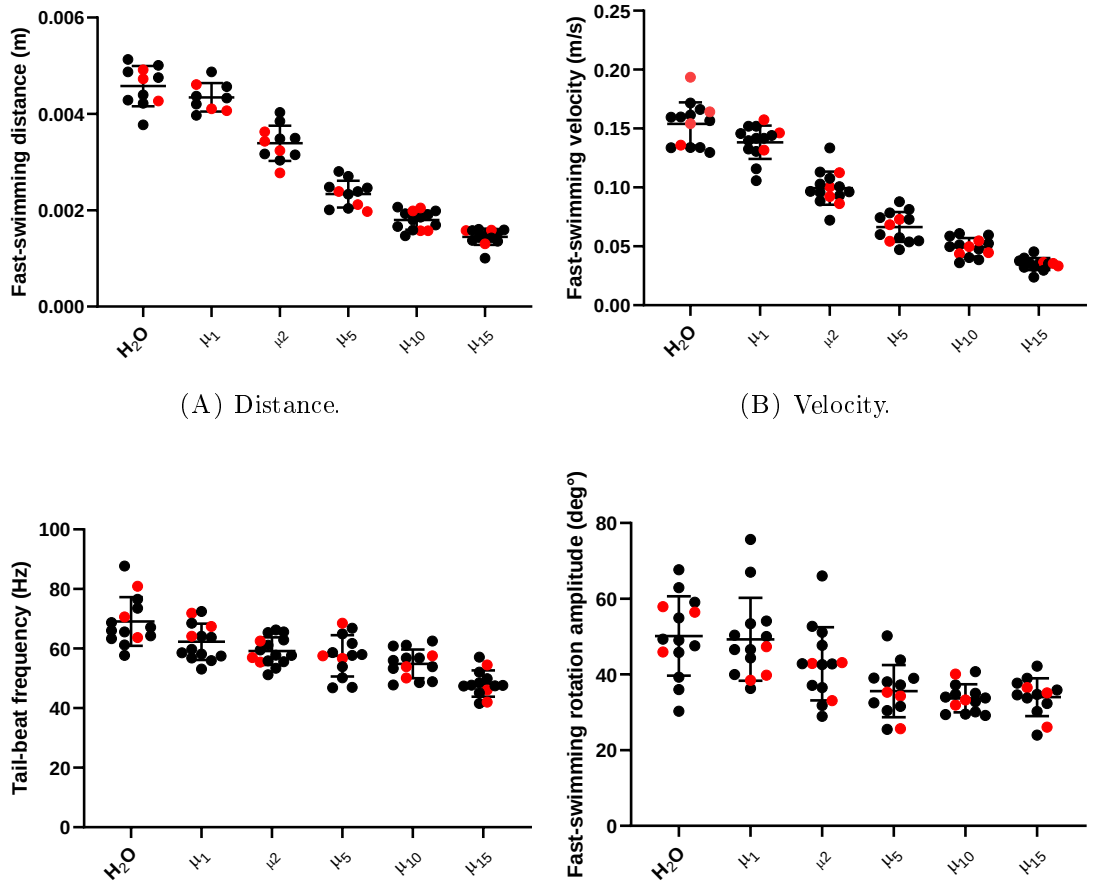


(A) Traveled distance. (B) Displacement.

FIGURE 5.3: **Two distances have been computed in the entire escape locomotion for each experiment and analyzed according to fluid viscosity.** For each viscous condition, we analyzed 11 (μ_w), 9 (μ_1), 11 (μ_2), 12 (μ_5), 14 (μ_{10}), 13 (μ_{15}) experimental videos, for both the total traveled distance (Fig. 5.3 A) and the raw displacement (Fig. 5.3 B). Each experimental escape response is represented by an individual point. Note we show in red the experimental cases we further modeled with the numerical simulation. The mean value is represented alongside the SEM. Multiple statistical comparisons can be found in Table 5.1.

In addition to the latency time, C-start motions and more precisely the C-bend and counter-bend amplitudes were found highly variable, as demonstrated by the C-start rotations (Fig. 5.5) which impacted the overall changes in the escape direction within the experimental cell (Fig. 5.4 D). That is why, we measured several quantities based on the fast-swimming stage only (Fig. 5.4) including the fast-swimming traveled distance (Fig. 5.4 A), the fast-swimming average velocity (Fig. 5.4 B), the fast-swimming tail-beat frequency (Fig. 5.4 C), and fast-swimming rotation amplitude (Fig. 5.4 D). We assumed the tail-beat frequency of body deformations is well-approximated by body rotations as only one tail-beat is performed during every rotation change. As for the fast-swimming amplitude, it denotes the mean amplitude of rotation during the last four tail-beats. As previously, the kinematic results were overall very consistent and reproducible, except for the rotation amplitude (Fig. 5.4 D). Due to the high variability explained before, which essentially translates into important exploratory movements especially in water, the latter variable will not be considered in what follows. As for the fast-swimming distance (Fig. 5.4 A) and velocity (Fig. 5.4 B), the trends seem as similar as the global distance and

displacements. In fact, the tail-beat frequency which constitutes the most representative variable of the actual body deformations, seems to present a minor decrease over the considered fluid viscosity range (Fig. 5.4 C).



(A) Distance.

(B) Velocity.

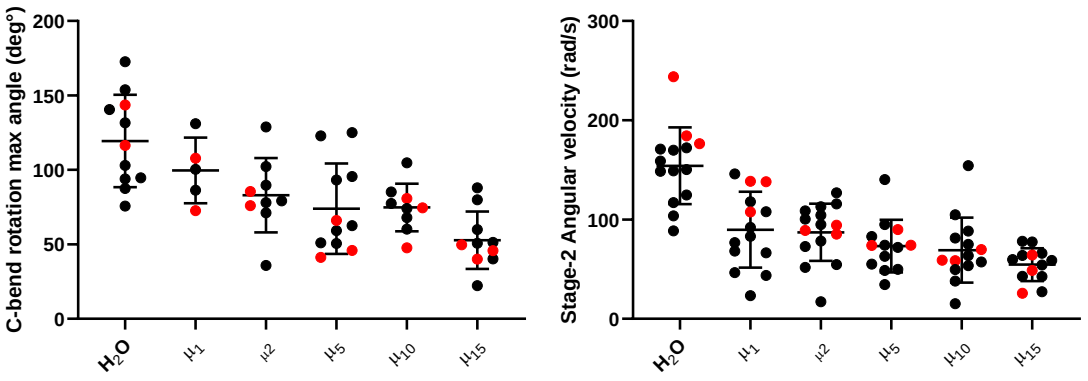
(C) Tail-beat frequency.

(D) Rotation amplitude.

FIGURE 5.4: **Several variable have been computed only in the fast-swimming stage for each experiment and analyzed according to fluid viscosity.** For each viscous condition, we analyzed 11 (μ_w), 9 (μ_1), 11 (μ_2), 12 (μ_5), 14 (μ_{10}), 13 (μ_{15}) experimental videos, for the fast-swimming traveled distance (Fig. 5.4 A) and 14 (μ_w), 14 (μ_1), 15 (μ_2), 13 (μ_5), 14 (μ_{10}), 13 (μ_{15}) escape responses for the translation velocity (Fig. 5.4 B), tail-beat frequency (Fig. 5.4 C) and mean rotation amplitude (Fig. 5.4 D). For the distance, two fast-swimming cycles were required while other quantities can be averaged. Each experimental escape response is represented by an individual point. Note we show in red the experimental cases we further modeled with the numerical simulation. The mean value is represented alongside the SEM. Multiple statistical comparisons can be found in Table 5.1.

As mentioned before, additional variables have been extracted to quantify the C-start deformations, including the maximum rotation angle due to C-start motions, or the angular velocity between C-bend and counter-bend rotations (Fig. 5.5). Note only escape responses where the maximal rotation is attained during the C-bend, have been selected regarding the maximum C-bend rotation (Fig. 5.6 A), in order to analyze the most stereotyped locomotions. Despite the important variability, we can see the maximum rotation is overall decreasing for high-viscous flows, especially when compared against

water (Fig. 5.5 A). Similarly, we can also see a slight angular velocity decrease for viscous flows (Fig. 5.5 B).



(A) Maximal rotation angle.

(B) C-start angular velocity.

FIGURE 5.5: **C-starts have also been analyzed for each experiment, according to fluid viscosity.** For each viscous condition, we analyzed 11 (μ_w), 5 (μ_1), 9 (μ_2), 11 (μ_5), 9 (μ_{10}), 10 (μ_{15}) experimental videos, for the maximal C-bend rotation angle (Fig. 5.5 A) and 14 (μ_w), 14 (μ_1), 15 (μ_2), 13 (μ_5), 14 (μ_{10}), 13 (μ_{15}) escape responses for the C-bend rotation velocity (Fig. 5.5 B). For the maximum rotation, we only considered escape responses where the rotation is maximum during the C-bend motion to filter non-stereotyped behaviors. Each experimental escape response is represented by an individual point. Note we show in red the experimental cases we further modeled with the numerical simulation. The mean value is represented alongside the SEM. Multiple statistical comparisons can be found in Table 5.1.

For each kinematic variable, we performed one-way ANOVA statistical tests combined with multi-comparison tests, to analyze in more detail the differences of the kinematic variables between each viscosity condition (Table 5.1). Translation kinematics represented by distances, displacement, and velocities, has been found significantly impacted by fluid viscosity, excepted between μ_w and μ_1 for the displacement and the distance traveled during fast-swimming, and between μ_{10} and μ_{15} for the displacement and the average fast-swimming velocity. The rotation performance which includes maximal rotation and C-start angular velocity seems to be considerably different between water and viscous flows, especially for the angular velocity of the C-start. In fact, the four other variables are hardly significant between high-viscous flows excepted the tail-beat frequency between specific viscous conditions.

In definitive, we observed zebrafish eleuthero-embryos tended to perform lighter C-bend motion (Fig. 5.5 A) which produces weaker and slower rotation movements during the C-start, in viscous flows (Fig. 5.5 B). Rigid rotation kinematics was consistent with translation kinematics described by velocity and traveled distances which were drastically decayed over the viscosity range (Figs. 5.3 A, 5.3 B, 5.4 A, 5.4 B). As for the tail-beat frequency which may represent the body kinematics, we observed a minor decrease over the viscosity which can result in significant differences, between distant viscosity conditions (Fig. 5.4 C).

TABLE 5.1: A one-way ANOVA and Tukey's test: Evaluation of viscosity on the different kinematic variables.

	Disp.	Total Dist.	Dist.	Velocity
μ_w vs. μ_1	ns	*	ns	*
μ_w vs. μ_2	***	****	****	****
μ_w vs. μ_5	****	****	****	****
μ_w vs. μ_{10}	****	****	****	****
μ_w vs. μ_{15}	****	****	****	****
μ_1 vs. μ_2	****	****	****	****
μ_1 vs. μ_5	****	****	****	****
μ_1 vs. μ_{10}	****	****	****	****
μ_1 vs. μ_{15}	****	****	****	****
μ_2 vs. μ_5	****	****	****	****
μ_2 vs. μ_{10}	****	****	****	****
μ_2 vs. μ_{15}	****	****	****	****
μ_5 vs. μ_{10}	**	****	***	*
μ_5 vs. μ_{15}	****	****	****	****
μ_{10} vs. μ_{15}	ns	*	*	ns (0.0582)

	Tail-beat freq.	Rot. Amp.	CBmax	Ang. Velocity
μ_w vs. μ_1	*	ns	ns	****
μ_w vs. μ_2	***	ns	*	****
μ_w vs. μ_5	****	***	**	****
μ_w vs. μ_{10}	****	****	**	****
μ_w vs. μ_{15}	****	****	****	****
μ_1 vs. μ_2	ns	ns	ns	ns
μ_1 vs. μ_5	ns	***	ns	ns
μ_1 vs. μ_{10}	*	****	ns	ns
μ_1 vs. μ_{15}	****	***	*	ns (0.0511)
μ_2 vs. μ_5	ns	ns	ns	ns
μ_2 vs. μ_{10}	ns	ns (0.0531)	ns	ns
μ_2 vs. μ_{15}	***	ns (0.0776)	ns	ns (0.0787)
μ_5 vs. μ_{10}	ns	ns	ns	ns
μ_5 vs. μ_{15}	**	ns	ns	ns
μ_{10} vs. μ_{15}	ns (0.0572)	ns	ns	ns

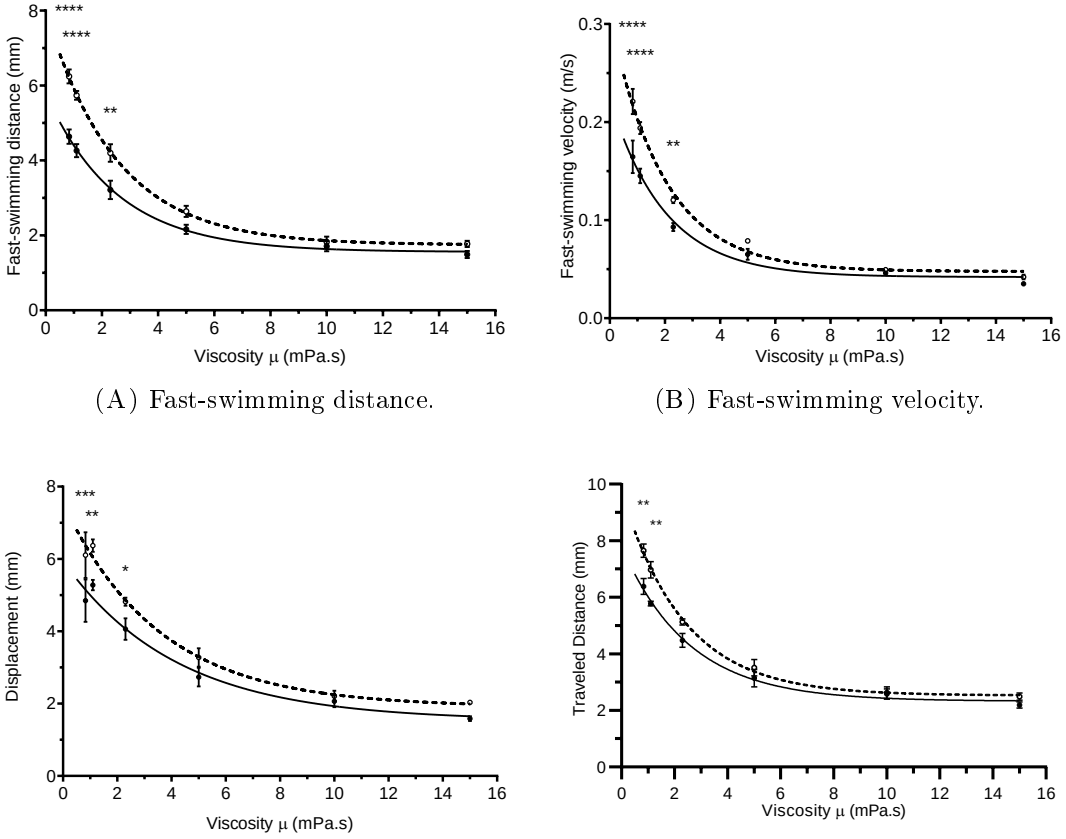
For each kinematic variable, we show the experimental impact of viscosity between each condition (N=10-15) by summarizing the p-value. Non-significant (ns): p $\geq$0 .05, *: p $<$ 0.05, **: p $<$ 0.01, ***: p $<$ 0.001, ****: p $<$ 0.0001.

In what follows, we focused on the translation variables associated to the center of mass, as the body rotation was not as straightforward to compute as the center-of-mass position as it required an intermediate *Procrustes Analysis* step. Moreover, the body rotation was more complex to characterize with simple variables. Furthermore, we observed the four translation variables were much more impacted than rotation by viscous flows and thus were sufficient to analyze the viscous effects. As explained previously, we only conserved the three best videos per viscosity condition for conducting the numerical simulations. Nevertheless, we considered the reduced sample may characterize the whole experiments, as the viscosity impacts seemed to be particularly important and stereotyped.

5.1.3 Impacts of numerical simulations

Afterward, we performed $N = 3$ numerical simulations per viscosity condition, associated with the subset of experimental data previously described. First, we compared the resulting kinematics against the experiments (Fig. 5.6). Unlike previous Figs. 5.3, 5.4, notice we represented all results according to viscosity to assess precisely the correlation with the fluid viscosity. Overall, we can see an identical general trend

according to fluid viscosity for all kinematic variables: a rapid exponential decay which tends to attain a plateau for viscosity values higher than μ_5 (Fig. 5.6). As for the numerical comparison, the general shape of kinematics is generally well-reproduced in numerical simulation excepted in μ_w, μ_1, and μ_2 fluids where all kinematic quantities from numerical simulations, are slightly overestimated. This comparison can be directly related to the comparison of the center-of-mass trajectories (Fig. 4.25) where most of the numerical simulations were found very satisfying. Only a few simulations seemed to accelerate in the fast-swimming phase with, no specific correlation to the medium viscosity, as analyzed in Chapter 4.



(A) Fast-swimming distance.

(B) Fast-swimming velocity.

(C) Displacement.

(D) Traveled distance.

FIGURE 5.6: **Comparison of numerical simulation kinematics against experiments.** We represented in parallel both the experimental (filled circles) and the numerical (empty circles) kinematics including the fast-swimming traveled distance (Fig. 5.6 A), the fast-swimming average velocity (Fig. 5.6 B), the raw displacement (Fig. 5.6 C) and the global traveled distance (Fig. 5.6 D). Note for each condition, we show the mean of $N = 3$ escape responses alongside the SEM. In addition, we computed the non-linear correlations (exponential decay) for the experimental (solid line) and numerical (dashed line) values. Finally, we annotated the results of a Sidak's test when numerical values deviate significantly from experiments. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

Therefore, the impacts of numerical errors and overall viscosity effects have been analyzed with a two-way ANOVA test (Table 5.2). As a result, the impact of the numerical error can be neglected given that viscosity affected kinematic variations by 90% whereas

TABLE 5.2: A two-way ANOVA test: Evaluation of kinematic impacts of numerical simulations and fluid viscosity.

Source of Variation	Traveled distance % TV		Velocity % TV		Displacement % TV		Distance % TV	
Viscosity μ	92.84	****	90.16	****	87.77	****	88.09	****
From exp. to num. values	2.905	****	4.467	****	4.472	****	6.609	****
Interaction	1.481	*	2.702	***	1.301	*	3.205	***

Summary of the two-way ANOVA test regarding numerical modeling and viscous impacts based on N=18 experimental and numerical data, including % of total variation (TV). Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

simulations had an impact up to 6.6%. In definitive, numerical simulations do not reproduce exactly experimental kinematics in low-viscous flows but the biomechanical response was found highly sensitive to viscosity change. As viscosity effects were largely predominant in kinematic variables compared to numerical errors, the numerical solutions have then been studied regardless of the numerical error, although important impacts were found on low-viscous flows.

Afterward, multi-comparison statistical tests have been conducted to analyze the effects of viscous flows on numerical solutions (Table 5.3). As a result, we found numerical simulations reproduce well enough the global biomechanical effects due to viscosity change. Most of the multiple comparisons raised an identical trend between experimental and numerical kinematics. Note experimental statistics differ from Table 5.1 as we strongly reduced the data-set from the original experiment. This result was expected as we demonstrated viscosity change was largely prevalent compared to simulation influence. Nevertheless, note the numerical simulation overestimated the discrepancy for a few comparisons: two close comparisons for the displacement, and fast-swimming errors between high viscous flows. Thus, it is important to note those specific comparisons should be analyzed with special attention. In fact, fast-swimming variables such as velocity or traveled distance may not be well-suited to analyze numerical simulations as they do not assess accurately the actual effects of viscosity, which is consistent with previous analyses in Chapter 4. Indeed, we observed the most important numerical errors occur most of the time during the fast-swimming stage. That is why, global variables should be preferred to analyze biological effects. In conclusion, nearly all comparisons between the three highest fluid viscosity were non-significant for all variables while water and low-viscosity fluids differed significantly from highly-viscous flows.

As for rotation motions, we did not compare specific quantities as explained previously. However, we noticed rotation peaks are not synchronized with bend amplitude extrema (see Figs. 5.2, 5.7). In fact, a constant delay between the occurrence of each peak was observed. In consequence, the time delay *Tdef-rot* between body deformations and body rotation was determined by averaging the six time differences. Therefore, the time delay was quantified for each escape response to precise the relationship with viscosity (Fig. 5.8). As a result, the delay observed in water was not due to numerical approximations

TABLE 5.3: A one-way ANOVA and multiple comparison Tukey's test: Evaluation of experimental and numerical impacts across the different viscosity conditions.

	Traveled distance		Displacement		Fast-swimming distance		Fast-swimming velocity	
Tukey's multiple comparisons test	exp	num	exp	num	exp	num	exp	num
μ_w vs. μ_1	ns	ns	ns	ns	ns	ns	ns	ns
μ_w vs. μ_2	***	****	ns	ns	***	****	***	****
μ_w vs. μ_5	****	****	**	***	****	****	****	****
μ_w vs. μ_{10}	****	****	***	****	****	****	****	****
μ_w vs. μ_{15}	****	****	***	****	****	****	****	****
μ_1 vs. μ_2	*	***	ns (0.0931)	* (0.0287)	**	***	**	****
μ_1 vs. μ_5	****	****	***	****	****	****	***	****
μ_1 vs. μ_{10}	****	****	****	****	****	****	****	****
μ_1 vs. μ_{15}	****	****	****	****	****	****	****	****
μ_2 vs. μ_5	**	**	ns (0.0601)	* (0.0297)	**	***	ns (0.2102)	** (0.0046)
μ_2 vs. μ_{10}	***	****	**	***	***	****	*	****
μ_2 vs. μ_{15}	****	****	**	***	***	****	**	****
μ_5 vs. μ_{10}	ns	ns	ns	ns	ns (0.4539)	* (0.0326)	ns (0.5596)	* (0.048)
μ_5 vs. μ_{15}	ns	ns	ns	ns	ns (0.1216)	* (0.0214)	ns (0.1557)	* (0.0116)
μ_{10} vs. μ_{15}	ns	ns	ns	ns	ns	ns	ns	ns

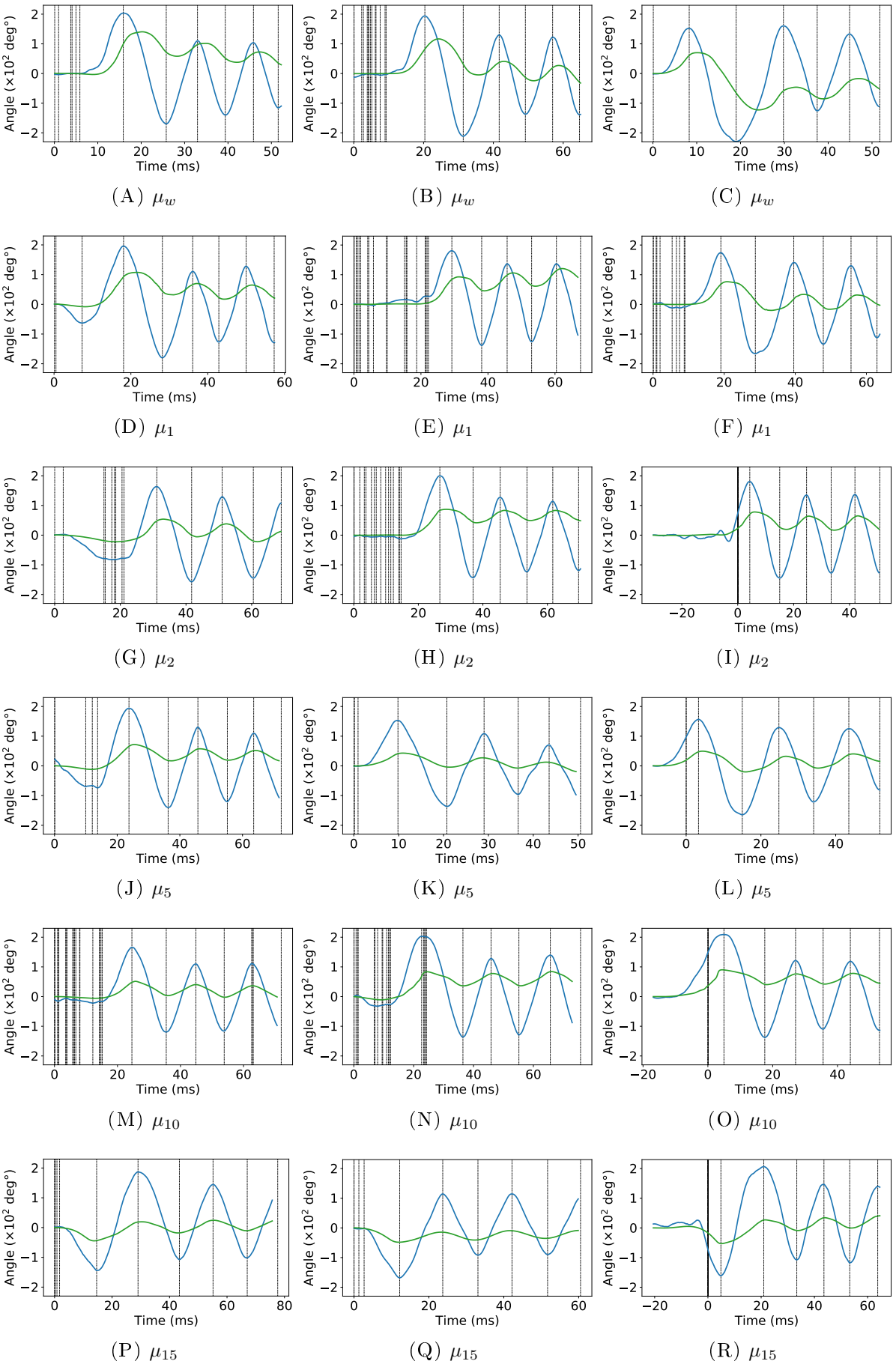
For each kinematic variable, we show the experimental and numerical impacts between each viscosity condition (N=3) by summarizing the p-value. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

and in addition, both the deformation and the rotation tended to be synchronized in the highest viscous flows.

In fact, the observed time delay between body deformations and the rigid rotation can be seen as a result of the inertia motion which is particularly present in high Reynolds flows. *Re* (see Eq. 2.6) was computed by using the average velocity of the entire escape simulation (Fig. 5.9). As a result, the general shape described previously (Figs. 5.6, 5.8) can be highlighted *via Re* which describes the flow regime, as explained in Chapter 2.

Above all, highly inertial flow regimes generate an important inertia motion represented by the delay between the body kinematics and the corresponding fluid motions *i.e.* the rigid body rotation. Instead, high-viscosity flows enable a direct impact of body deformations by producing almost instantly the associated body rotation. As with distance and velocity, the time delays in rotation and *Re* were analyzed statistically between all viscosity values (Table 5.4). As a result, variations as similar as translation kinematics can be observed (Table 5.3). In definitive, the whole kinematics represented by traveled distances, fast-swimming velocity, displacement, and *Tdef-rot* is well-consistent with the Reynolds shape.

Furthermore, the visualization of the wake generated by the zebrafish escape response was also very consistent with *Re* (see Fig. 5.10). Indeed, the produced vortex shedding phenomenon decreased dramatically according to fluid viscosity. These observations are mainly due to the presence of highly viscous forces which inhibit the generation of further vortices. Indeed, vortices are produced by the fluid flow moved during each tail-beat and body deformations but cannot remain in the fluid. That is why, a rapid dissipation of the wake can be observed in high viscous flows, especially for fluid viscosity higher than μ_5.



(A) μ_w (B) μ_w (C) μ_w

(D) μ_1 (E) μ_1 (F) μ_1

(G) μ_2 (H) μ_2 (I) μ_2

(J) μ_5 (K) μ_5 (L) μ_5

(M) μ_{10} (N) μ_{10} (O) μ_{10}

(P) μ_{15} (Q) μ_{15} (R) μ_{15}

FIGURE 5.7: **Illustration of body deformation effects on the rigid rotation, for all test cases.** All Figs. 5.7A-5.7R correspond to an individual escape response, each row being associated to a viscous condition. For all figures, we represented in blue the bend amplitude alongside the body rotation in green.

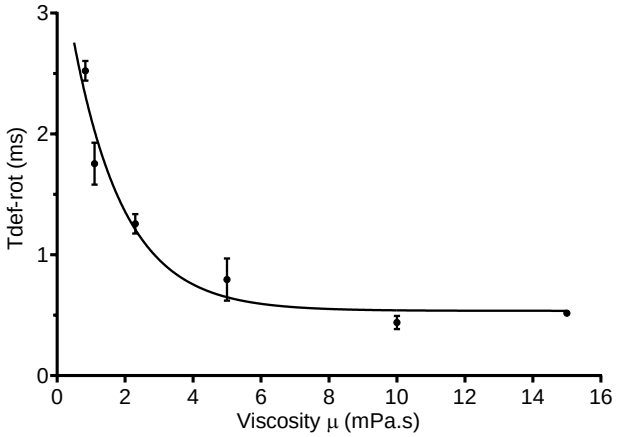


FIGURE 5.8: **Average time delay between body deformations and rotation according to viscosity.** We computed the quantity *Tdef-rot* to characterize the time delay between the bend amplitude and the rotation angle, visible in Fig. 5.7. We represented the mean (points) with the SEM and the non-linear regression which decays at an exponential rate. Multiple statistical comparisons can be found in Table 5.4.

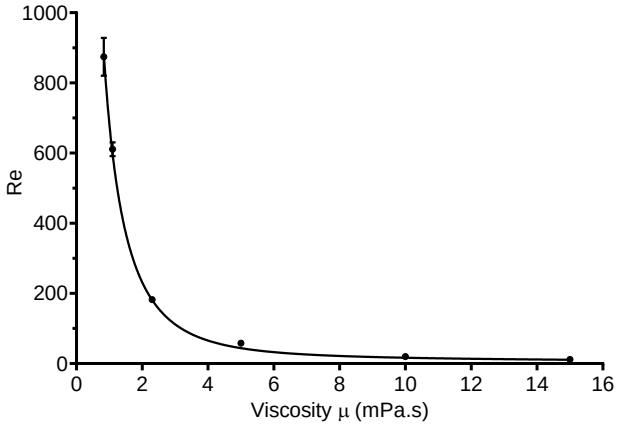
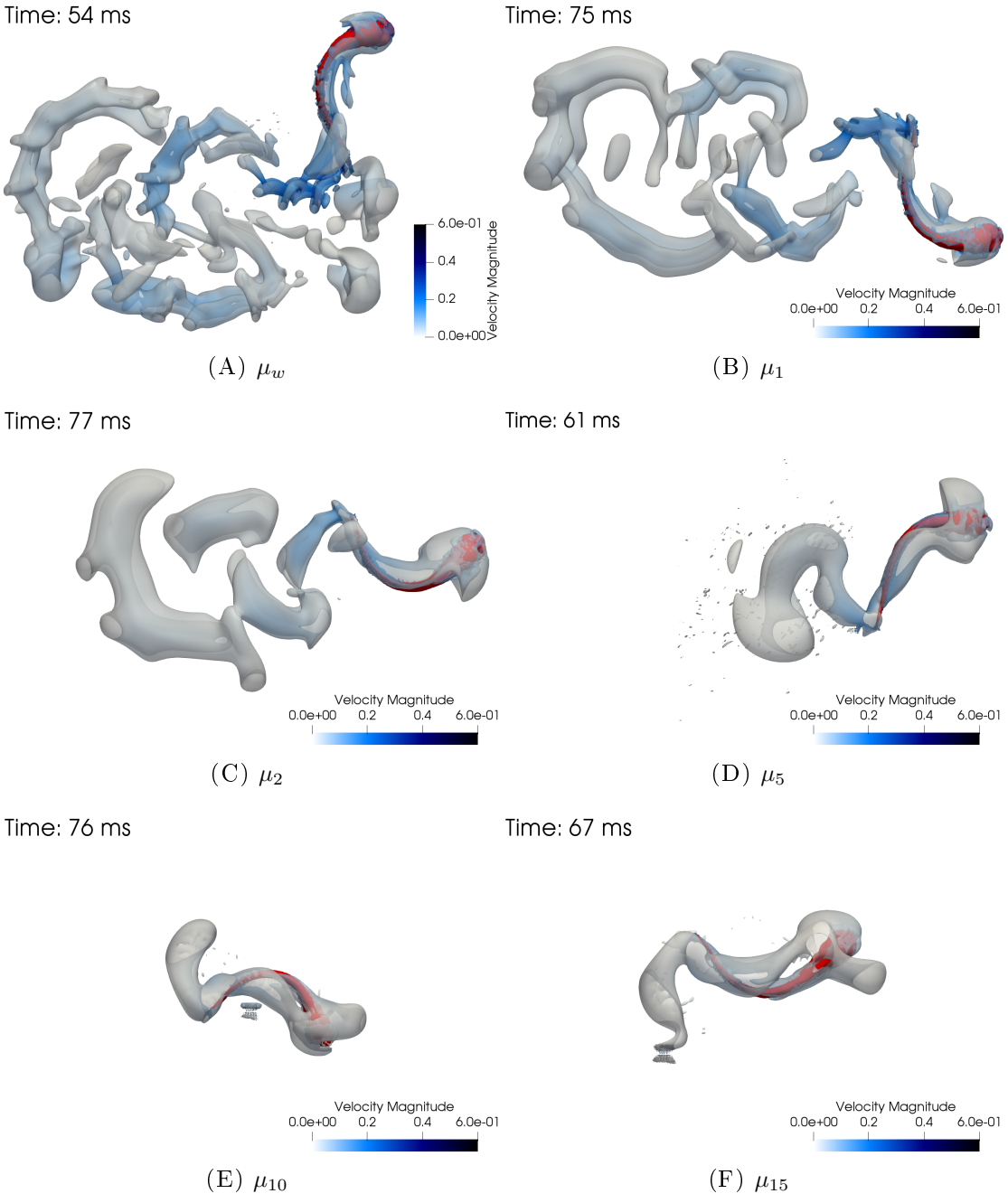


FIGURE 5.9: **Flow regime: variation of Re according to the fluid viscosity.** Note the Reynolds number has been computed with the average velocity of the complete escape simulations. We represented the mean (points) with the SEM and the corresponding non-linear regression (solid line). The regression decays as $\frac{\exp(-\mu)}{\mu}$. Multiple statistical comparisons Tukey's test can be found in Table 5.4.

On the whole, we highlighted how all experiment-driven numerical simulations have well-reproduced the physical mechanisms caused by high-viscous flows. In particular, simple and accessible kinematic variables have been compared against experimental measurements such as traveled distances or velocity magnitude. However, kinematic variables were found not suitable to assess and quantify highly viscous flows, especially in the case of a tiny subset of experimental data with only three escape responses per viscosity condition. Obviously, the integration of more test cases would certainly provide finer results. In addition, kinematic variables need to be analyzed as a whole to provide full insights about the swimming performance of the escape response which is complex due to the difficulty to characterize rotation motions. Similarly, body deformations could



(A) μ_w (B) μ_1

(C) μ_2 (D) μ_5

(E) μ_{10} (F) μ_{15}

FIGURE 5.10: **Illustration of various advanced wake according to fluid viscosity.** We represented an example of a top-view of three-dimensional numerical simulation for each viscosity μ_w (Fig. 5.10 A), μ_1 (Fig. 5.10 B), μ_2 (Fig. 5.10 C), μ_5 (Fig. 5.10 D), μ_{10} (Fig. 5.10 E), μ_{15} (Fig. 5.10 F). Notice the zebrafish body in red and the fluid vorticity represented by an iso-contour of the Q-criterion (Q=150), colored by velocity magnitude. Note the water test case correspond to the initial experimental locomotion considered in Chapter 4.

TABLE 5.4: A one-way ANOVA and multiple comparison Tukey's test: Evaluation of both *Re* and *Tdef-rot* variations across the different viscosity conditions.

	Re	Tdef-rot
μ_w vs. μ_1	****	**
μ_w vs. μ_2	****	****
μ_w vs. μ_5	****	****
μ_w vs. μ_{10}	****	****
μ_w vs. μ_{15}	****	****
μ_1 vs. μ_2	****	ns (0.0752)
μ_1 vs. μ_5	****	***
μ_1 vs. μ_{10}	****	****
μ_1 vs. μ_{15}	****	****
μ_2 vs. μ_5	*	ns
μ_2 vs. μ_{10}	**	**
μ_2 vs. μ_{15}	**	**
μ_5 vs. μ_{10}	ns	ns
μ_5 vs. μ_{15}	ns	ns
μ_{10} vs. μ_{15}	ns	ns

For both *Re* and *Tdef-rot*, we show the variations between each viscosity condition (N=3) by summarizing the p-value. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

be directly analyzed and compared to predict the overall swimming performances but are also not straightforward to study due to the complexity of zebrafish escape behavior. Therefore, we studied the swimming energetic performances to enhance the swimming performance understanding. Swimming kinematics is caused by body deformations and body kinematics is generated by the underlying biomechanics related to zebrafish muscles, locomotor system, and its energetic capacity. That is why, we assumed the study of zebrafish energetic performances could provide finer information about the biological processes involved in locomotion, especially in the case of EFP stimulation which is assumed to maximize the escape response capacity. Thus, zebrafish escape energetic performances in highly viscous flows are recapitulated in what follows.

5.1.4 Energetic performances

First of all, numerical simulation enabled to compute accurately the power consumption across the escape swimming for all body kinematics (Fig. 5.11). In particular, we can see the power shape is composed of multiple peaks such that one single power spike is occurring during each tail-beat. Moreover, power minimums seem to occur for each bend amplitude change (denoted by vertical dashed lines). As a result, the power output seems to be well-synchronized with the body deformations, unlike the body rotation. The zebrafish propulsion is *a priori* mainly generated by the tail-beats but a proper analysis of the tail tip velocity did not show a signal as synchronized as with the bend amplitude. Thus, we can assume the zebrafish power output is essentially based on the body kinematics timing. At first, the previous assertion seems reasonable as body deformations are related to muscle activity and we assume the muscle activity can be estimated through the hydrodynamic power we compute *i.e.* the power required by the

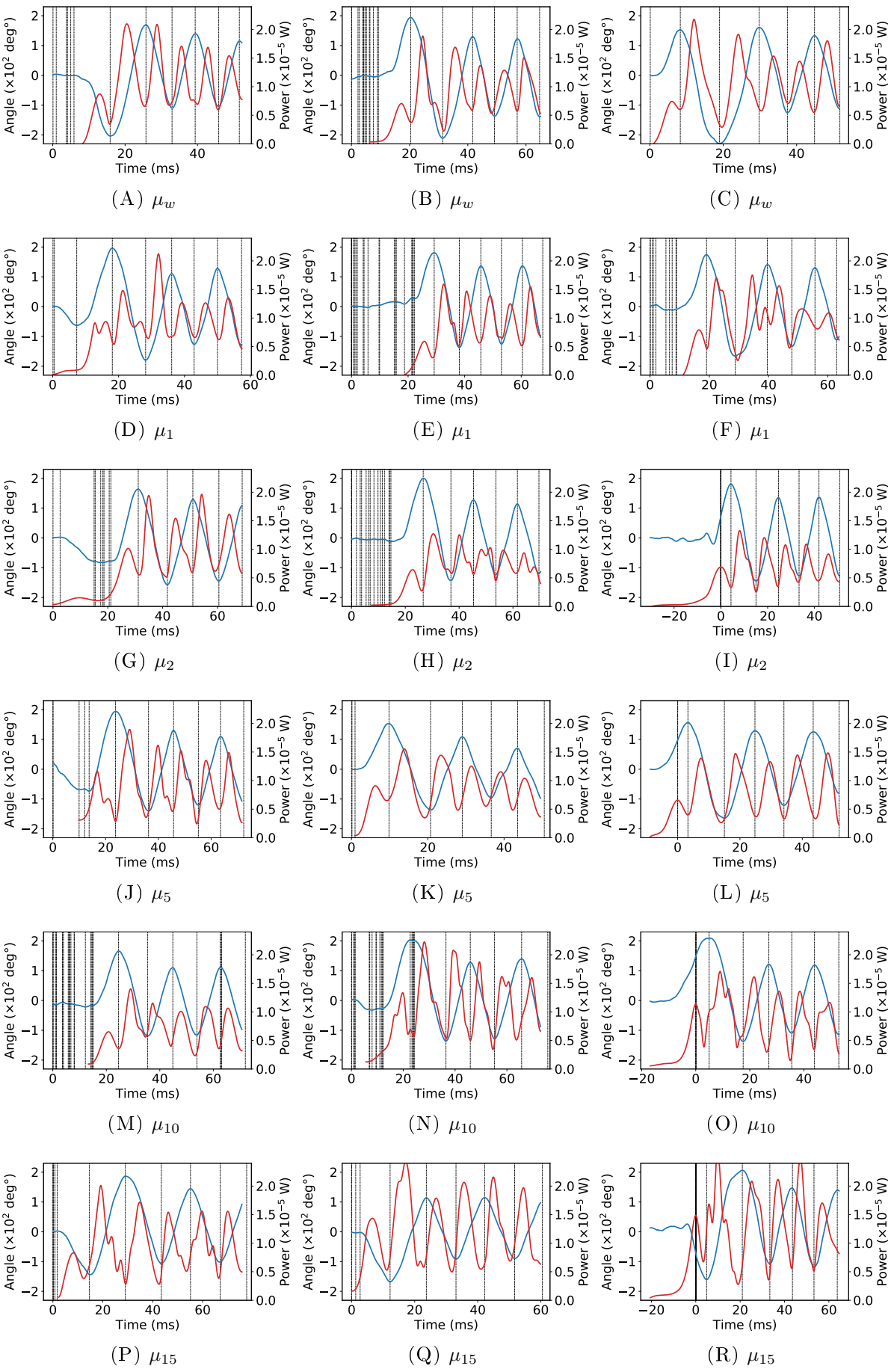


FIGURE 5.11: **Illustration of body deformation effects on the power output, for all test cases.** All Figs. 5.11 A-5.11 R correspond to an individual escape response, each row being associated to a viscous condition. For all figures, we represented in blue the bend amplitude alongside the power in red.

zebrafish eleuthero-embryo body to compensate for pressure and viscous fluid forces (see Chapter 2). Therefore, we used the escape responses challenged in high-viscosity fluids to test our observations. To that end, we calculated the average time delay *Tdef-pow* (Fig. 5.12) between power minimums and bend amplitude peaks similarly to *Tdef-rot* (Fig. 5.8). Although *Tdef-pow* was correlated to fluid viscosity (Fig. 5.12), all time delays were found inferior to 1 ms except for the highest viscous flow, slightly above the average and the 1 ms-threshold. Indeed, we considered 1 ms as a threshold value given the numerical noise present in the power signal (see Fig. 4.12), due to a large amount of CFD time steps. In definitive, we consistently observed the same synchronization for each experiment, and no considerable changes have been found according to fluid viscosity, excepted minor delays for μ_{15}-simulations, which were effectively significant (see Table 5.5). However, the μ_{15} result may be due to the particular high amount of CFD iterations observed for highly viscous flows, which increased the data noise.

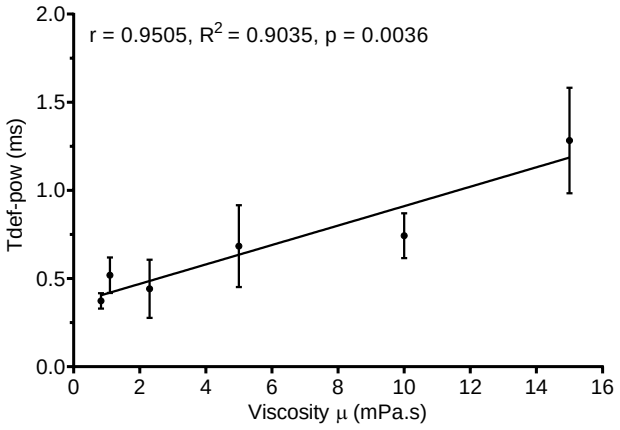


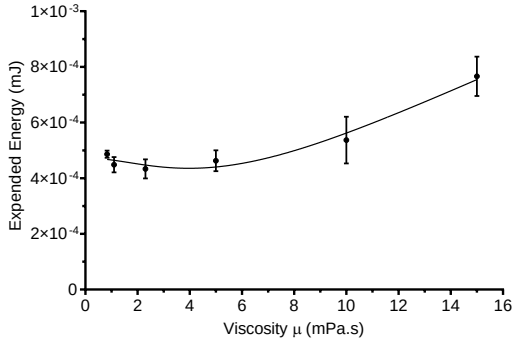
FIGURE 5.12: **Average delay between body deformation maximum and power minimum according to viscosity.** We computed the quantity *Tdef-pow* to characterize the time delay between the bend amplitude and the power cycles, visible in Fig. 4.13 for example. Data are mean ± SEM. The slope of the linear regression is significantly different from zero (p=0.0007). A Pearson correlation test showed a significant correlation between *Tdef-pow* and μ. Multiple statistical comparisons can be found in Table 5.5.

As explained in the previous Chapter 4, we defined three main quantities to analyze the escape power for each zebrafish escape locomotion: P_{mean}, E_t and CoT. Thus, we studied the swimming energetic performances through these three variables (Fig. 5.13). First, we can see the expended energy is almost identical regardless of the fluid viscosity (Fig. 5.13 A). Indeed, the global energy shape seems to correspond to a slight and low variation in low and intermediate viscosity flows before increasing in μ_{15}-flows; the non-linear regression being in the form of $\mu \exp(\mu)$. That is why, significant differences are found when comparing viscous flows against μ_{15} (Table 5.6). Unlike the expended energy, P_{mean} seems to remain around a fixed value and no correlation with viscosity is visible (Fig. 5.13 B). Those results suggest in particular zebrafish eleuthero-embryo cannot consume more than a fixed amount of power, regardless of the fluid viscosity, to produce maximal C-start escape responses. As for the CoT (Fig. 5.13 C), we emphasize the

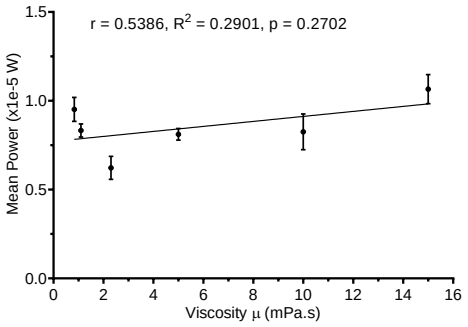
TABLE 5.5: A one-way ANOVA and multiple comparison Tukey's test: Evaluation of *Tdef-pow* variations across the different viscosity conditions.

	Tdef-pow
μ_w vs. μ_1	ns
μ_w vs. μ_2	ns
μ_w vs. μ_5	ns
μ_w vs. μ_{10}	ns
μ_w vs. μ_{15}	*
μ_1 vs. μ_2	ns
μ_1 vs. μ_5	ns
μ_1 vs. μ_{10}	ns
μ_1 vs. μ_{15}	ns (0.0958)
μ_2 vs. μ_5	ns
μ_2 vs. μ_{10}	ns
μ_2 vs. μ_{15}	ns (0.0586)
μ_5 vs. μ_{10}	ns
μ_5 vs. μ_{15}	ns
μ_{10} vs. μ_{15}	ns

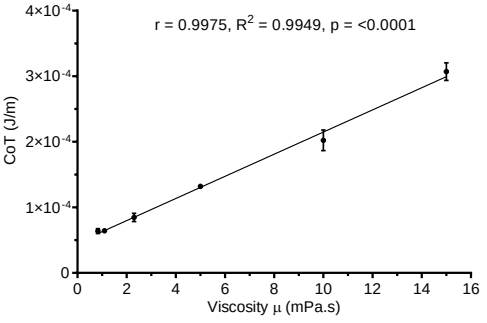
We show the variations of *Tdef-pow* between each viscosity condition (N=3) by summarizing the p-value. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.



(A) Expended energy E_t.



(B) Mean power P_{mean}.



(C) CoT.

FIGURE 5.13: **Representation of escape swimming energetic performances according to viscosity.** For each escape response, we computed E_t (Fig. 5.13 A), tP_{mean} (Fig. 5.13 B) and CoT (Fig. 5.13 C). Note for each condition, we show the mean of $N = 3$ locomotion alongside the SEM. The slope of the linear regression of P_{mean} (Fig. 5.13 B) was non-significantly different from zero (p=0.0684) while the slope of the linear regression of CoT (Fig. 5.13 C) was significantly different from zero (p<0.0001). Pearson correlation tests showed a significant correlation between CoT and μ (Fig. 5.13 C) and a non-significant correlation between P_{mean} and μ (Fig. 5.13 B). Multiple comparisons Tukey's test can be found in Table 5.6.

TABLE 5.6: A one-way ANOVA and multiple comparison Tukey's test: Evaluation of energetic performances between the different viscosity conditions.

	E_t	P_{mean}	CoT
μ_w vs. μ_1	ns	ns	ns
μ_w vs. μ_2	ns	*	ns
μ_w vs. μ_5	ns	ns	**
μ_w vs. μ_{10}	ns	ns	****
μ_w vs. μ_{15}	*	ns	****
μ_1 vs. μ_2	ns	ns	ns
μ_1 vs. μ_5	ns	ns	**
μ_1 vs. μ_{10}	ns	ns	****
μ_1 vs. μ_{15}	**	ns	****
μ_2 vs. μ_5	ns	ns	*
μ_2 vs. μ_{10}	ns	ns	****
μ_2 vs. μ_{15}	**	**	****
μ_5 vs. μ_{10}	ns	ns	**
μ_5 vs. μ_{15}	*	ns	****
μ_{10} vs. μ_{15}	ns (0.0672)	ns	****

For all energetic variables, we show the variations between each viscosity condition (N=3) by summarizing the p-value. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

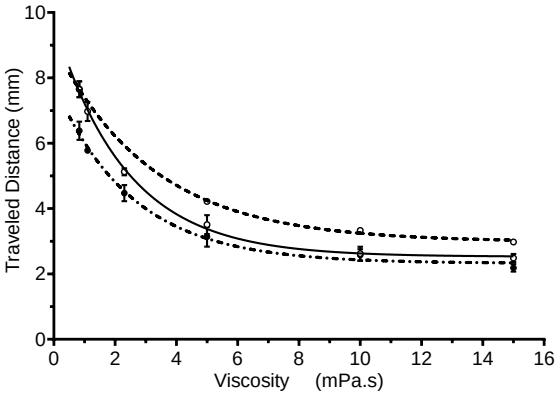
particularly strong linear correlation according to fluid viscosity. Although viscous forces are proportional to dynamic fluid viscosity μ, this result was not *a priori* straightforward as the hydrodynamic power is formed by a pressure term in addition to the viscous force term. Nevertheless, viscous forces might be predominant in the CoT given the presence of highly viscous flows, which might explain the linear correlation of the CoT with respect to viscosity. As previously, we highlighted the energetic variations by performing multi-comparisons ANOVA statistical tests (Table 5.6). In particular, the CoT enlightened high-viscosity differences in energetic behaviors. For instance, the CoT measured significant differences between μ_5, μ_{10}, μ_{15} whereas kinematic variables such as distance and velocity measured non-significant differences. Indeed, the CoT provides finer information than swimming kinematics in the highest viscosity flows, due to its linear shape. Thus, we can see highly significant differences between high-viscosity flow solutions. In definitive, energetic performances complete the kinematic analysis and can provide supplementary insights about the underlying biomechanics which govern zebrafish eleuthero-embryo locomotion.

5.1.5 Impacts of body deformations: a fictitious approach

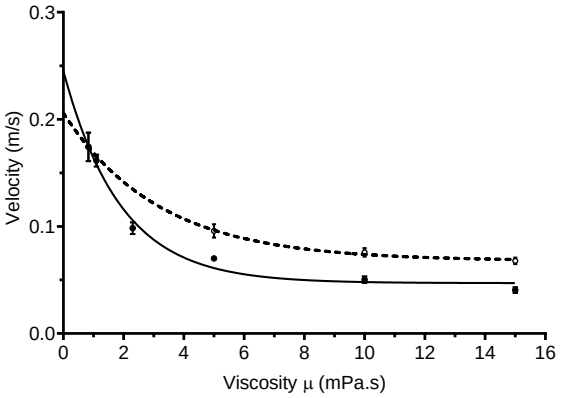
To obtain more information about the power involved in fish-like swimming in nature, we used our numerical model as a fictitious approach to study viscosity change. To be more precise, we separated the modification of body kinematics from the fluid viscosity change by simulating the swimming performance across the viscosity range, from a fixed body deformation. Previously, we briefly described how body kinematics is modified across fluid viscosity, by using the tail-beat frequency or angular measurements. In particular, we observed a slight decrease in tail-beat frequency according to viscosity. As

detailed previously, swimming energetic performances provided additional information which neither swimming kinematics nor body kinematics could provide. However, performance differences observed through energy consumption or CoT are caused both by fluid viscosity and the resulting modifications in body kinematics. That is why, we investigated the impact of body deformations on the escape swimming performances. As the considered body deformations may not correspond to the actual deformations, we performed fictitious numerical simulations by modifying the fluid dynamic viscosity (Fig. 5.14).

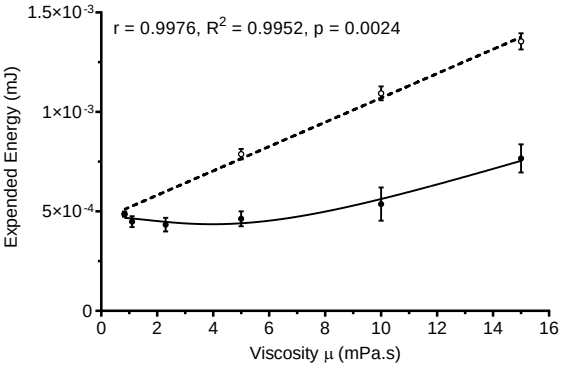
As a result, the actual zebrafish body reaches similar traveled distances or escape speeds at an inferior energetic cost. Indeed, body deformations observed experimentally in water (def-μ_w) would have shown poor swimming performances by expending a lot of more power (Figs. 5.14 C-5.14 E) for identical kinematics (see Figs. 5.14 A, 5.14 B). Thus, the zebrafish eleuthero-embryo actually alters its body kinematics to adapt to a different environment. This result would suggest μ_w-deformations are hardly adequate in viscous flows. Furthermore, the zebrafish eleuthero-embryo could actually minimize its CoT (Fig. 5.14 E) to escape with a maximal performance in terms of swimming kinematics (Figs. 5.14 A, 5.14 B). Nevertheless, the power output required by the eleuthero-embryo to escape with μ_w-body deformations in viscous flows would be higher than the actual amount of energy expended by the eleuthero-embryo, which is certainly limited to the observed value (Figs. 5.14 C, 5.14 D). Thus, these results suggest the zebrafish eleuthero-embryo alters its body deformations to expend all the available power with maximal efficiency. To assess this assertion, the frequency of μ_w-body kinematics was also modified to compare the resulting swimming performances with the actual body deformations in viscous media. Indeed, the swimming performances of modified μ_w-deformations (def-H_2O^{freq}) immersed in viscous flows and actual body deformations used in viscous flows were compared in order to analyze the effects of fictitious body kinematics (see Fig. 5.15). In particular, the μ_w-body deformations (def-H_2O) were altered such that the same amount of energy is expended, relatively to actual experiments. To that end, the acquisition time has been increased by 30%. As a result, we first observed we effectively obtained an identical amount of E_t by a tail-beat frequency modification of 30% (see Fig. 5.15 C). The results also suggest the traveled distance may be correlated to expended energy as no significant change was observed between actual and fictitious traveled distances (Fig. 5.15 A). Besides, it seemed P_{mean} was slightly decreased (Fig. 5.15 D) while the swimming velocity was significantly reduced (Fig. 5.15 B). On the one hand, the amount of consumed power on average seems to be correlated to the escape velocity output. Note those trends can partly be explained by the relationship between body kinematics and power output. Indeed, power maxima were hardly impacted by frequency modification. Thus, an important decrease in frequency was required to strongly decrease the energetic output. That is why, P_{mean} was impacted alongside the velocity. On the other hand, we highlight the fictitious body deformations have been found less efficient than actual experiments as the resulting motion was significantly slower (Fig. 5.15 B)



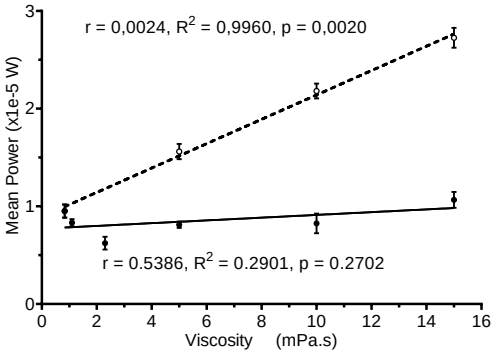
(A) Traveled distance.



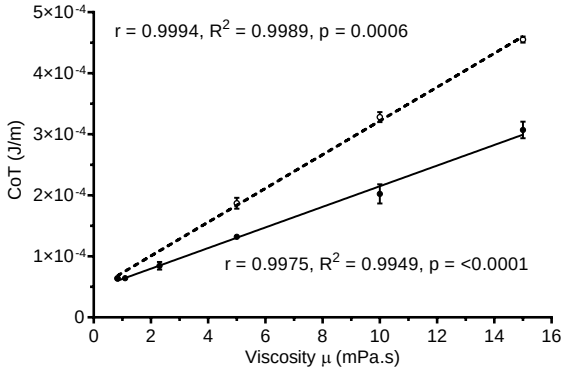
(B) Velocity.



(C) Expended energy E_t.

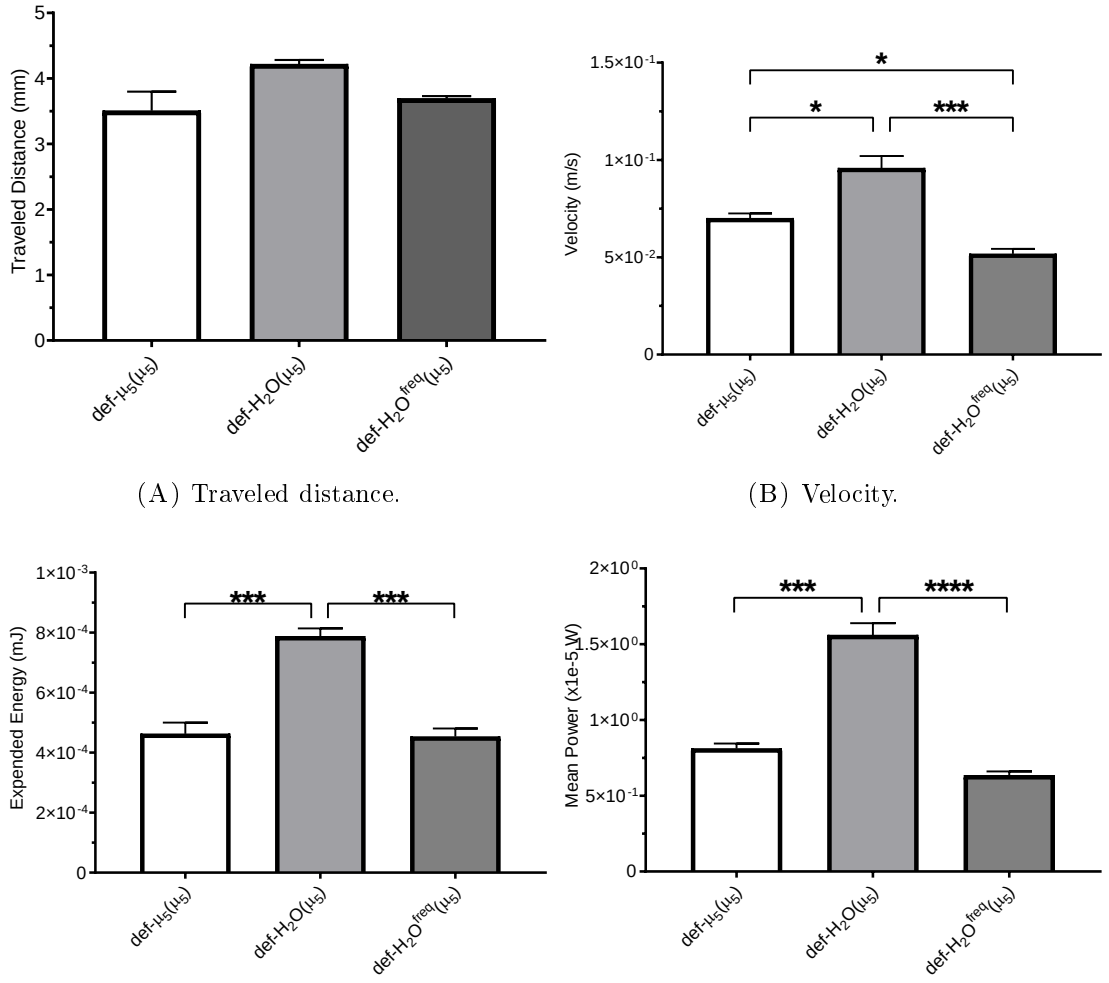


(D) Mean power P_{mean}.



(E) CoT.

FIGURE 5.14: **Comparison of escape swimming performances according to viscosity for fictitious and actual body deformations.** The previously described swimming performances based on experimental body kinematics (solid lines) are represented in parallel to fictitious simulations (dashed lines). Eventual experimental kinematics are represented in dash-dotted lines. The so-called fictitious simulations denote viscous simulations based on body deformations observed in water (μ_w). For each escape response, we computed the total traveled distance (Fig. 5.14 A), the total average velocity (Fig. 5.14 B), E_t (Fig. 5.14 C), P_{mean} (Fig. 5.14 D) and CoT (Fig. 5.14 E). Note for each condition, we show the mean of $N = 3$ locomotion alongside the SEM. The slopes of the linear regressions of fictitious P_{mean} (Fig. 5.13 B) and fictitious CoT (Fig. 5.13 C) were significantly different from zero ($p<0.0001$). The corresponding Pearson correlation tests showed significant correlations.



(A) Traveled distance.

(B) Velocity.

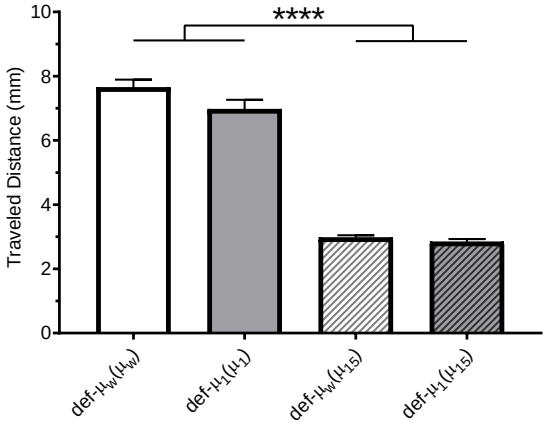
(C) Expended energy.

(D) Mean power.

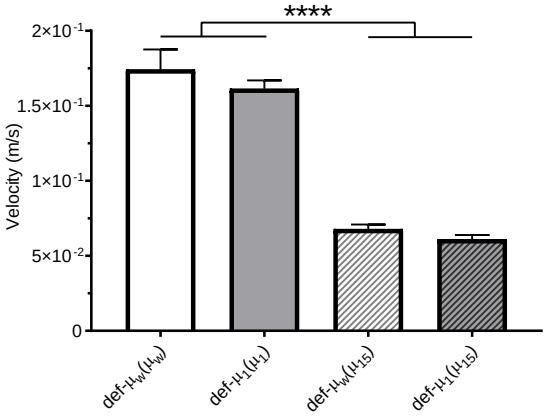
FIGURE 5.15: **Comparison of actual swimming performances in μ_5 against fictitious numerical simulations where the μ_w-deformations have been altered.** To be more precise, we decreased the tail-beat frequency of 30% to expend the same amount of energy than μ_5-body kinematics. In particular, we mainly compared the kinematics of μ_w (def-H_2O), μ_5 (def-μ_5) deformations, and altered μ_w-deformations (def-μ_w^{freq}). The results of fictitious simulations in μ_5-fluid are described by traveled distance (Fig. 5.15 A), average velocity (Fig. 5.15 B), and the swimming energetics represented by E_t (Fig. 5.15 C) and P_{mean} (Fig. 5.15 D). Note for each condition, we show the mean of $N = 3$ locomotion alongside the SEM. Statistics from a one-way ANOVA and Tukey's test. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

while expending the same amount of energy (Fig. 5.15 C). On the whole, we illustrated zebrafish eleuthero-embryos adapt its body kinematics to the surrounding fluid in order to gain efficiency with fixed energy expenditure.

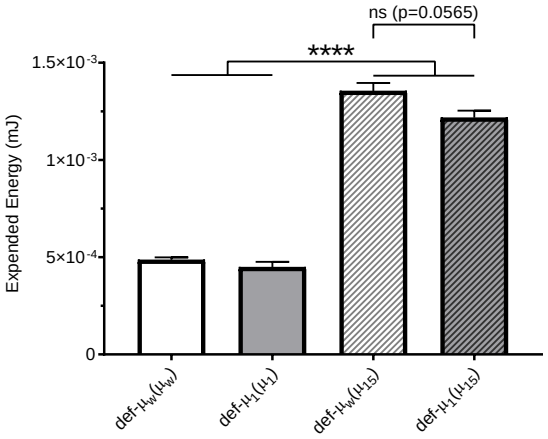
Finally, we used both the experimental data and numerical modeling to initiate the implementation of an effort test to be used by biologists. Let consider altered escape responses which are hardly distinguishable by analyzing either strictly body deformations or rigid kinematics. Moreover, energetic variables might not be fine enough to provide significant differences between the two considered escape motions. The effort test



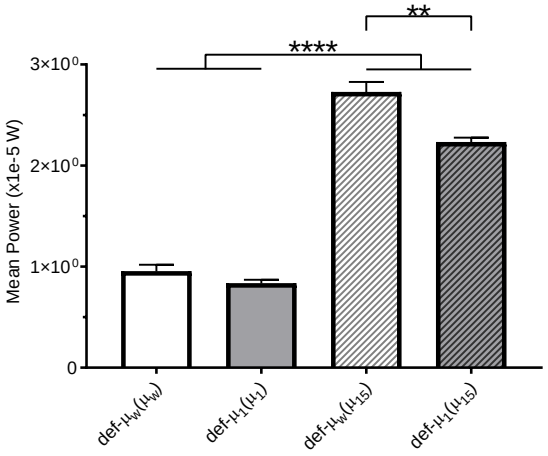
(A) Traveled distance.



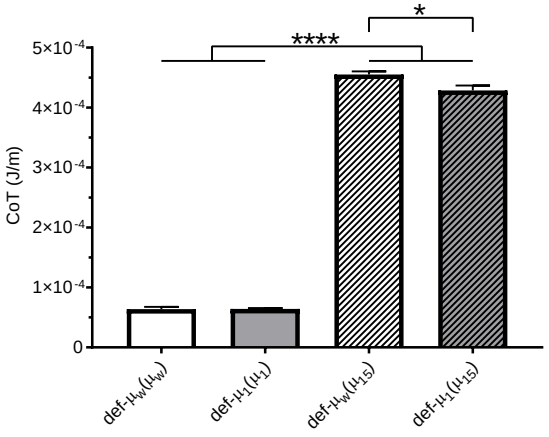
(B) Velocity.



(C) Expended energy E_t.



(D) Mean power P_{mean}.



(E) CoT.

FIGURE 5.16: **A first effort test: comparison of escape swimming performances for two escape conditions against fictitious simulations in high viscous flows (μ_{15}).** For all considered variables (total traveled distance (Fig. 5.16 A), total average velocity (Fig. 5.16 B), E_t (Fig. 5.16 C), P_{mean} (Fig. 5.16 D) and CoT (Fig. 5.16 E)), we compared two experimental escape conditions (def-$\mu_w(\mu_w)$ and def-$\mu_1(\mu_1)$) against their fictitious counterparts in high viscosity fluids (def-$\mu_w(\mu_{15})$% and def-$\mu_1(\mu_{15})$). Note for each condition, we show the mean of $N = 3$ locomotion alongside the SEM. Statistics from a one-way ANOVA and Tukey's test. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

consists of increasing the fluid viscosity to amplify tiny differences in body kinematics and quantify the resulting escape swimming performances with better accuracy. To demonstrate such capabilities, we focused on μ_w and μ_1 escape deformations (def-μ_w and def-μ_1, respectively) which we immersed into μ_{15} fluid viscosity (see Fig. 5.16). Note experimental body kinematics in μ_x-fluid are denoted def-μ_x. No significant differences have originally been found between μ_w (def-$\mu_w(\mu_w)$) and μ_1 (def-$\mu_1(\mu_1)$) neither in swimming kinematics (Figs. 5.16 A, 5.16 B) nor energetic performances (Figs. 5.16 C, 5.16 D, 5.16 E). Thus, we simulated the corresponding escape swimming responses within high-viscosity flows (μ_{15}-conditions). As a result, we found significant differences between the considered escape kinematics (def-$\mu_w(\mu_{15})$) and (def-$\mu_1(\mu_{15})$), in escape energetic performances including E_t (Fig. 5.16 C), CoT (Fig. 5.16 E) and above all P_{mean} (Fig. 5.16 D), which emphasized the actual differences between μ_w and μ_1 body kinematics. This result constituted the first step towards the implementation of an effort test based on energetic performances and viscous flows, in order to highlight biological effects after exposure to low concentrations of toxicants.

5.2 A Toxicological study

As a second application of our experiment-driven numerical approach, we considered more biological-oriented experiments. Generally, biologists expose zebrafish eleuthero-embryos to particular drugs or chemicals to observe the main effects on escape swimming behavior. To that end, we first considered Chlorpyrifos-oxon (CPO) as a toxicant molecule which directly inhibits acetylcholinesterase at the neuromuscular junction and thus, directly affects the locomotor system. This molecule is well-studied alongside other organophosphorus (OP) which can be found in pesticides or chemical warfare nerve agents.Very low CPO concentrations were used which did not appear to affect the swimming kinematics of the EFP escape response. The aim of this preliminary work was to set up the conditions for an effort test by increasing the viscosity of the medium after exposure to the neurotoxicant and EFP stimulation. Numerical simulations could then enable access to energetic variables in addition to swimming kinematics, such as power and expended energy which cannot be computed directly from the experimental data. As introduced previously, our numerical approach may also enable to perform effort tests in order to analyze more accurately toxicological effects. By considering toxicological experiments, effort tests could be conducted both experimentally and numerically, unlike for the viscous modification. However in this work, we only focused as a first step, on numerical effort tests.

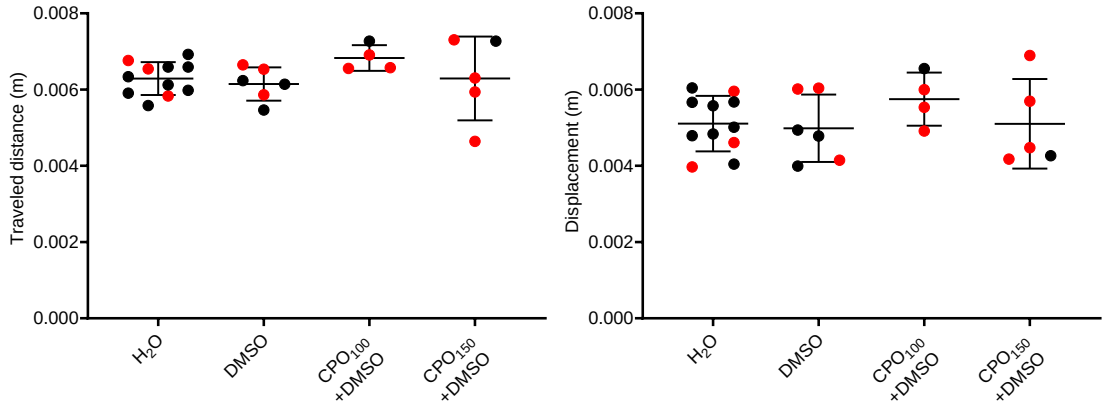
5.2.1 Experimental protocol

Individual videos were recorded for each condition in order to calculate a mean measurement. This section focuses on toxicological experiments and more particularly by using CPO toxicants. Zebrafish eleuthero-embryo was individually stimulated in water after having been exposed for 1 h to a nanomolar concentration of CPO. The CPO is stored as a stock solution with DMSO which is used as a solvent. DMSO does not *a priori* affects the locomotor system at low concentration. Therefore, a DMSO exposure is required in parallel to CPO for the control animals as DMSO was used as a vehicle solution for CPO. Each experiment consisted of two parts. First, zebrafish eleuthero-embryos were exposed to the chemical compounds by immersing the eleuthero-embryo into a solution with the presence of DMSO only or DMSO plus CPO. Afterward, we recorded up to three escape responses per eleuthero-embryo in the first 10 min after chemical exposure. On the one hand, only a few EFP stimulations have to be performed as the same eleuthero-embryo might be over-challenged and thus behave differently. Moreover, the amount of time was limited due to the rapid dissipation of the molecule effects into the zebrafish biological system. Indeed, zebrafish escape locomotion has been found to be partially recovered around ten minutes after the exposure, according to the exposition time or the toxicant concentration. On the other hand, several stimulations might be required to obtain a satisfying escape video representing the altered C-start escape swimming and including C-bend and counter-bend motions plus two cycles of fast-swimming. The typical phenotype induced by high CPO concentrations consists of balance loss and strong inhibition of locomotor activity which may result in an important body contraction [35]. With low concentrations, the three phases of the C-start are still well-perceptible as no such phenotype has been observed except for slight balance loss which cannot be modeled with our 3-DoF modeling. That is why, we also made sure the eleuthero-embryo was initially straight horizontal and did not show visible 3D motions. On the whole, we experimented two CPO concentrations: 100 nM and 150 nM alongside the control DMSO with a 0.1% concentration (v/v). We denote each respective experimental condition as: CPO_{100} (DMSO 0.1% plus CPO 100 nM), CPO_{150} (DMSO 0.1% plus CPO 150 nM) and $DMSO_{0.1}$. Pools of five eleuthero-embryos were exposed to the toxicant in six-well micro-plates in a total volume of 4 ml. Neurotoxicant stock solutions in 100% DMSO were diluted in fish water to achieve the desired working nominal concentration in 0.1% DMSO. Animals were exposed to CPO for 1 h. At the end of the exposure period, eleuthero-embryos were rinsed twice in clean fish water before performing the EFPMR test. Maximum three eleuthero-embryos per well have been individually stimulated, under the $t = 10$ min time limit. As a result, a total of 39 $DMSO_{0.1}$, 42 CPO_{100} and 32 CPO_{150} escape responses were recorded. As previously with the viscosity experiments, only the kinematics of one single video per eleuthero-embryo was analyzed which resulted in 16 $DMSO_{0.1}$, 13 CPO_{100}, and 11

CPO_{150} escape responses. In the end, we only conserved three $DMSO_{0.1}$-and-CPO_{100}-escape swimming and four CPO_{150}-escape responses, for modeling purpose. Modeled escape responses have been selected based on experimental and modeling constraints identical to the previous viscosity study.

5.2.2 Experimental kinematic performance

Similarly to the previous viscosity study, we first measured the global swimming kinematics by total traveled distance and displacement (Fig. 5.17), fast-swimming kinematics by traveled distance, swimming speed, tail-beat frequency, and rotation amplitudes (Fig. 5.18), and additional variables regarding the C-start characterization such as maximal C-bend angle or C-start angular velocity (Fig. 5.19).

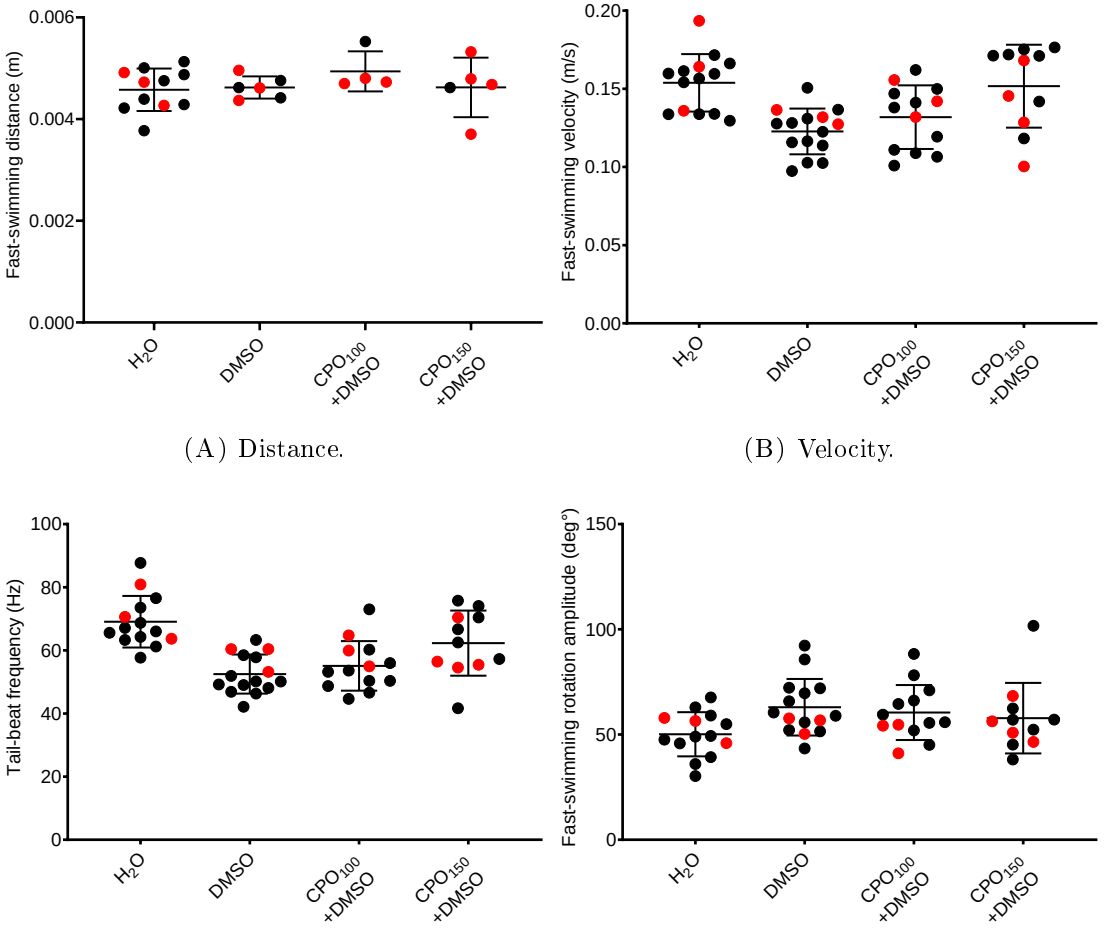


(A) Traveled distance. (B) Displacement.

FIGURE 5.17: **Two distances have been computed in the entire escape locomotion for each experiment and analyzed according to drug exposure.** For each toxicological condition, we analyzed 11 (μ_w), 6 ($DMSO_{0.1}$), 4 (CPO_{100}), 5 (CPO_{150}) experimental videos, for both the total traveled distance (Fig. 5.17A) and the raw displacement (Fig. 5.17B). Each experimental escape response is represented by an individual point. Note we show in red the experimental cases we further modeled with the numerical simulation. The mean value is represented alongside the SEM. Multiple statistical comparisons can be found in Table 5.7.

First, a multi-comparison statistical test has been conducted to evaluate the experimental trend (Table 5.7). As a result, no considerable toxicological effects have been observed through most of the kinematic variables. Only the fast-swimming average velocity and the tail-beat frequency demonstrated significant differences between water and $DMSO_{0.1}$, water and CPO_{100}, and between $DMSO_{0.1}$ and CPO_{150}.

As previously with the viscosity study, we selected $N = 3$ experimental escape response per experimental condition, which satisfied the demanding constraints required for numerical modeling. In the process, we also tried to choose representative experimental locomotions with respect to the mean of the original kinematics. Note we actually



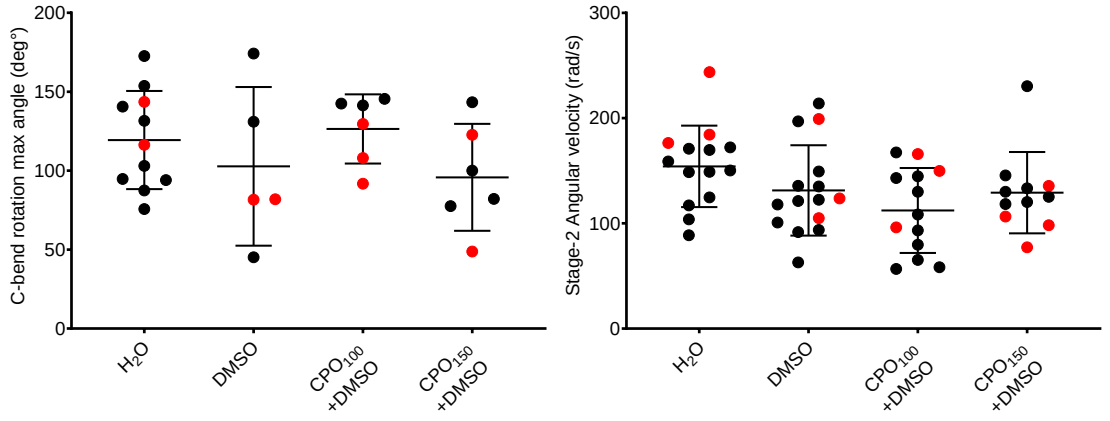
(A) Distance.

(B) Velocity.

(C) Tail-beat frequency.

(D) Rotation amplitude.

FIGURE 5.18: **Several variables have been computed only in the fast-swimming stage for each experiment and analyzed according to chemical molecule exposure.** For each toxicological condition, we analyzed 11 (μ_w), 6 ($DMSO_{0.1}$), 4 (CPO_{100}), 5 (CPO_{150}) experimental videos, for the fast-swimming traveled distance (Fig. 5.18 A) and 11 (μ_w), 16 ($DMSO_{0.1}$), 13 (CPO_{100}), 11 (CPO_{150}) escape responses for the translation velocity (Fig. 5.18 B), tail-beat frequency (Fig. 5.18 C) and mean rotation amplitude (Fig. 5.18 D). For the distance, two fast-swimming cycles were required while other quantities can be averaged. Each experimental escape response is represented by an individual point. Note we show in red the experimental cases we further modeled with the numerical simulation. The mean value is represented alongside the SEM. Multiple statistical comparisons can be found in Table 5.1.



(A) Maximal rotation angle. (B) C-start angular velocity.

FIGURE 5.19: **C-start response has also been analyzed for each experiment, according to chemical compound exposure.** For each toxicological condition, we analyzed 11 (μ_w), 5 ($DMSO_{0.1}$), 6 (CPO_{100}), 6 (CPO_{150}) experimental videos, for the maximal C-bend rotation angle (Fig. 5.19 A) and 11 (μ_w), 16 ($DMSO_{0.1}$), 13 (CPO_{100}), 11 (CPO_{150}) escape responses for the C-bend rotation velocity (Fig. 5.19 B, Stage-2 defined by counter-bend and C-bend rotations). For the maximum rotation, we only considered escape responses where the rotation is maximum during the C-bend motion to filter non stereotyped-behaviors. Each experimental escape response is represented by an individual point. Note we show in red the experimental cases we further modeled with the numerical simulation. The mean value is represented alongside the SEM. Multiple statistical comparisons can be found in Table 5.1.

TABLE 5.7: A one-way ANOVA and Tukey's test: Evaluation of toxicant exposure on the different kinematic variables.

	Disp.	Total Dist.	Dist.	Velocity
H_2O vs. $DMSO_{0.1}$	ns	ns	ns	***
H_2O vs. CPO_{100}	ns	ns	ns	*
H_2O vs. CPO_{150}	ns	ns	ns	ns
$DMSO_{0.1}$ vs. CPO_{100}	ns	ns	ns	ns
$DMSO_{0.1}$ vs. CPO_{150}	ns	ns	ns	**
CPO_{100} vs. CPO_{150}	ns	ns	ns	ns (0.0846)

	Tail-beat freq.	Rot. Amp.	CBmax	Ang. Velocity
H_2O vs. $DMSO_{0.1}$	****	ns (0.0616)	ns	ns
H_2O vs. CPO_{100}	***	ns	ns	*
H_2O vs. CPO_{150}	ns	ns	ns	ns
$DMSO_{0.1}$ vs. CPO_{100}	ns	ns	ns	ns
$DMSO_{0.1}$ vs. CPO_{150}	*	ns	ns	ns
CPO_{100} vs. CPO_{150}	ns	ns	ns	ns

For each kinematic variable, we show the experimental impact of viscosity between each condition (N=11-16) by summarizing the p-value. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

TABLE 5.8: A two-way ANOVA test: Evaluation of kinematic impacts of numerical simulations and drug exposure.

Source of Variation	Traveled distance % TV		Velocity % TV		Displacement % TV		Distance % TV	
Chemical exposure	7.172	ns	26.73	ns	3.356	ns	3.231	ns
From exp. to num. values	33.63	****	33.46	****	20.86	****	60.69	****
Interaction	2.245	ns	3.537	*	2.197	ns	4.032	ns

Summary (Sum.) of the two-way ANOVA test regarding numerical modeling and drug exposure impacts based on N=13 experimental and numerical data, including % of total variation (TV). Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

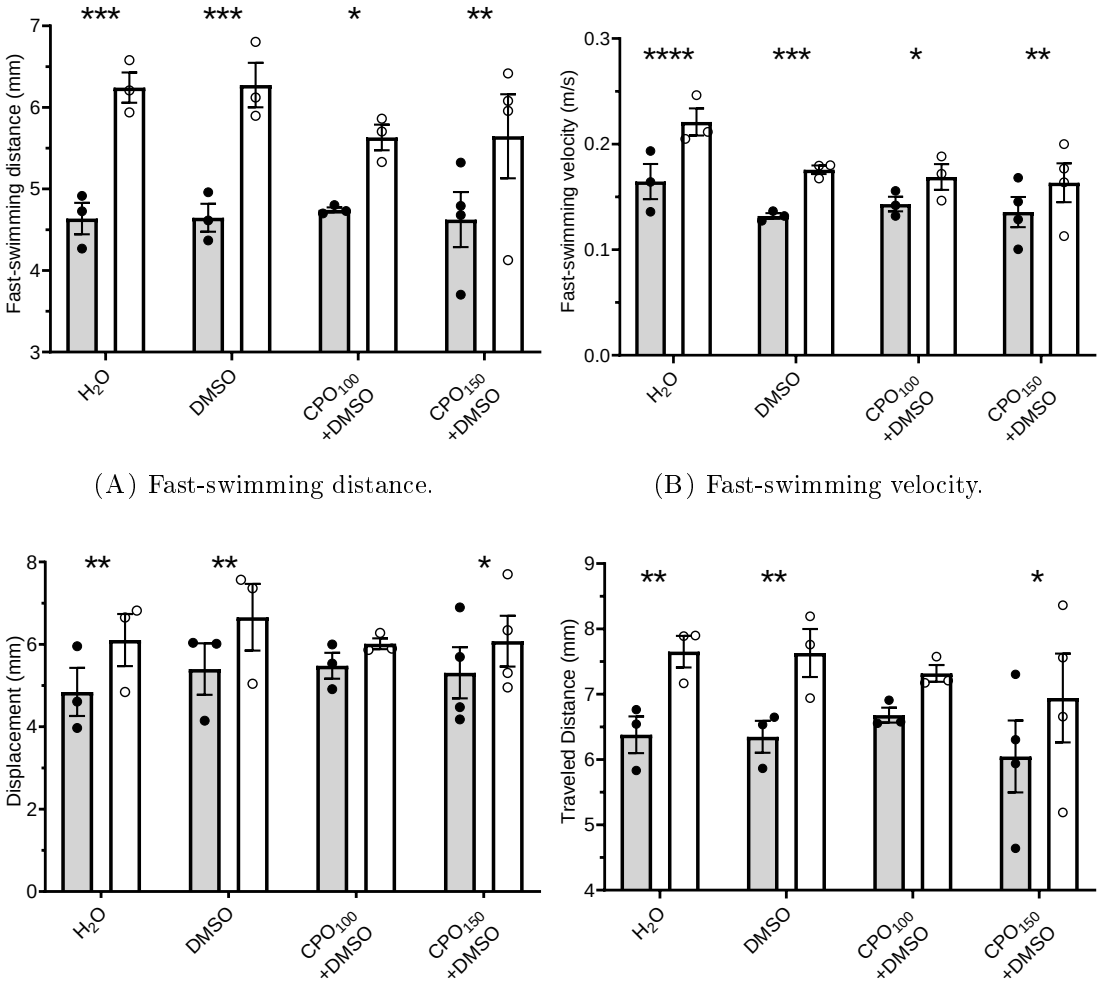
selected $N = 4$ experimental videos for CPO_{150}-conditions, as kinematic results were highly more variable.

5.2.3 Impacts of numerical simulations

Numerical simulations have then been performed for each experimental condition. Note we simulated $N = 3$ experimental escape response for μ_w, $DMSO_{0.1}$, and CPO_{100} conditions while $N = 4$ experimental locomotions have been considered for the CPO_{150}-condition due to higher variability observed in the biological phenotype. This work constituted a preliminary computational study in order to provide first numerical results for biological applications. That is why, only a few experiments have been modeled. Beforehand, we compared the numerical simulations against the experiments (Fig. 5.20). Although escape responses seemed well-reproduced at first (see Chapter 4), all considered kinematic variables were over-estimated due to minor accelerations, for all toxicological conditions. Previously, we showed numerical errors were not correlated to chemical compounds (Chapter 4). That is why, all numerical distances (Figs. 5.20 A, 5.20 C, 5.20 D) and velocity (Fig. 5.20 B) were found significantly higher than experimental measurements.

Afterward, we evaluated and compared more specifically the toxicological effects between the experiments and the numerical kinematics (Fig. 5.21). As a result, experimental kinematics was not affected by the chemical compounds exposure in the conditions used, as observed within the original data-set, unlike the previous viscosity study where kinematic variables were found very sensitive to viscosity change. Similarly, numerical simulations did not present any particular effects. In addition to one-way ANOVA tests, two-way ANOVA statistical tests have also been conducted to analyze both chemical exposures and impacts of numerical simulations (Table 5.8). As previously analyzed, the influence of numerical errors was significantly larger than toxicant effects. In definitive, the whole biological impact on translation kinematics meaning the lack of considerable effects due to toxicants, has been well-reproduced by numerical simulations although a constant error seems to be introduced for all chemical-exposure experiments.

For information, we also computed Re from numerical simulations (Fig. 5.22) in order to illustrate the swimming regime in which zebrafish eleuthero-embryos evolve after



(A) Fast-swimming distance.

(B) Fast-swimming velocity.

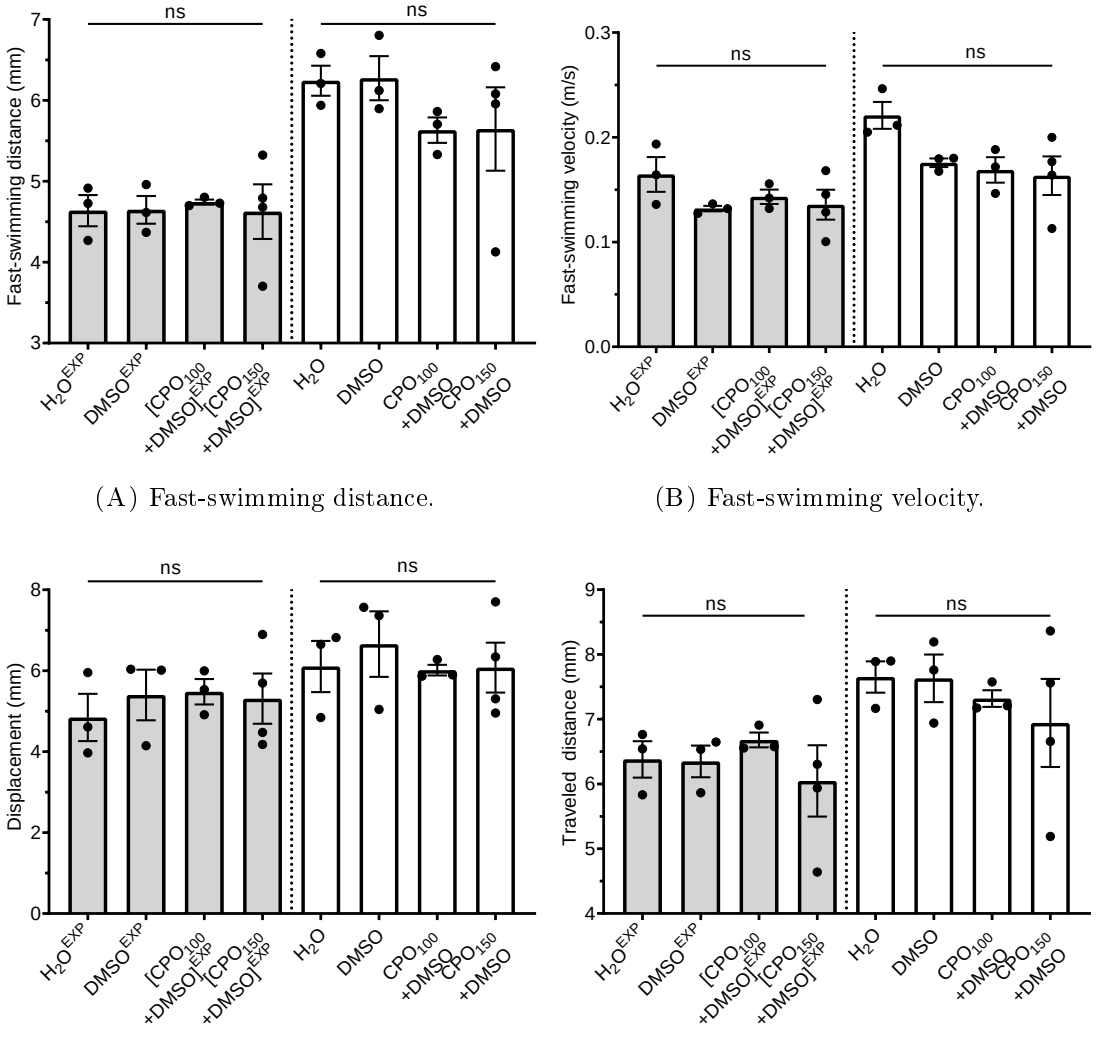
(C) Displacement.

(D) Traveled distance.

FIGURE 5.20: **Comparison of numerical simulation kinematics against toxicological experiments.** We represented in parallel both the experimental (filled circles, colored bars) and the numerical (empty circles, white bars) kinematics represented by the fast-swimming traveled distance (Fig. 5.20 A), the fast-swimming average velocity (Fig. 5.20 B), the raw displacement (Fig. 5.20 C) and the global traveled distance (Fig. 5.20 D). Note for each condition, we show the mean of escape responses alongside individual values and the SEM. Statistics from a multiple comparisons Sidak's test. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

chemical exposure to low concentrations. As all escape responses have been performed in water, the flow regime was expected to not change considerably although chemical exposure could have affected dramatically the escape swimming speed.

In definitive, zebrafish eleuthero-embryos seem to not have been affected by chemical exposure at the concentrations used, regarding the considered kinematic variables. On the one hand, a larger sample size might be required to observe larger biological effects of toxicants due to the high variability and lack of control among individuals. On the other hand, slight differences between altered body kinematics might be emphasized by numerical simulations.



(A) Fast-swimming distance.

(B) Fast-swimming velocity.

(C) Displacement.

(D) Traveled distance.

FIGURE 5.21: **Comparison of toxicological effects between numerical simulation kinematics and experiments.** We represented in parallel both the experimental (filled circles, colored bars) and the numerical (empty circles, white bars) kinematics represented by the fast-swimming traveled distance (Fig. 5.21 A), the fast-swimming average velocity (Fig. 5.21 B), the raw displacement (Fig. 5.21 C) and the global traveled distance (Fig. 5.21 D). Note for each condition, we show the mean of escape responses alongside individual values and the SEM. Statistics from two distinct multiple comparisons Tukey's tests for experimental and numerical kinematics. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

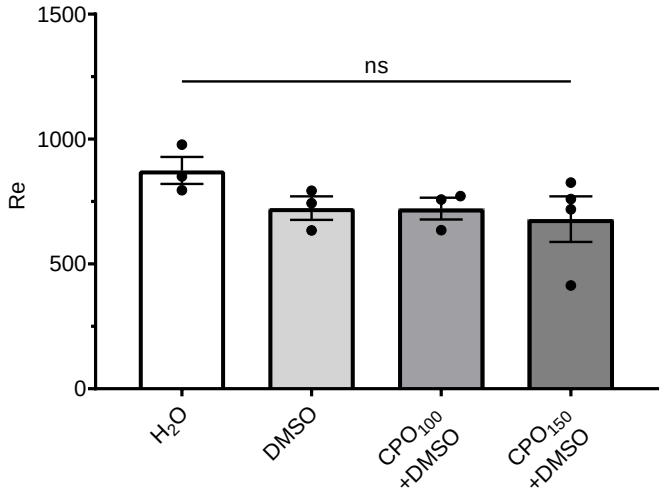


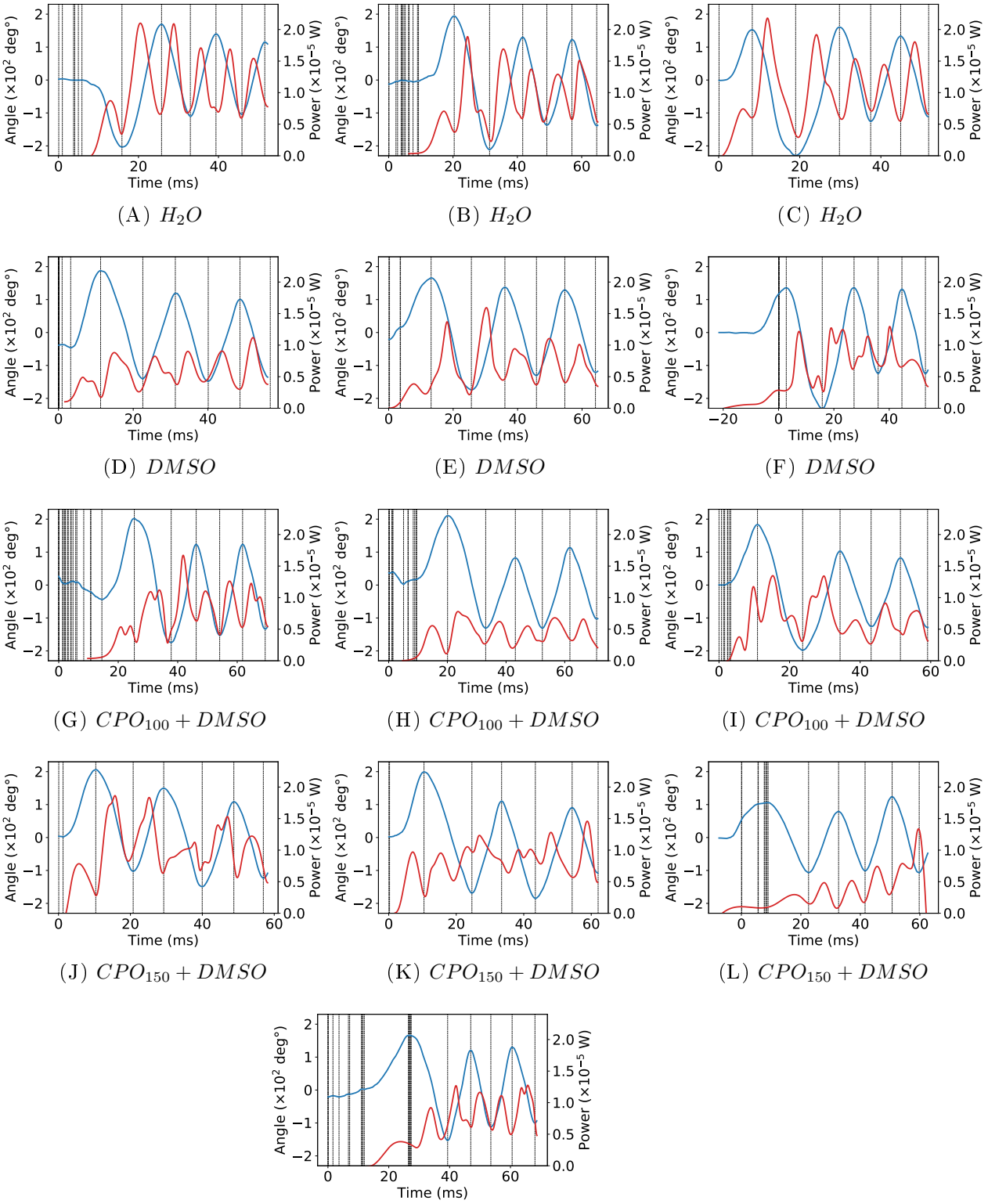
FIGURE 5.22: **Flow regime: variation of Re according to chemical compound exposure.** Note Re has been computed with the average velocity of the complete escape simulations. We represented the mean and the SEM. Non-significant variations have been found. Statistics from a one-way ANOVA and Tukey's test. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

5.2.4 Energetic performances

First of all, we computed the power output for all chemical-exposed test cases (Fig. 5.23). Regarding energetic performances, we computed for each escape response E_t, P_{mean} and the CoT, as in the previous study. Due to the high variability in experimental kinematics and the overall close kinematics between each experimental condition, energetic measurements also showed no particular variation (see Fig. 5.24).

As with kinematic quantities, no supplementary amount of energy has been expended or saved by intoxicated zebrafish eleuthero-embryos. In fact, the result cannot be as highlighted as in viscous flows due to the lack of characterization of the impact of toxicant conditions. Statistical tests have been conducted to demonstrate chemical exposure had no particular impacts on energetic performances.

Although no statistical variation can be emphasized by the different energetic quantities, a slight decrease in $DMSO$ seems visible, especially regarding the mean power. The poor results of our numerical approach with a more real-like biological application have several explanations due to chemical concentration used, type of toxicant used, experimental process, sample size, mathematical modeling and numerical errors, which will be reviewed hereafter. Nevertheless, potential chemical-exposure effects could be observed in the case of specific effort tests, which may amplify potential differences in swimming performances.



(A) H_2O (B) H_2O (C) H_2O

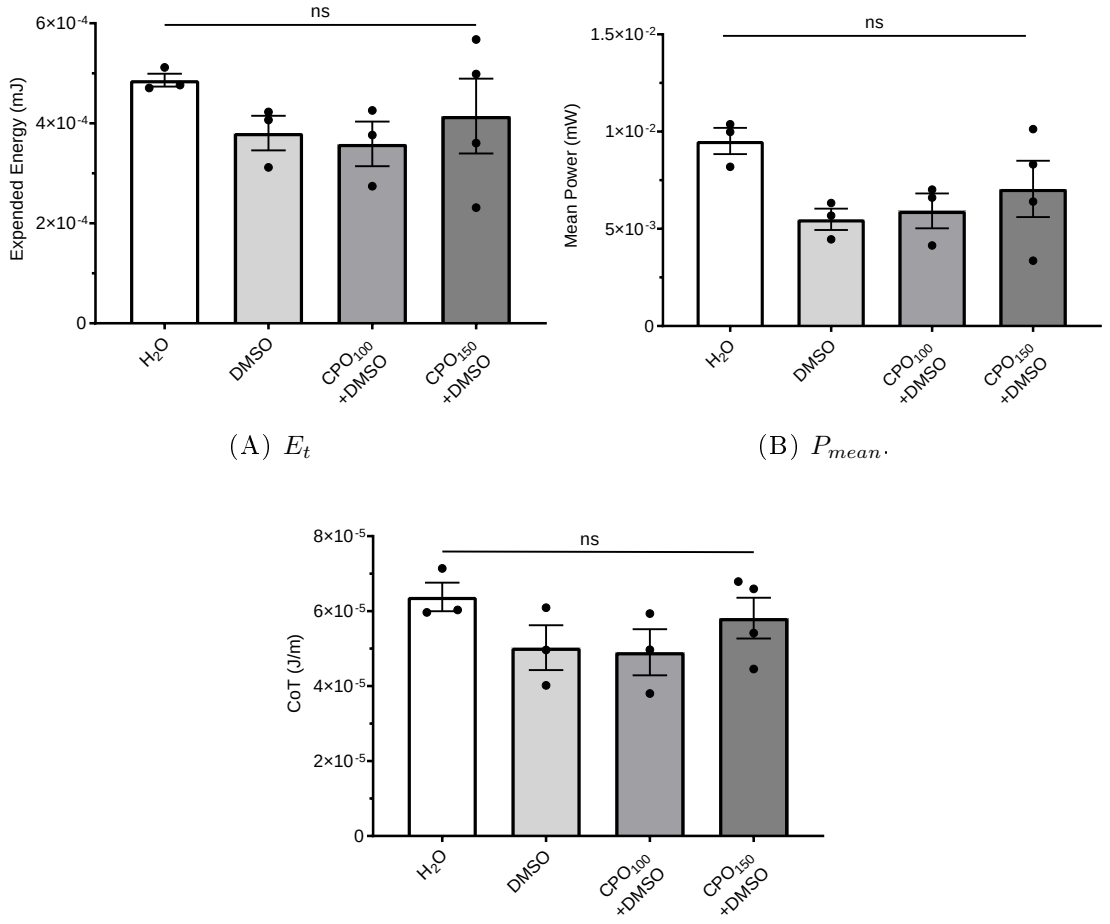
(D) $DMSO$ (E) $DMSO$ (F) $DMSO$

(G) $CPO_{100} + DMSO$ (H) $CPO_{100} + DMSO$ (I) $CPO_{100} + DMSO$

(J) $CPO_{150} + DMSO$ (K) $CPO_{150} + DMSO$ (L) $CPO_{150} + DMSO$

(M) $CPO_{150} + DMSO$

FIGURE 5.23: **Illustration of body deformation effects on the power output, for all exposed test cases.** All Figs. 5.23 A-5.23 M correspond to an individual escape response, each row being associated to a toxicant condition. For all figures, we represented in blue the bend amplitude alongside the power output in red.



(A) E_t

(B) P_{mean}.

(C) CoT.

FIGURE 5.24: **Representation of escape swimming energetic performances according to chemical molecule exposure.** For each escape response, we computed E_t (Fig. 5.24 A), P_{mean} (Fig. 5.24 B) and CoT (Fig. 5.24 C). Note for each condition, we show the mean of $N = 3$ locomotion alongside individual values and the SEM. Statistics from a one-way ANOVA and Tukey's test. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

5.2.5 Mix with numerical viscosity: a first numerical effort test

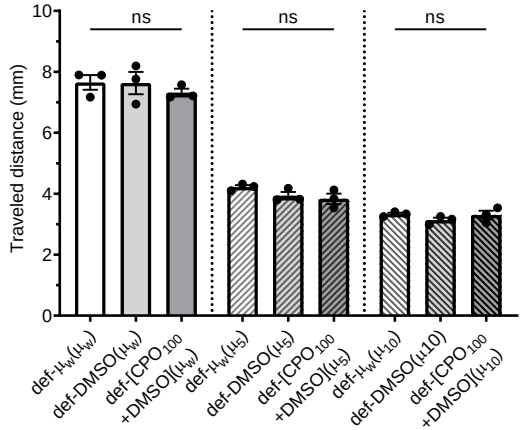
As introduced previously, different effort tests can be conducted when considering toxicological or pharmacological experiments as the zebrafish eleuthero-embryo can be challenged by fluid viscosity, either experimentally, or numerically. Numerical simulations can easily be performed using different fluid viscosity provided the Reynolds regime does not increase dramatically. For low-Reynolds numbers, only numerical stability might raise slight issues if not handled carefully. We demonstrated in the previous viscosity study, zebrafish eleuthero-embryos can actually adapt its body deformations to the surroundings, by using efficient and adapted body kinematics. Therefore, experimental and numerical effort tests constitute very different tests as potential adaptations cannot be modeled by numerical effort tests. In contrast, zebrafish eleuthero-embryos may adapt

its kinematics when performing experimental effort tests. Note the full experimental study needs to be conducted to investigate the response to viscosity after being exposed to chemical compounds. As a preliminary work, we first assumed impacts of chemical compounds would be more sensible to a numerical effort test as no efficient adaptation is incorporated. Considerable energetic performances have already been emphasized by performing such an effort test in the previous viscosity study. That is why, we implemented in a straightforward fashion first numerical effort tests by performing fictitious numerical simulations in highly viscous flows based on chemical-exposed zebrafish body deformations. To that end, we first considered μ_5 and μ_{10} fluid conditions (see Fig. 5.25).

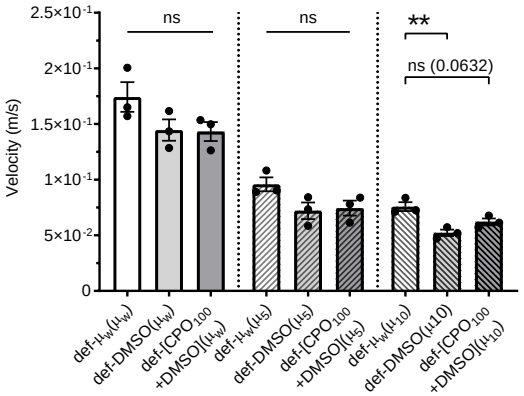
As a result, no significant variation was found for the three viscous media, in total traveled distance (Fig. 5.25 A), E_t (Fig. 5.25 C), and CoT (Fig. 5.25 E). For the μ_5 viscosity, neither P_{mean} (Fig. 5.25 D) nor the average velocity (Fig. 5.25 B) was particularly affected. However, a significant impact has been observed on the total average velocity in the μ_{10}-viscous condition. As noticed before when modifying the swimming frequency, it seems P_{mean} was also slightly impacted by the μ_{10} fluid condition. Therefore, those results suggest zebrafish eleuthero-embryos might be less efficient after being exposed to chemical compounds when immersed in highly viscous flows. Nevertheless, no difference could have been highlighted between control animals *i.e.* DMSO-exposed, and [DMSO+CPO]-exposed eleuthero-embryos which was not expected. Indeed, the previous viscosity study had strongly suggested viscous flows can amplify considerably minor differences in swimming performances. Presently, all differences between DMSO-exposed and [DMSO+CPO]-exposed experiments seemed to be preserved throughout the viscosity increase. Thus, the numerical effort test has been found to be less efficient in the case of toxicological experiments, which may be due to the impacts we can observe on the power output (see Fig. 5.23). Indeed, it seems power outputs have been slightly affected by chemical exposure as power peaks seem to be less pronounced relatively to H_2O or viscous escape responses. As a consequence, the energetic quantities we defined for characterizing the power output may need to be complemented by other measurements, more adapted toward toxicological effects.

5.3 Conclusion

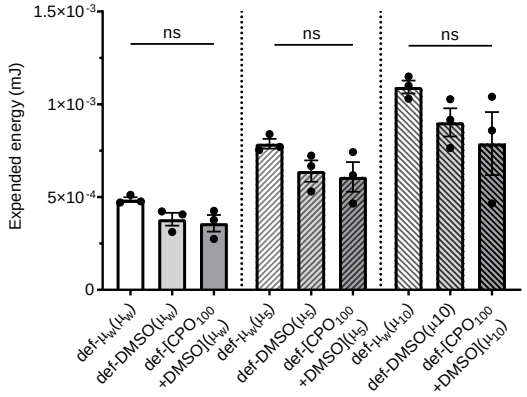
The experiment-driven numerical approach was performed onto two types of biological experiments of 5 dpf eleuthero-embryo escape responses. First, a full viscosity study has been explored both experimentally and numerically. Secondly, the effects of a neurotoxicant on the kinematic and energetic performances have been investigated. The important factor which drives biologists towards numerical modeling resides in the ability to compute accurate experimental swimming energetic performances while simulating viscous flow conditions. Thus, the viscosity factor has been studied extensively in order



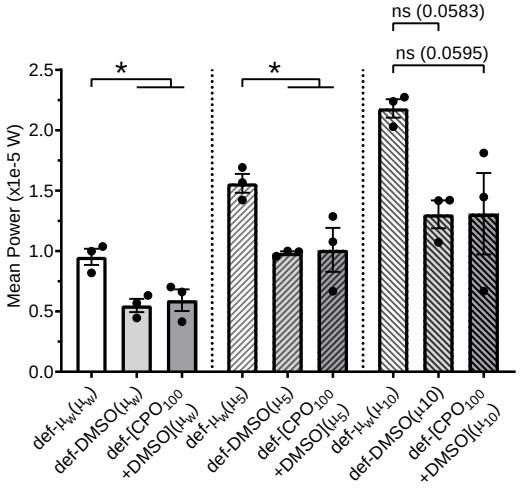
(A) Traveled distance.



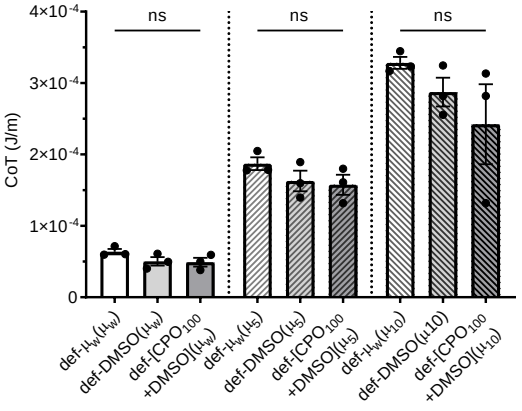
(B) Velocity.



(C) E_t.



(D) P_{mean}.



(E) CoT.

FIGURE 5.25: **Effort test: comparison of escape swimming kinematic and energetic performances according to chemical compound exposure and fluid viscosity (μ_w-μ_5-μ_{10}).** For each escape response, we computed the total traveled distance (Fig. 5.25 A), the average velocity (Fig. 5.25 B), the total expended energy (Fig. 5.25 C), the mean power output (Fig. 5.25 D) and the cost of transport (Fig. 5.25 E). For all considered variables, we compared three experimental escape conditions ($def-H_2O(\mu_w)$, $def-DMSO(\mu_w)$ and $def-CPO_{100}(\mu_w)$) against their fictitious counterparts in high viscosity fluids ($def-H_2O(\mu_5)$, $def-DMSO(\mu_5)$, $def-CPO_{100}(\mu_5)$, and $def-H_2O(\mu_{10})$, $def-DMSO(\mu_{10})$, $def-CPO_{100}(\mu_{10})$). Note for each condition, we show the mean of $N = 3$ locomotion alongside the SEM. Statistics from three distinct one-way ANOVA and Tukey's test are shown for each viscous condition, respectively. Non-significant (ns): $p \geq 0.05$, *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$, ****: $p < 0.0001$.

to implement experimental or numerical effort tests *via* our numerical tool. Beforehand, the numerical results provided by numerical simulations have been discussed in Chapter 4 to ensure all computed quantities would faithfully represent the real-world phenomena. In particular, numerical errors obtained on both viscous and toxicological experiments have been compared and highlighted a slight over-estimation of experimental kinematics although most of the numerical simulations tended to well-reproduce experimental trajectories. For all experimental conditions, we only modeled three escape responses. In the viscosity study, the modeled data-set has been considered as well-representative of the original experiments. Furthermore, viscosity changes were predominant in kinematic variations, compared to numerical errors as swimming kinematics was found very sensitive to viscosity. In particular, low-viscosity conditions provided considerably different kinematic results except between μ_w and μ_1 while no significant variations have been observed between the three highest viscous flows. However, energetic quantities provided multiple supplementary amounts of information. First, both the amount of energy expended or consumed power were nearly constant regardless of fluid viscosity. In addition, the CoT was found linearly dependent on fluid viscosity which enabled to highlight significantly high viscous flow differences. Nevertheless, very close body deformations as between μ_w and μ_1 have not been emphasized by energetic performances. That is why, a first effort test has been performed by immersing μ_w and μ_1 body kinematics into higher viscous flows, numerically. As a result, significant impacts have been observed in consumed power and CoT between those fictitious simulations. As for toxicological experiments, the effects of CPO dissolved in $DMSO_{0.1\%}$ have been investigated. Thus, zebrafish eleuthero-embryos have been exposed to DMSO, and DMSO plus CPO before the stimulation. As CPO might affect eleuthero-embryo's body balance, only low concentrations have been considered. Swimming kinematic variables suggested higher escape response variability relatively to viscous experiments. Besides, only three escape responses per chemical compound exposure have been modeled and thus, the considered experimental escape responses may not properly represent the actual effects of CPO on escape locomotion. Moreover, numerical simulations had a more important impact than toxicological treatments onto escape motions as no specific toxicological effects have been observed. In this case, swimming energetic performances have not particularly enhanced the analysis of DMSO, and DMSO plus CPO results. That is why, we then challenged the original experimental conditions by performing a numerical effort test based on chemical-exposed zebrafish eleuthero-embryos. We numerically increased the fluid viscosity to observe different energetic behaviors. As a result, we only observed a significant impact on the average velocity when immersed in μ_{10}-viscosity, between non-exposed and chemical-exposed zebrafish eleuthero-embryos. These results suggest that we added an important amount of biases which might have lessened considerably either the biological effects due to toxicological treatments or the effort test efficiency. First, only escape locomotions which were the least affected by chemicals in terms of visual phenotype, were considered. Thus, we might have chosen

only the less sensitive zebrafish eleuthero-embryos. Secondly, we performed numerical simulations with only less than five experimental escape responses to represent a high variability phenotype, especially compared to the weak impact of chemical compounds on swimming kinematics. Therefore, statistical differences are more difficult to highlight due to the tiny sample size. Thirdly, we only considered low concentrations of toxicants to not produce important phenotypes we cannot properly model. Thus, we do not know whether zebrafish eleuthero-embryos are affected by such low concentrations. Finally, although numerical errors were negligible in the viscosity study due to high sensitivity to fluid viscosity, numerical errors might have larger impacts with finer biological experiments. In particular, we emphasized the largest numerical errors might be due to modeling issues such as 3D motions or elastic parts of the body. According to the biological impacts of CPO, our mathematical model might not encompass precisely the effects of such toxicant. That is why, the choice of this chemical compound might also be questionable as maybe not well-suited to our numerical modeling. Further experiments and simulations such as effort tests would be required to provide larger insights. Furthermore, other neurotoxicants must be investigated.

Chapter 6

General conclusion and perspectives

In this chapter, the essence of the experiment-driven numerical approach will be recapitulated first. Secondly, the numerical and modeling validation test data have been summarized. The results of our computational model when applied toward biological studies including a viscosity and a toxicological study are recalled in a larger context. Afterward, the main limitations of our numerical approach are discussed regarding either the general approach used or the methodology used in biological studies. Finally, the main perspectives of this work might enable are developed. The work carried out as part of this *PhD* project will be formalized in the form of scientific manuscripts, the first of which being *Ravel et al., Modeling escape response of zebrafish eleuthero-embryo reveals energetic constraints and efficient body deformation adaptation to viscous flows* (manuscript in preparation).

6.1 Overview

An experiment-driven numerical approach has been developed to realistically assess the swimming performances of moving and deforming bodies immersed within fluid flows. In particular, our approach focused on zebrafish eleuthero-embryo escape swimming but could be generalized to biomimicry, engineering, or any field involving motions with experimental imaging data. The goal was to develop a numerical tool able to simulate the motion of the body in a realistic way while providing supplementary amounts of information about the swimming performances. That is why, this work was oriented toward an approach directly based on experimental imaging data. As for the swimming performances, biologists are particularly interested in energetic performances as it links directly with the underlying biological processes and biomechanics such as muscle activity or locomotor system which produce the swimming kinematics we can observe. Thus, the energetic performances are particularly attractive to enhance the understanding of swimming and enable to capture finer and additional amounts of

information to explain swimming kinematics. To that end, CFD methods were involved as they constitute an important numerical tool to provide complete insights of flow motions alongside its interaction with immersed bodies which includes the calculation of body motion due to fluid forces. In the case of fish-like swimming simulations, self-propelled fish-like motions are generally focused on steady and cyclic movements which are modeled *via* mathematical formulations such as harmonic body deformations [9, 10, 12, 16, 24, 41, 43, 51, 58, 59, 66, 67, 69, 108, 125]. Eventually, the mathematical model might be calibrated using experimental observations but do not generally represent specific experimental kinematics [66, 67]. In the literature, only a few studies have integrated experimental swimming kinematics at the cost of actual body computations *i.e.* the computation of body kinematics *via* Newton's laws [27, 65, 67, 68]. Others have directly incorporated experimental kinematics into a CFD framework without a full fluid-structure interaction [49, 129].

In this work, we first extracted the body kinematics from experimental imaging by using the optimal transportation theory as it is particularly adapted to finding the optimal deformation path followed by each pixel of the zebrafish silhouette. However, 2D numerical simulations were not in good accordance with experimental kinematics although particular simulations might seem well-satisfying. In fact, the results suggested an accumulation of several errors due to the Eulerian representation of both the silhouette *i.e.* imaging approximations and the deformation velocity field. This could be due to optimal transportation approximations which are based on segmented and aligned zebrafish images. Indeed, the *Procrustes Analysis* step required to subtract rigid kinematics to obtain the aligned original body deformations might also affect the resulting velocity field. Finally, 2D numerical simulations are based on 2D hypotheses which may not be adequate to model a 3D motion. As a result, we obtained very sensitive numerical simulations especially regarding the trajectory comparison across various experimental escape responses. In addition to realistic body deformations, we aimed to base numerical simulations on a faithful representation of the zebrafish eleuthero-embryo in 3D.

In the literature, only a few studies have reconstructed the whole morphology of a zebrafish eleuthero-embryo and larva in 3D [58, 65, 67, 124, 131]. In this work, we used a high-resolution transverse-section dataset combined with the dorsal and lateral views of a zebrafish eleuthero-embryo to interpolate the 3D zebrafish shape by using optimal transportation. The resulting 3D reconstruction has then been processed to be represented by Lagrangian markers. We then used the initial and straight 3D reconstruction and the extraction of the experimental kinematics of the midline to deform the 3D shape according to midline deformations. For this purpose, we assumed each cross-section remains orthogonal to the midline during deformations as previous studies [12, 23, 25, 30, 38, 131]. As a result, we tracked the deformation velocity at the surface *via* Lagrangian markers. Finally, we enhanced and adapted the *Procrustes Analysis* towards 3D deformations which separate accurately the rigid rotation from the

body deformations.

In the literature, few groups have implemented efficient and automated algorithms to track the zebrafish body across the locomotion either in 2D [38] or in 3D [131], by fitting a zebrafish model to experimental images. In our case, the eleuthero-embryo model has been reconstructed on top of each deformed midline and thus, we tracked the midline kinematics instead of the zebrafish silhouette displacements. As for the *Procrustes Analysis*, most of the advanced studies which integrated the experimental kinematics into a CFD code [65, 67, 68] mainly considered the body rotation could be represented by the heading angle. Recently, in an inverse-dynamics framework [124, 131], body angular quantities have been computed through the momentum conservation. Actually, we used a similar method to compute the body angle *i.e.* tilt angle as the implemented 2D *Procrustes Analysis* can be viewed as an estimation of the momentum conservation. Eventually for straight-line swimming, SVD techniques can be used to obtain the direction of the fish frame of reference [50]. Finally, we can find few studies which directly prescribed the full experimental swimming kinematics including both rigid and deformation motions, to the CFD solver, what bypasses the *Procrustes Analysis* step [49, 129]. Therefore, our experiment-driven numerical approach embeds the experimental deformations while computing the rigid kinematics from body dynamics equations. It is important to note the body deformations are not modeled from an elastic or a biological model, unlike few groups which implemented a full neuro-mechanical model for fish swimming [98, 119]. Besides, others have reconstructed the bending moments from the force distribution to characterize the body deformations [129]. In this work, only the bend amplitudes have been calculated while tracking the midline kinematics.

In terms of numerical validation, we first studied the convergence of 3D numerical simulations according to the considered computational mesh. Afterward, the importance of a proper *Procrustes Analysis* has been illustrated. The actual numerical validation against an experimental case showed overall satisfying accordance although a slight acceleration has been observed during the fast-swimming stage of the zebrafish escape response. Nevertheless, the most complex body deformations which consist of a C-shape and a string counter-bend motion, have been well-reproduced. A sensitivity analysis for various parameters such as *Procrustes Analysis* parameters and center-of-mass approximations, boundary effects and 3D modeling, and data noise processing, has then been conducted. The objectives were to show the impacts of particular parameter choices on the overall simulation and the resulting swimming performances and to explain accurately the numerical error between experimental and numerical kinematics which especially occurred during the fast-swimming stage. As a result, swimming energetic performances do not change considerably even though swimming kinematics might be slightly altered. Above all, the results suggest mathematical modeling is more prevalent in the observed numerical error than various numerical approximations. In particular, we highlighted 3D motions and elastic deformations might have an important role. Finally, the reproducibility of our experiment-driven numerical approach has been analyzed

based on numerical simulations performed with viscous and toxicological experiments. Thus, most of the experiments have been well-reproduced except few over-estimations of the trajectory as in the initial test case. In definitive, first numerical validations were very satisfying although the numerical errors raised might be considerable. In the literature, only few numerical approaches have been validated directly against the actual experiments [27, 65, 67, 68]. Indeed, computational studies generally either compare the solution with the experimental flow through PIV or other numerical studies. As a result, the overall kinematics were overall very satisfying. However, these preliminary results have not been validated over a wide range of test cases including short and long escape responses, weak and strong C-starts, or even altered escape locomotions. In this work, the impacts of numerical errors and more largely the sensitivity of the numerical method have been extensively investigated.

In addition to evaluating the performance of our numerical tool, viscous and toxicological experiments have been conducted to assess the impacts of our computational model. The first application consisted of modifying the actual fluid viscosity to analyze various swimming performances for a flow regime ranging from $Re = 10$ to 1000. As a result, an exponential decay has been observed in kinematic variables including traveled distance, raw displacement, and average velocity which tended to not differentiate high-viscosity flows. Afterward, we highlighted zebrafish eleuthero-embryos seem to expend a limited amount of energy and consume an identical amount of power across the escape motion for all fluid viscosity conditions. Moreover, we defined the CoT to measure the escape efficiency as the ratio of expended energy over the traveled distance, and the CoT increased linearly according to fluid viscosity which was particularly interesting to observe significant differences in highly viscous flows. Besides, viscous experiments have been used to initiate the implementation of fictitious simulations such as effort tests which demonstrated great capabilities by considering water and viscous body deformations. The actual biological application consisted of zebrafish eleuthero-embryo exposure to specific chemical compounds as DMSO or CPO, before maximizing the eleuthero-embryo's escape response by EFP stimulation. As discussed previously, numerical simulations did not provide considerable insights on molecule effects, supplementary to the experimental observations as no impact could have been emphasized, either in kinematic or energetic performances. That is why, a numerical effort test has also been performed by simulating fictitious viscous escape responses. However, results were found hardly significant due to several experimental and numerical limitations. Note a whole viscosity study has been specifically conducted both experimentally and numerically in order to implement an automated effort test capable of measuring an alteration of the locomotor system Notice effort tests are particularly attractive as fluid viscosity can be modified directly within the computational code and thus, both experimental and numerical effort tests could be analyzed and compared. In the literature, only few experimental viscosity studies have been conducted by using a Newtonian fluid [28–30, 55]. However, very few have analyzed EFPMR as EFP stimulations have only

emerged in recent years. In addition, no study has analyzed the swimming kinematics of zebrafish eleuthero-embryo across a wide range of fluid viscosity. Danos et al. [30] found in particular that escape response timings and stage duration were conserved by viscosity change. In this work, traveled distances and fast-swimming velocity have been found decaying over viscosity but no significant variation has been observed regarding stage duration. As for computational studies, several groups have numerically modified the dynamic viscosity to change the Reynolds regime of fish swimming [66, 119]. Nevertheless, numerical simulations were necessarily counter-factual as the considered body deformations were not "real-world" kinematics. Besides, our results seemed in good accordance with the few existing results as the swimming speed has been found to be reduced approximately by 50 % between μ_w and μ_{10} [66, 119], which is similar to the fast-swimming decay observed when performing the fictitious simulations. To our knowledge, effort tests have not been exploited in the literature as both experimental and numerical studies need to be conducted.

6.2 Main limitations

On the one hand, we discussed the limitations of our numerical approach. First, the extraction of the midline kinematics was not straightforward to compute given the high-resolution in space and time of the experimental imaging data. In particular, the tracking of the zebrafish head has to be handled carefully to not degrade significantly the resulting 3D body reconstruction and its deformation velocity. Then, the following *Procrustes Analysis* had to be processed carefully to not introduce spurious rotation movements within body deformations, although energetic performances do not seem highly sensitive to smoothing parameters. The general limitations of the experiment-based numerical simulations reside first in the three degree-of-freedom motion modeling due to zebrafish morphology. Indeed, we observed the zebrafish eleuthero-embryo was unstable in the water column when we did not impose specific constraints and modeled the entire motion as free-swimming, due to the characteristic asymmetry of the zebrafish eleuthero-embryo shape regarding the dorso-ventral axis in particular. In addition, we conducted various test cases to explore the source of the acceleration in kinematics with respect to experimental measurements. Only three hypotheses have not been challenged: the actual zebrafish shape, the homogeneity of mass density, and the elastic component of specific deformations. Indeed, we do not know how important the role of the zebrafish shape is. In particular, the MFF measurements might be refined to correspond more adequately to the actual zebrafish shape, especially regarding its thickness. As for the mass density, the MFF does certainly have a different mass density with respect to the rest of the body given the particular biological difference. However, the importance of such difference is largely unknown as all previous studies in the literature have modeled mass density as homogeneous [131].. Regarding elasticity, the

hypothesis concerns both the MFF which might not be as rigid as modeled in this work, and the tail tip which was modeled as slightly rigid to impose an identical length for each 3D reconstruction. Again, further investigations might be required to understand entirely the origin of numerical errors mainly observed in the center-of-mass trajectory and the corresponding velocity. On the other hand, we discussed the limitations of the experimental methodology used for demonstrating the potential of our experiment-driven numerical approach. First, the biological phenotype had to be predominant relatively to numerical errors due to the modeling considerations described previously. Viscous experiments were particularly significant and reproducible which enabled a reliable analysis of numerical simulations. However, toxicological experiments with the molecules used, showed no significant results which lessened the potential insights provided by numerical simulations. Secondly, the modeled experiment had to be representative of the experimentally observed phenotypes. For instance, each viscous experiment considered toward modeling was representing the experimental observations on swimming and body kinematics. Instead, chemical-exposure experiments were more difficult to model simply due to the higher variability in escape responses with intermediate chemical concentrations. Indeed, high CPO concentrations (> 200 nM) escape responses could not be modeled as they highly impacted the locomotor behavior while low concentrations might not be relevant enough to be emphasized by numerical simulations. The amount of difference in body kinematics required to observe significant energetic performances through numerical simulations is complex to predict due to the potential interference with numerical errors. More importantly, the number of experimental escape responses considered had to be tuned according to biological effects observed in kinematics. For instance, we were able to observe significant results in energetic performances by using only three experimental videos per viscous condition. Instead, the numerical results obtained with the toxicological study strongly suggest a larger sample size is certainly needed to provide a finer analysis of the actual energetic performances. The main constraints for performing numerical simulations in a massive amount reside both in the quality of experimental videos and in the process of midline tracking which is currently neither optimized nor fully automated. Finally, the choice of the considered toxicant might be important due to its potential biological effects. Therefore, other chemical compounds and conditions need to be explored. Indeed, swimming performances might be enlightened by molecules which affect directly the muscle activity while numerical simulations might be challenged when considering molecules which affect the neurological system and thus can potentially impact the zebrafish body balance for example. We underline biological processes can *a priori* be emphasized by numerical simulations provided it falls into our mathematical modeling framework. Finally, the last limitation might be the counter-part to the main contribution developed in this work: we focused on performing point-to-point numerical simulations regarding the corresponding experiments to obtain specific and realistic swimming performances. Nevertheless, we need to perform several numerical simulations to obtain global insights regarding a

particular experiment, although numerical simulations would probably require a less amount of data and therefore fewer animals to be enrolled in experimental studies.

6.3 Application perspectives

As discussed previously, further investigations would be required to reduce the numerical error between the numerical solution and experiments. Nevertheless, the numerical error might be acceptable to analyze specific biological effects provided the phenotype can be well-modeled by our numerical approach. In that case, several numerical simulations might be needed to properly reproduce the biological response. However, we underlined an effort test might additionally be conducted to emphasize altered escape responses. Besides, multiple toxicological and pharmacological treatments could be experimented to challenge swimming energetic performances either naturally or combined with an effort test. In definitive, underlying biological processes and mechanisms mainly produce the experimentally observed body kinematics. Each body deformation is directly causing the final swimming kinematics. That is why, the resulting swimming kinematics can be directly computed through CFD provided the 3D experimental body deformations are known. Thus, we developed a numerical approach capable of simulating escape responses associated with specific body kinematics. However, biological experiments require several zebrafish eleuthero-embryos and multiple escape responses. In consequence, many numerical simulations are needed to assess overall biological effects. As a result, numerical simulations cannot currently be used for predicting biological experiments as experimental body deformations are required. Eventually, intermediate body deformations could be interpolated between two drug or chemical concentrations for instance, by using Wasserstein barycenters or artificial intelligence. Indeed, we can find in the recent literature a few studies which used deep reinforcement learning and neural networks to estimate specific body deformations [42, 63, 64, 126]. As biologists can conduct experiments and provide a massive amount of data of high spatial and temporal resolutions, those tools might be efficient to help in predicting particular deformations. As for Wasserstein barycenters, they could be used to provide a mean escape response among multiple experimental escape responses, in the sense of Wasserstein measures. Such an approach could be extended towards a better representation of experimental phenotypes and swimming performances. Excepted for those particular developments, numerical simulations might be used for prediction in the framework of effort tests. Indeed, we might be able to predict how a chemical compound-exposed zebrafish eleuthero-embryo should perform into high-viscosity flows after a few calibration experiments as conducted in this work, with no drug or chemical exposure. Afterward, only the actual chemical-exposed experiments are required to run effort tests based on chemical-exposed zebrafish eleuthero-embryos. Therefore, we can reduce significantly the number of experiments needed. Besides, we already demonstrated

in this work, only a few experimental locomotions might be required to obtain fine numerical results in terms of energetic performances, whereas experimental studies often require a larger amount of experimental kinematics. In addition, this work strongly suggests the number of experimental conditions such as chemical concentrations might be reduced due to the potential of finer comparisons by using energetic performances or effort tests. Therefore, our experiment-driven numerical approach can be considered as an important tool to support biologists in experiment-driven studies while reducing the number of animals used in experiments and replacing experimental conclusions with numerical estimations of swimming performances. Finally, note we particularly focused on modeling zebrafish swimming and experimental escape phenotypes although this work can be extended to any study which aims to simulate the flow around an immersed body moving with an experimental deformation, based on experimental imaging data. In particular, our work might bring specific insights into engineering and biomimicry applications.

Appendix A

Optimal transportation

A.1 Optimal transportation generalization: how to integrate rigid motions

In [6], the authors have developed an algorithm to solve the space-time problem (3.10). In that regard, Benamou and Brenier transformed (ρ, v) into conservative variables $(\rho, m = \rho v)$ to define a saddle-point problem *via* a Lagrangian formulation:

$$\inf_{(\rho,m)} \sup_{\Phi} \mathcal{L}(\Phi, \rho, m) \tag{A.1}$$

where the Lagrangian $\mathcal{L}$ is constructed from the Lagrange multiplier Φ and the constraints (3.7) and (3.9) as:

$$\mathcal{L}(\Phi, \rho, m) = \int_{t=0}^{1} \int_{\Omega} \frac{|m|^2}{2\rho} - \partial_t \Phi \rho - \nabla \Phi \cdot m - \int_{\Omega} \Phi(\cdot, 0)\rho_0 - \Phi(\cdot, 1)\rho_1 \tag{A.2}$$

Therefore, solving the saddle-point problem (A.1) is equivalent to the dynamic formulation (3.10) with its corresponding constraints (3.7), (3.9). To that end, Benamou and Brenier first rewrote problem (A.1) with new notations:

$$\sup_{\mu=(\rho,m)} \inf_{(\Phi,q)} F(q) + G(\Phi) + < \mu, \nabla_{t,x}\Phi - q > \tag{A.3}$$

In this saddle-point problem, the variable $\mu = (\rho, m)$ can be considered as the Lagrange multiplier of the constraint $\nabla_{t,x}\Phi - q$. Finally, Benamou and Brenier defined the so-called augmented Lagrangian:

$$\mathcal{L}_r(\Phi, q, \mu) = F(q) + G(\Phi) + < \mu, \nabla_{t,x} - q > + \frac{r}{2} < \nabla_{t,x}\Phi - q, \nabla_{t,x}\Phi - q > \tag{A.4}$$

which solves an equivalent saddle-point problem to Eq. (A.3). Note previous Lagrangian formulations are generally used to solve problems of the form: $\min_{x\in\mathcal{H}} F(Kx) + G(x)$ where K is a bounded linear operator, $\mathcal{H}$ an Hilbertian space and F, G simple functions.

The final saddle-point problem is then solved *via* iterations of the Uzawa algorithm with respect to Φ^n, q^n and μ^n. The resulting algorithm namely the so-called Benamou-Brenier (BB) algorithm is described in details in [6]. Basically, the algorithm corresponds to an Alternating Direction Method of Multiplier (ADMM) (see review [96]). Performed on few standard transportation tests, the BB algorithm showed remarkable results (Fig. A.1).

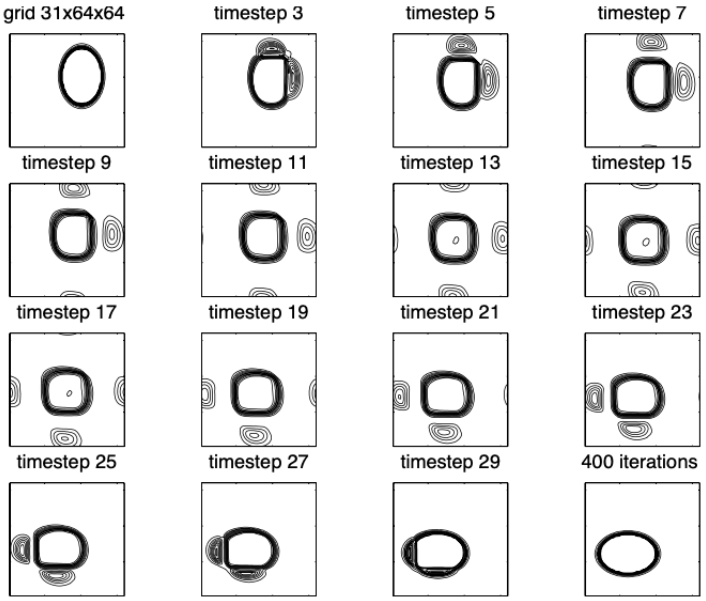


FIGURE A.1: **Illustration of a computational optimal transportation solving by using the BB algorithm, from [6].** The example is based on the transportation of vertical ellipse to an horizontal ellipse, using a $32 \times 64 \times 64$ grid, 400 iterations and periodic boundary conditions. In particular, we can see the impacts of periodic boundary conditions: densities are splitting across the boundary if the path is shorter.

The BB algorithm is particularly interesting as it is based on an Augmented Lagrangian formulation which can easily be modified. That is why, most of the generalized algorithms for solving optimal transportation problems are focusing on the BB algorithm.

The first issue with optimal solutions comes from boundary conditions. Previously, we illustrated (Fig. A.1) how periodic boundary conditions can affect the optimal path. Such considerations are well-known and can easily be implemented accordingly in numerical methods as in Lagrangian and Eulerian algorithms we used [18]. Furthermore, we previously described how an optimal path necessarily follows a straight line trajectory. This basic property can become a crucial limitation as rotational motions cannot be approximated *via* an optimal transportation computation. To visualize the modeling

issue, let consider two initially vertical gaussians and two final horizontal gaussian distributions. Although a gaussian rotation is expected, the optimal motion is actually attained by splitting the mass density of each gaussian distributions (see Fig. A.2).

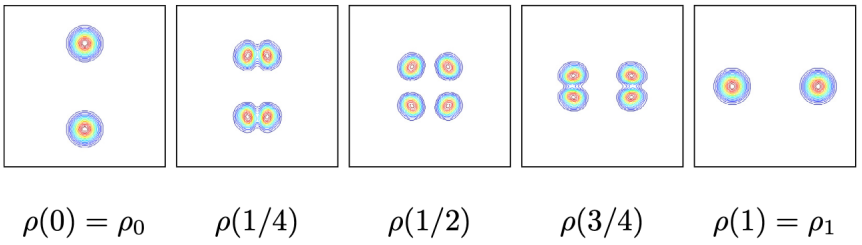


FIGURE A.2: **Representation of the well-known mass-splitting phenomenon with two gaussian distribution, from [54].** The two gaussian densities are initially vertical and we target a $\pi/2$-rotation. However, there is no specific constraint which imposes the shape preservation. Note three intermediate state are represented between the initial and final configurations.

Indeed, the mass density can be split accordingly as long as the whole mass distribution follows an optimal path in the sense of the $\mathcal{L}^2$-Wasserstein metric. Therefore, basic optimal transportation algorithm such as BB algorithm or Lagrangian and Eulerian algorithms we used [18], cannot prevent spurious mass movements especially when considering complex and real-world mass density such as zebrafish larval body kinematics. As a consequence, we investigated into potential optimal transportation spurious effects and the most visible and striking result we can highlight resided in the mass transfer from the tail tip to the head, occurring during the C-bend motion. However, further unexpected behaviors might occur with no possibility to properly isolate and emphasize. In the literature, we can find generalized optimal transportation algorithms which impose physical constraints such as rigidity or incompressibility conditions [19, 54]. The goal of such penalizations or flow constraints is to preserve the compactness and local properties of the mass densities. For instance, standard optimal transportation validation tests such as the rotation of two gaussian distributions demonstrated a considerable improvement regarding the conservation of physics (see Fig. A.3).

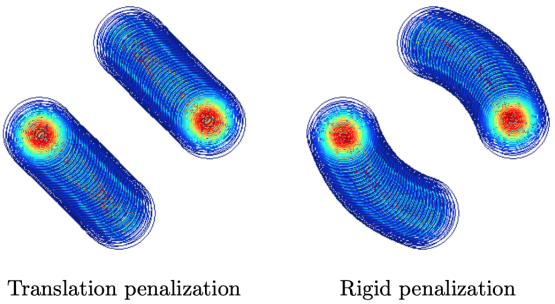


FIGURE A.3: **Physic-like properties are imposed to recover proper rotation and mass movements, from [54].** The rotation of gaussians shows the impact of translation (left) and rigid (right) penalization: rigid constraints embed a rigid rotational motion.

On the whole, we do know spurious affects are occurring in the computed deformation velocity field, due to specific properties of optimal transportation explained in this section. However, we do not know the exact limitations of standard optimal transportation with our specific application: extracting a velocity field from experimental body kinematics of a living animal. In that regard, we investigated into the integration of incompressible and above all rigid penalization techniques within our real-world application. On the one hand, Lagrangian and Eulerian algorithms were not particularly designed for simple modeling enhancements and would require more investigations. On the other hand, the BB algorithm on which these penalizations have previously been carried out was found unable to provide satisfying results with such a large amount of high-resolution data. In particular, we desired to perform BB iterations to assess the importance of spurious optimal transportation phenomena, but the lack of a well-defined convergence criterion and straightforward characterization of numerical parameters for complex, high-resolution data and real-like flows emphasized the BB algorithm was not well-suited.

A.2 A regularized formulation: the Sinkhorn algorithm

Finally, new advances developed particularly fast and efficient optimal transportation computational methods, by relaxing the classic MKP (3.4). In what follows, I will essentially and repidly present all the framework detailed in [109]. Thus, we refer the reader to [109] and recent reviews [100], [102] for all the theoretical and implementation details. Regularized optimal transportation methods have been initially developed to define a smooth similarity measure between probability measures and fast to compute, notably for machine-learning [103], inverse problem and imagery fields. However, this method demonstrated great capabilities in shape interpolation and image reconstruction techniques [109]. In this *PhD* work, regularized optimal transportation has been used precisely to reconstruct a full 3D volume of a zebrafish eleuthero-embryo. The present approach was found very efficient to rapidly reconstruct a massive amount of density distributions. The shape interpolation we considered consisted in finding the density distribution at a certain barycenter of two regularized-shape distributions. As explained previously, linear interpolations cannot be implemented with no proper Lagrangian representation of cross-section shapes. Therefore, regularized optimal transportation is particularly suitable. The main idea consists in regularizing the MKP by adding an Entropic term which makes the optimization problem strictly convex.

First, let define the Entropy $\mathcal{H}(\pi)$ as:

$$\mathcal{H}(\pi) = -\int_{X\times X} \pi(x,y)\ln\pi(x,y)dxdy \tag{A.5}$$

The entropy-regularized $\mathcal{L}^2$-Wasserstein distance is then defined as:

$$\mathcal{W}_{2,\gamma}^2(\mu,\nu) = \inf_{\pi} \int_{X\times X} \|y-x\|^2 \pi(x,y)dxdy - \gamma\mathcal{H}(\pi) \tag{A.6}$$

where $\gamma > 0$ is a regularization parameter. In the limit $\gamma = 0$, problem (A.6) is equivalent to (3.5) while the solution of the suboptimal problem is smoothed according to the choice of γ. As a consequence, we do not consider the exact optimal transportation problem and instead, we regularize MKP by introducing an Entropic penalty. Besides, let define the so-called Kullback-Leibler (KL) divergence:

$$KL(\pi|\mathcal{K}) = \int_{X\times X} \pi(x,y)(\ln \frac{\pi(x,y)}{\mathcal{K}(x,y)} - 1)dxdy \tag{A.7}$$

Then, the diffusion-based approximation of regularized Wasserstein metric can be written as:

$$\mathcal{W}_{2,\mathcal{K}_\gamma}^2(\mu,\nu) = \gamma(1 + \min_{\pi} KL(\pi|\mathcal{K}_\gamma)) \tag{A.8}$$

when considering the distance to a gaussian kernel $\mathcal{K}_\gamma$:

$$\mathcal{K}_\gamma(x,y) = e^{-\frac{d(x,y)^2}{\gamma}} \tag{A.9}$$

Let consider λ_1, λ_2, two Lagrange multiplier of the two constraints on π. We can demonstrate the optimality condition are necessarily given by: $\pi(x,y) = \mathcal{K}_\gamma(x,y)e^{\frac{\lambda_1(x)+\lambda_2(y)}{\gamma}}$. The unique Lagrange multipliers are thus given by:

$$\lambda_1 = -\gamma \log(\int_\Omega \mathcal{K}_\gamma(x,y)e^{\frac{\lambda_2(y)}{\gamma}} d\nu(y))$$
$$\lambda_2 = -\gamma \log(\int_\Omega \mathcal{K}_\gamma(x,y)e^{\frac{\lambda_1(x)}{\gamma}} d\mu(x))$$

which are solved *via* the so-called Sinkhorn algorithm. Let discretize the previous continuous framework. The transportation plan π is necessarily of the form:

$$\pi = D(\mathbf{u})\mathcal{K}_\gamma D(\mathbf{v}) \tag{A.10}$$

where D denotes a diagonal matrix, $\mathbf{u}, \mathbf{v}$ are unique vectors which verify the constraints:

$$\sum_j \pi_{ij} = \sum_j u_i \mathcal{K}_{\gamma,ij} v_j = \mu \tag{A.11}$$
$$\sum_i \pi_{ij} = \sum_i u_i \mathcal{K}_{\gamma,ij} v_j = \nu \tag{A.12}$$

The so-called Sinkhorn algorithm computes $\mathbf{u}$ and $\mathbf{v}$ *via* iterative projections:

$$u_i \leftarrow \mu/(\mathcal{K}_\gamma \mathbf{v})_i$$
$$v_j \leftarrow \nu/(\mathcal{K}_\gamma^T \mathbf{u})_j$$

The Sinkhorn algorithm converges towards the unique optimal solution. The main advantage for using the gaussian kernel $\mathcal{K}_\gamma$ is to faster the numerical computation of Sinkhorn iterations. Indeed, the application of a gaussian filter can easily be approximated by convolution operations. In particular, the product $\mathcal{K}_\gamma \mathbf{u}$ can be reduced to its most significant terms *via* convolutions. That is why, we used Kernel convolutions to compute the products $\mathcal{K}_\gamma \mathbf{v}$ and $\mathcal{K}_\gamma^T \mathbf{u}$ which enabled the computation of the regularized optimal transportation problem on large data-sets.

Finally, let define the so-called Wasserstein barycenter problem:

$$\inf_{\mu} \sum \lambda_k \mathcal{W}_2^2(\mu, \mu_k) \tag{A.13}$$

where the (λ_k) are the weights corresponding to a specific barycenter of densities (μ_k). Therefore, problem (A.13) defines a barycenter between probability measures (μ_k) in the sense of $\mathcal{L}^2$-Wasserstein metric. By considering the previous discrete Sinkhorn iterations, the barycentric distribution is given by:

$$(\mu_\lambda)_i = \Pi_k (u_i (\mathcal{K}_\gamma \mathbf{v})_i)^{\lambda_k} \tag{A.14}$$

In [109], the authors have illustrated Wasserstein barycenter problems with several examples by computing intermediate shapes. By definition, Wasserstein barycenters are much more well-suited to shape interpolation than linear interpolation (see Fig. A.5).

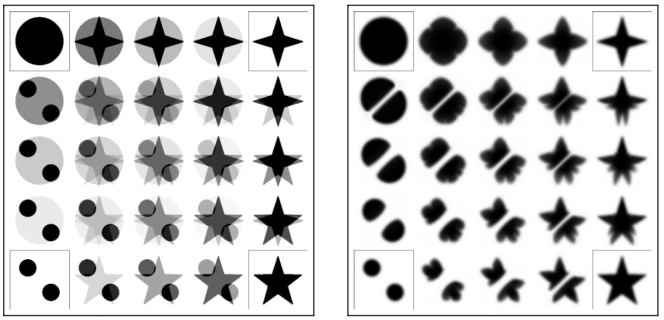


FIGURE A.4: **Comparison between linear interpolation and Wasserstein barycenters, from [109].** Initially, four distributions are provided as input: a disk, two gaussians, a 5-pointed and a 4-pointed stars. Then all intermediate shapes are computed either with linear interpolation (left) or Wasserstein barycenter (right). We can see the difference between a classic interpolation and a Mc Cann's interpolation: the mass density is actually transported, according to an optimal path.

In facts, the strict interpolation problem is more straightforward than Wasserstein barycenter as only two densities are needed as input to compute the intermediate distributions along the optimal transportation path.

On the whole, barycenter problems can lead to various and multiple applications especially after defining a smooth metric to measure similarities. We refer to reviews

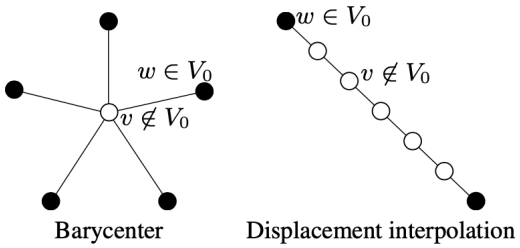


FIGURE A.5: **Illustrations of two different applications: barycenter of multiple distributions and interpolation between two distributions, from [109].** Obviously, the interpolation problem (right) is a particular case of the general barycenter question (left). We can see how the Wasserstein metric and particularly regularized Wasserstein metric is useful to measure similarities between *a priori* different distributions. Note how smoothed the generated shapes are.

[100], [102] which provide a wide spectrum of applications. In our case, we only focus on Wasserstein interpolations with a two-dimensional shape interpolation application. In the case of two mass density, the time interpolation in the sense of the dynamic point of view, is given by:

$$\log(\mu_t)_i = t \log((u_1)_i(\mathcal{K}_\gamma \mathbf{v_1})_i)) + (1-t)\log((u_2)_i(\mathcal{K}_\gamma \mathbf{v_2})_i)) \tag{A.15}$$

In the discrete context of Sinkhorn iterations, time interpolation μ_t (A.15) is computed at the end of Sinkhorn iterations *via* the convolution of the gaussian kernel:

$$\log(\mu_t)_i = \sum_k \lambda_k \log((u_k)_i(\mathcal{K}_\gamma \mathbf{v_k})_i)) \tag{A.16}$$

Bibliography

[1] N. Akashi, J.-I. Kushibiki, and F. Dunn. Measurements of acoustic properties of aqueous dextran solutions in the VHF/UHF range. *Ultrasonics*, 38(9):915–919, sep 2000.

[2] S. Angenent, S. Haker, and A. Tannenbaum. Minimizing flows for the monge–kantorovich problem. *SIAM Journal on Mathematical Analysis*, 35(1):61–97, jan 2003.

[3] P. Angot, C.-H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81(4):497–520, Feb. 1999.

[4] D. S. Barrett, M. S. Triantafyllou, D. K. P. Yue, M. A. Grosenbaugh, and M. J. Wolfgang. Drag reduction in fish-like locomotion. *Journal of Fluid Mechanics*, 392:183–212, Aug. 1999.

[5] D. N. Beal, F. S. Hover, M. S. Triantafyllou, J. C. Liao, and G. V. Lauder. Passive propulsion in vortex wakes. *Journal of Fluid Mechanics*, 549(-1):385, Feb. 2006.

[6] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, Jan. 2000.

[7] J.-D. Benamou, F. Collino, and J.-M. Mirebeau. Monotone and consistent discretization of the monge-ampère operator. *Mathematics of Computation*, 85(302):2743–2775, mar 2016.

[8] M. Bergmann. *Contributions à la simulation numérique en mécanique des fluides et à la réduction de modèle.* Habilitation à diriger des recherches, université de Bordeaux, June 2014.

[9] M. Bergmann, J. Hovnanian, and A. Iollo. An Accurate Cartesian Method for Incompressible Flows with Moving Boundaries. *Communications in Computational Physics*, 15(05):1266–1290, May 2014.

[10] M. Bergmann and A. Iollo. Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, 230(2):329–348, Jan. 2011.

[11] M. Bergmann and A. Iollo. Bioinspired swimming simulations. *Journal of Computational Physics*, 323:310–321, Oct. 2016.

[12] M. Bergmann, A. Iollo, and R. Mittal. Effect of caudal fin flexibility on the propulsive efficiency of a fish-like swimmer. *Bioinspiration & Biomimetics*, 9(4):046001, Sept. 2014.

[13] I. Borazjani. Simulations of Unsteady Aquatic Locomotion: From Unsteadiness in Straight-Line Swimming to Fast-Starts. *Integrative and Comparative Biology*, 55(4):740–752, Oct. 2015.

[14] I. Borazjani and F. Sotiropoulos. Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology*, 211(10):1541–1558, May 2008.

[15] I. Borazjani and F. Sotiropoulos. Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology*, 212(4):576–592, Feb. 2009.

[16] I. Borazjani and F. Sotiropoulos. On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming. *Journal of Experimental Biology*, 213(1):89–107, Jan. 2010.

[17] I. Borazjani, F. Sotiropoulos, E. D. Tytell, and G. V. Lauder. Hydrodynamics of the bluegill sunfish c-start escape response: three-dimensional simulations and comparison with experimental data. *Journal of Experimental Biology*, 215(4):671–684, feb 2012.

[18] A. Bouharguane, A. Iollo, and L. Weynans. Numerical solution of the Monge–Kantorovich problem by density lift-up continuation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1577–1592, Nov. 2015.

[19] A. Bouharguane, E. Maitre, E. Oudet, and N. Papadakis. Multiphysics optimal transportation and image analysis. working paper or preprint, Oct. 2012.

[20] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, jun 1991.

[21] S. A. Budick and D. M. O'Malley. Locomotor repertoire of the larval zebrafish: swimming, turning and prey capture. *The Journal of Experimental Biology*, 203(Pt 17):2565–2579, Sept. 2000.

[22] H. A. Burgess and M. Granato. Sensorimotor Gating in Larval Zebrafish. *Journal of Neuroscience*, 27(18):4984–4994, May 2007.

[23] S. Butail and D. A. Paley. Three-dimensional reconstruction of the fast-start swimming kinematics of densely schooling fish. *Journal of The Royal Society Interface*, 9(66):77–88, Jan. 2012.

[24] n. Carling, n. Williams, and n. Bowtell. Self-propelled anguilliform swimming: simultaneous solution of the two-dimensional navier-stokes equations and Newton's laws of motion. *The Journal of Experimental Biology*, 201 (Pt 23):3143–3166, Dec. 1998.

[25] J. Cheng, T. J. Pedley, and J. D. Altringham. A continuous dynamic beam model for swimming fish. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1371):981–997, jun 1998.

[26] A. J. Chorin. The numerical solution of the navier-stokes equations for an incompressible fluid. *Bull. Amer. Math. Soc.*, 73(6):928–931, 11 1967.

[27] O. M. Curet, I. K. AlAli, M. A. MacIver, and N. A. Patankar. A versatile implicit iterative approach for fully resolved simulation of self-propulsion. *Computer Methods in Applied Mechanics and Engineering*, 199(37-40):2417–2424, Aug. 2010.

[28] N. Danos. Locomotor Development of Zebrafish (Danio Rerio) Under Novel Hydrodynamic Conditions: viscosity effects on turning development. *Journal of Experimental Zoology Part A: Ecological Genetics and Physiology*, 317(2):117–126, Feb. 2012.

[29] N. Danos and G. V. Lauder. The ontogeny of fin function during routine turns in zebrafish Danio rerio. *Journal of Experimental Biology*, 210(19):3374–3386, Oct. 2007.

[30] N. Danos and G. V. Lauder. Challenging zebrafish escape responses by increasing water viscosity. *Journal of Experimental Biology*, 215(11):1854–1862, June 2012.

[31] n. Domenici and n. Blake. The kinematics and performance of fish fast-start swimming. *The Journal of Experimental Biology*, 200(Pt 8):1165–1178, 1997.

[32] C. Eloy. Optimal Strouhal number for swimming animals. *Journal of Fluids and Structures*, 30:205–218, Apr. 2012.

[33] C. Eloy and L. Schouveiler. Optimisation of two-dimensional undulatory swimming at high Reynolds number. *International Journal of Non-Linear Mechanics*, 46(4):568–576, May 2011.

[34] C. J. Esposito, J. L. Tangorra, B. E. Flammang, and G. V. Lauder. A robotic fish caudal fin: effects of stiffness and motor program on locomotor performance. *Journal of Experimental Biology*, 215(1):56–67, Jan. 2012.

[35] M. Faria, N. Garcia-Reyero, F. Padrós, P. J. Babin, D. Sebastián, J. Cachot, E. Prats, M. Arick, E. Rial, A. Knoll-Gellida, G. Mathieu, F. Le Bihanic, B. L. Escalon, A. Zorzano, A. M. V. M. Soares, and D. Raldúa. Zebrafish models for human acute organophosphorus poisoning. *Scientific Reports*, 5(1):15591, dec 2015.

[36] F. Fish and G. Lauder. Passive and active flow control by swimming fishes and mammals. *Annual Review of Fluid Mechanics*, 38(1):193–224, Jan. 2006.

[37] M. Fleuren, J. L. van Leeuwen, E. M. Quicazan-Rubio, R. P. M. Pieters, B. J. A. Pollux, and C. J. Voesenek. Three-dimensional analysis of the fast-start escape response of the least killifish, *Heterandria formosa*. *The Journal of Experimental Biology*, 221(7):jeb168609, Apr. 2018.

[38] E. Fontaine, D. Lentink, S. Kranenbarg, U. K. Muller, J. L. van Leeuwen, A. H. Barr, and J. W. Burdick. Automated visual tracking for studying the ontogeny of zebrafish swimming. *Journal of Experimental Biology*, 211(8):1305–1316, Apr. 2008.

[39] T. Friedrich, A. M. Lambert, M. A. Masino, and G. B. Downes. Mutation of zebrafish dihydrolipoamide branched-chain transacylase E2 results in motor dysfunction and models maple syrup urine disease. *Disease Models & Mechanisms*, 5(2):248–258, Mar. 2012.

[40] M. Gazzola, M. Argentina, and L. Mahadevan. Scaling macroscopic aquatic locomotion. *Nature Physics*, 10(10):758–761, Oct. 2014.

[41] M. Gazzola, P. Chatelain, W. M. van Rees, and P. Koumoutsakos. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, 230(19):7093–7114, Aug. 2011.

[42] M. Gazzola, B. Hejazialhosseini, and P. Koumoutsakos. Reinforcement Learning and Wavelet Adapted Vortex Methods for Simulations of Self-propelled Swimmers. *SIAM Journal on Scientific Computing*, 36(3):B622–B639, Jan. 2014.

[43] M. Gazzola, W. M. Van Rees, and P. Koumoutsakos. C-start: optimal start of larval fish. *Journal of Fluid Mechanics*, 698:5–18, May 2012.

[44] M. J. Gilbert, T. C. Zerulla, and K. B. Tierney. Zebrafish (Danio rerio) as a model for the study of aging and exercise: Physical ability and trainability decrease with age. *Experimental Gerontology*, 50:106–113, Feb. 2014.

[45] A. Gilmanov and F. Sotiropoulos. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, 207(2):457–492, Aug. 2005.

[46] R. Glowinski, T. Pan, T. Hesla, D. Joseph, and J. Périaux. A Fictitious Domain Approach to the Direct Numerical Simulation of Incompressible Viscous Flow past Moving Rigid Bodies: Application to Particulate Flow. *Journal of Computational Physics*, 169(2):363–426, May 2001.

[47] N. Gravish and G. V. Lauder. Robotics-inspired biology. *The Journal of Experimental Biology*, 221(7):jeb138438, Apr. 2018.

[48] J. GRAY. Studies in animal locomotion. *Journal of Experimental Biology*, 13(2):192–199, 1936.

[49] M. H. Green, O. M. Curet, N. A. Patankar, and M. E. Hale. Fluid dynamics of the larval zebrafish pectoral fin and the role of fin bending in fluid transport. *Bioinspiration & Biomimetics*, 8(1):016002, Dec. 2012.

[50] M. H. Green, R. K. Ho, and M. E. Hale. Movement and function of the pectoral fins of the larval zebrafish (Danio rerio) during slow swimming. *Journal of Experimental Biology*, 214(18):3111–3123, Sept. 2011.

[51] S. Hieber and P. Koumoutsakos. An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers. *Journal of Computational Physics*, 227(19):8636–8654, Oct. 2008.

[52] D. G. C. Hildebrand, M. Cicconet, R. M. Torres, W. Choi, T. M. Quan, J. Moon, A. W. Wetzel, A. Scott Champion, B. J. Graham, O. Randlett, G. S. Plummer, R. Portugues, I. H. Bianco, S. Saalfeld, A. D. Baden, K. Lillaney, R. Burns, J. T. Vogelstein, A. F. Schier, W.-C. A. Lee, W.-K. Jeong, J. W. Lichtman, and F. Engert. Whole-brain serial-section electron microscopy in larval zebrafish. *Nature*, 545(7654):345–349, May 2017.

[53] J. Hovnanian. *Méthode de frontières immergées pour la mécanique des fluides. Application à la simulation de la nage.* Theses, Université Sciences et Technologies - Bordeaux I, Dec. 2012.

[54] R. Hug, E. Maitre, and N. Papadakis. Multi-physics optimal transportation and image interpolation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1671–1692, Nov. 2015.

[55] n. Johnson, n. Cullum, and n. Bennett. Partitioning the effects of temperature and kinematic viscosity on the C-start performance of adult fishes. *The Journal of Experimental Biology*, 201 (Pt 13):2045–2051, July 1998.

[56] B. Kadoch, D. Kolomenskiy, P. Angot, and K. Schneider. A volume penalization method for incompressible flows and scalar advection–diffusion with moving obstacles. *Journal of Computational Physics*, 231(12):4365–4383, 2012.

[57] L. V. Kantorovich. On the translocation of masses. *Journal of Mathematical Sciences*, 133(4):1381–1382, mar 2006.

[58] Y. Katumata, U. K. MüLler, and H. Liu. Computation of Self-Propelled Swimming in Larva Fishes. *Journal of Biomechanical Science and Engineering*, 4(1):54–66, 2009.

[59] S. Kern and P. Koumoutsakos. Simulations of optimized anguilliform swimming. *Journal of Experimental Biology*, 209(24):4841–4857, Dec. 2006.

[60] A. Knoll-Gellida, L. E.Dubrana, L. Bourcier, T. Mercé, G. Gruel, M. Soares, and P. J.Babin. Hyperactivity and seizure induced by tricresyl phosphate are isomer-specific and not linked to phenyl valerate-neuropathy target esterase activity inhibition in zebrafish. *Submitted publication for Toxicological Sciences.*

[61] G. V. Lauder. Experimental Hydrodynamics and Evolution: Function of Median Fins in Ray-finned Fishes. *Integrative and Comparative Biology*, 42(5):1009–1017, Nov. 2002.

[62] G. V. Lauder and E. D. Tytell. Hydrodynamics of Undulatory Propulsion. In *Fish Physiology*, volume 23, pages 425–468. Elsevier, 2005.

[63] G. Li, D. Kolomenskiy, H. Liu, B. Thiria, and R. Godoy-Diana. On the energetics and stability of a minimal fish school. *PLOS ONE*, 14(8):e0215265, Aug. 2019.

[64] G. Li, D. Kolomenskiy, H. Liu, B. Thiria, and R. Godoy-Diana. On the interference of vorticity and pressure fields of a minimal fish school. *Journal of Aero Aqua Bio-mechanisms*, 8(1):27–33, 2019.

[65] G. Li, H. Liu, U. K. Müller, and J. L. van Leeuwen. Swimming Hydrodynamics and Maneuverability in C-Start of Zebrafish Larvae: An Integrated Computational Study. In *ASME-JSME-KSME 2011 Joint Fluids Engineering Conference: Volume 1, Symposia – Parts A, B, C, and D*, pages 2059–2066, Hamamatsu, Japan, 2011. ASME.

[66] G. Li, U. K. Müller, J. L. van Leeuwen, and H. Liu. Fish larvae exploit edge vortices along their dorsal and ventral fin folds to propel themselves. *Journal of The Royal Society Interface*, 13(116):20160068, Mar. 2016.

[67] G. Li, U. K. Muller, J. L. van Leeuwen, and H. Liu. Body dynamics and hydrodynamics of swimming fish larvae: a computational study. *Journal of Experimental Biology*, 215(22):4015–4033, Nov. 2012.

[68] G. Li, U. K. Muller, J. L. van Leeuwen, and H. Liu. Escape trajectories are deflected when fish larvae intercept their own C-start wake. *Journal of The Royal Society Interface*, 11(101):20140848–20140848, Oct. 2014.

[69] N. Li, H. Liu, and Y. Su. Numerical study on the hydrodynamics of thunniform bio-inspired swimming under self-propulsion. *PLOS ONE*, 12(3):e0174740, Mar. 2017.

[70] J. C. Liao. A review of fish swimming mechanics and behaviour in altered flows. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1487):1973–1993, Nov. 2007.

[71] Lighthill. Large-amplitude elongated-body theory of fish locomotion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 179(1055):125–138, nov 1971.

[72] M. J. Lighthill. Note on the swimming of slender fish. *Journal of Fluid Mechanics*, 9(2):305–317, oct 1960.

[73] M. J. Lighthill. Hydromechanics of aquatic animal propulsion. *Annual Review of Fluid Mechanics*, 1(1):413–446, jan 1969.

[74] M. J. J. Lighthill. Aquatic animal propulsion of high hydromechanical efficiency. *Journal of fluid mechanics*, 44(NOV11).

[75] H. Liu, D. Kolomenskiy, T. Nakata, and G. Li. Unsteady bio-fluid dynamics in flying and swimming. *Acta Mechanica Sinica*, 33(4):663–684, Aug. 2017.

[76] H. Liu, R. Wassersug, and K. Kawachi. The three-dimensional hydrodynamics of tadpole locomotion. *The Journal of Experimental Biology*, 200(Pt 22):2807–2819, Nov. 1997.

[77] n. Liu, n. Wassersug, and n. Kawachi. A computational fluid dynamics study of tadpole swimming. *The Journal of Experimental Biology*, 199(Pt 6):1245–1260, 1996.

[78] M. A. MacIver and M. E. Nelson. Body modeling and model-based tracking for neuroethology. *Journal of Neuroscience Methods*, 95(2):133–143, Feb. 2000.

[79] M. J. McHenry. The mechanical scaling of coasting in zebrafish (Danio rerio). *Journal of Experimental Biology*, 208(12):2289–2301, June 2005.

[80] M. J. McHenry and G. V. Lauder. Ontogeny of form and function: Locomotor morphology and drag in zebrafish (Danio rerio). *Journal of Morphology*, 267(9):1099–1109, Sept. 2006.

[81] R. Mittal. Computational modeling in biohydrodynamics: Trends, challenges, and recent advances. *IEEE Journal of Oceanic Engineering*, 29(3):595–604, jul 2004.

[82] R. Mittal, H. Dong, M. Bozkurttas, G. Lauder, and P. Madden. Locomotion with flexible propulsors: II. Computational modeling of pectoral fin swimming in sunfish. *Bioinspiration & Biomimetics*, 1(4):S35–S41, Dec. 2006.

[83] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825–4852, may 2008.

[84] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, jan 2005.

[85] U. K. Müller, J. Smit, E. J. Stamhuis, and J. J. Videler. How the body contributes to the wake in undulatory fish swimming: flow fields of a swimming eel (Anguilla anguilla). *The Journal of Experimental Biology*, 204(Pt 16):2751–2762, Aug. 2001.

[86] U. K. Müller, E. J. Stamhuis, and J. J. Videler. Hydrodynamics of unsteady fish swimming and the effects of body size: comparing the flow fields of fish larvae and adults. *The Journal of Experimental Biology*, 203(Pt 2):193–206, Jan. 2000.

[87] U. K. Müller, E. J. Stamhuis, and J. J. Videler. Riding the waves: the role of the body wave in undulatory fish swimming. *Integrative and Comparative Biology*, 42(5):981–987, Nov. 2002.

[88] G. Monge. *Mémoire sur la théorie des déblais et des remblais.* De l'Imprimerie Royale.

[89] U. K. Muller. Swimming of larval zebrafish: ontogeny of body waves and implications for locomotory development. *Journal of Experimental Biology*, 207(5):853–868, Feb. 2004.

[90] U. K. Muller, J. G. M. van den Boogaart, and J. L. van Leeuwen. Flow patterns of larval fish: undulatory swimming in the intermediate flow regime. *Journal of Experimental Biology*, 211(2):196–205, Jan. 2008.

[91] V. Mwaffo, R. P. Anderson, S. Butail, and M. Porfiri. A jump persistent turning walker to model zebrafish locomotion. *Journal of The Royal Society Interface*, 12(102):20140884–20140884, Jan. 2015.

[92] V. Mwaffo, S. Butail, and M. Porfiri. In-silico experiments of zebrafish behaviour: modeling swimming in three dimensions. *Scientific Reports*, 7:39877, Jan. 2017.

[93] A. Nair, G. Azatian, and M. J. McHenry. The kinematics of directional control in the fast start of zebrafish larvae. *Journal of Experimental Biology*, 218(24):3996–4004, Dec. 2015.

[94] J. C. Nauen and G. V. Lauder. Hydrodynamics of mackerel caudal fin locomotion. page 16.

[95] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[96] N. Papadakis, G. Peyré, and E. Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):212–238, jan 2014.

[97] S. Patankar and D. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787–1806, oct 2020.

[98] N. K. Patel, A. P. Singh Bhalla, and N. A. Patankar. A new constraint-based formulation for hydrodynamically resolved computational neuromechanics of swimming animals. *Journal of Computational Physics*, 375:684–716, Dec. 2018.

[99] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, oct 1972.

[100] G. Peyré and M. Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5):355–607, sep 2019.

[101] G. Russo and P. Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, Sept. 2000.

[102] F. Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and Their Applications*. Springer International Publishing, 2015.

[103] M. A. Schmitz, M. Heitz, N. Bonneel, F. Ngolè, D. Coeurjolly, M. Cuturi, G. Peyré, and J.-L. Starck. Wasserstein Dictionary Learning: Optimal Transport-Based Unsupervised Nonlinear Dictionary Learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, Jan. 2018.

[104] W. W. Schultz. Power Requirements of Swimming: Do New Methods Resolve Old Questions? *Integrative and Comparative Biology*, 42(5):1018–1025, Nov. 2002.

[105] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, feb 1996.

[106] J. A. Sethian and J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Number 3 in Cambridge monographs on applied and computational mathematics. Cambridge University Press, 2nd ed edition, 1999.

[107] M. Sfakiotakis, D. Lane, and J. Davies. Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering*, 24(2):237–252, Apr. 1999.

[108] A. A. Shirgaonkar, M. A. MacIver, and N. A. Patankar. A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion. *Journal of Computational Physics*, 228(7):2366–2390, Apr. 2009.

[109] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional wasserstein distances: efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4):66:1–66, July 2015.

[110] F. Sotiropoulos and X. Yang. Immersed boundary methods for simulating fluid–structure interaction. *Progress in Aerospace Sciences*, 65:1–21, Feb. 2014.

[111] K. M. Tabor, S. A. Bergeron, E. J. Horstick, D. C. Jordan, V. Aho, T. Porkka-Heiskanen, G. Haspel, and H. A. Burgess. Direct activation of the Mauthner cell by electric field pulses drives ultrarapid escape responses. *Journal of Neurophysiology*, 112(4):834–844, Aug. 2014.

[112] J. L. Tangorra, S. N. Davidson, I. W. Hunter, P. G. A. Madden, G. V. Lauder, H. Dong, M. Bozkurttas, and R. Mittal. The development of a biologically inspired propulsor for unmanned underwater vehicles. *IEEE J. Oceanic Eng.*, 32(3):533–550, july 2007.

[113] G. K. Taylor, R. L. Nudds, and A. L. R. Thomas. Flying and swimming animals cruise at a strouhal number tuned for high power efficiency. *Nature*, 425(6959):707–711, Oct. 2003.

[114] A. Thibault, L. Chizat, C. Dossal, and N. Papadakis. Overrelaxed Sinkhorn-Knopp Algorithm for Regularized Optimal Transport. *arXiv:1711.01851 [math]*, Nov. 2017. arXiv: 1711.01851.

[115] A. Tingaud-Sequeira, D. Raldúa, J. Lavie, G. Mathieu, M. Bordier, A. Knoll-Gellida, P. Rambeau, I. Coupry, M. André, E. Malm, C. Möller, S. Andreasson, N. D. Rendtorff, L. Tranebjærg, M. Koenig, D. Lacombe, C. Goizet, and P. J. Babin. Functional validation of ABHD12 mutations in the neurodegenerative disease PHARC. *Neurobiology of Disease*, 98:36–51, feb 2017.

[116] R. Témam. Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires (i). *Archive for Rational Mechanics and Analysis*, 32(2):135–153, jan 1969.

[117] G. Triantafyllou, M. Triantafyllou, and M. Grosenbaugh. Optimal Thrust Development in Oscillating Foils with Application to Fish Propulsion. *Journal of Fluids and Structures*, 7(2):205–224, Feb. 1993.

[118] M. S. Triantafyllou, G. S. Triantafyllou, and D. K. P. Yue. Hydrodynamics of Fishlike Swimming. *Annual Review of Fluid Mechanics*, 32(1):33–53, Jan. 2000.

[119] E. D. Tytell, C.-Y. Hsu, T. L. Williams, A. H. Cohen, and L. J. Fauci. Interactions between internal forces, body stiffness, and fluid environment in a neuromechanical model of lamprey swimming. *Proceedings of the National Academy of Sciences*, 107(46):19832–19837, nov 2010.

[120] E. D. Tytell and G. V. Lauder. The c-start escape response of polypterus senegalus: bilateral muscle activity and variation during stage 1 and 2. *J. Exp. Biol.*, 205:2591–2603, sep 2002.

[121] E. D. Tytell and G. V. Lauder. Hydrodynamics of the escape response in bluegill sunfish, Lepomis macrochirus. *Journal of Experimental Biology*, 211(21):3359–3369, Nov. 2008.

[122] M. I. Uddin and O. M. Curet. Modeling and Control of a Bio-Inspired Underwater Vessel with Undulating-Fin Propulsion. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1–7, Charleston, SC, Oct. 2018. IEEE.

[123] J. G. M. van den Boogaart, M. Muller, and J. W. M. Osse. Structure and function of the median finfold in larval teleosts. *Journal of Experimental Biology*, 215(14):2359–2368, July 2012.

[124] J. L. van Leeuwen, C. J. Voesenek, and U. K. Müller. How body torque and Strouhal number change with swimming speed and developmental stage in larval zebrafish. *Journal of The Royal Society Interface*, 12(110):20150479, Sept. 2015.

[125] W. M. van Rees, M. Gazzola, and P. Koumoutsakos. Optimal shapes for anguilliform swimmers at intermediate Reynolds numbers. *Journal of Fluid Mechanics*, 722, May 2013.

[126] S. Verma, G. Novati, and P. Koumoutsakos. Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proceedings of the National Academy of Sciences*, 115(23):5849–5854, June 2018.

[127] C. Villani. *Topics in optimal transportation.* Number v. 58 in Graduate studies in mathematics. American Mathematical Society, 2003.

[128] C. Villani. *Optimal transport: old and new.* Number 338 in Grundlehren der mathematischen Wissenschaften. Springer, 2009. OCLC: 271643433.

[129] C. J. Voesenek, G. Li, F. T. Muijres, and J. L. van Leeuwen. Experimental–numerical method for calculating bending moments in swimming fish shows that fish larvae control undulatory swimming with simple actuation. *PLOS Biology*, 18(7):e3000462, july 2020.

[130] C. J. Voesenek, F. T. Muijres, and J. L. van Leeuwen. Biomechanics of swimming in developing larval fish. *The Journal of Experimental Biology*, 221(1):jeb149583, Jan. 2018.

[131] C. J. Voesenek, R. P. M. Pieters, and J. L. van Leeuwen. Automated Reconstruction of Three-Dimensional Fish Motion, Forces, and Torques. *PLOS ONE*, 11(1):e0146682, Jan. 2016.

[132] L. Weynans. *Methode particulaire multiniveaux pour la dynamique des gaz, application au calcul d'ecoulements multifluides*. Theses, Université Joseph-Fourier - Grenoble I, Dec. 2006.

[133] L. Weynans and A. Magni. Consistency, accuracy and entropy behaviour of remeshed particle methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(1):57–81, Jan. 2013.

[134] T. Ye, R. Mittal, H. Udaykumar, and W. Shyy. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156(2):209–240, dec 1999.