Modelling of aeroelasticity in large transformations by partitioned coupling : application to large wind turbines

Caroline Le Guern

université de BORDEAUX

THÈSE PRÉSENTÉE
POUR OBTENIR LE GRADE DE

DOCTEURE DE

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE

SPÉCIALITÉ MATHÉMATIQUES

Par Caroline LE GUERN

MODÉLISATION DE L'AÉRO-ÉLASTICITÉ EN GRANDES TRANSFORMATIONS PAR UN COUPLAGE PARTITIONNÉ : APPLICATION AUX GRANDES ÉOLIENNES.

Sous la direction de : Michel BERGMANN

Soutenue le : 06/12/2024

Membres du jury :

M. BERGMANN, Michel, Directeur de recherche, INRIA Bordeaux	Directeur de thèse
M. DEGROOTE, Joris, Full Professor, Ghent University	Rapporteur
M. SØRENSEN, Niels, Full Professor, TU of Denmark	Rapporteur
Mme VIRÉ, Axelle, Full Professor, TU Delft	Examinateur
M. CAPALDO, MATTEO, Docteur, Total Energies	Examinateur
M. IOLLO, Angelo, Professeur, Université de Bordeaux	Président du jury

Membres invités :

M. GUY, Nicolas, Docteur, IFPEN	Encadrant
M. BLONDEL, Frédéric, Docteur, IPFEN	Encadrant

Acknowledgments

This PhD work is the result of a three-year collaboration between IFP Energies Nouvelles (IFPEN) and INRIA of Bordeaux. I would first like to express my gratitude to both institutions for supporting this research and for providing the necessary resources to make this project possible.
I would also like to thank the members of my jury, Mrs. Viré, Mr. Sørensen, Mr. Degroote, and Mr. Iollo, for taking the time to read my manuscript and for their constructive feedback. Their insights and discussions during the defense were invaluable, and I truly appreciate their contribution to this work.
I am also very grateful to my thesis director, Michel Bergmann, for his guidance and valuable insights throughout these three years, as well as for his continued involvement and the time he regularly dedicated to this work. I would like to extend my sincere gratitude to my thesis advisors at IFPEN: Frédéric Blondel, Jean-Lou Pfister, and Nicolas Guy, who have supported me every step of the way, always being there whenever I needed help. Their feedback, advice, and discussions have been incredibly helpful, and I deeply appreciate their patience and encouragement throughout this whole process. A special thanks to Fred, who played a big role in helping me develop (and naming!) the PITCHOU code, now evolving into the new SVEN. Thank you for your patience and for the insightful discussions, as well as for bringing a positive and encouraging atmosphere, even during the more stressful times. Thank you, Jean-Lou, for always sharing your knowledge, I have truly enjoyed learning from you, and your guidance has been a great help in shaping this thesis and keeping it on the right track (and who knows, maybe one day you'll even convince me to linearize some codes). And finally, Nicolas, not only have you brought a lot of knowledge to this work, but you've also played a key role in managing the thesis, keeping everything organized, and being a great source of support, which I truly appreciate. Even with such great advisors, I also need to extend my gratitude to another crucial member of IFPEN: Mr. Gilles Ferrer. Gillou, your coding expertise and all the insightful discussions we've had have been invaluable to this thesis. But beyond that, having you around during these years made the experience all the more enjoyable, and it was truly a pleasure to work with you!
Beyond the scientific contributions, this thesis would not have been the same without the support of my colleagues, friends, and family. First, I would like to thank everyone from the two departments at IFPEN, R172 and R174. It has been a pleasure to have so many great discussions over the years, and I look forward to continuing to work with you and sharing even more coffee breaks! A big thank you to everyone who shared an office with me, whether for a short or long time: Erwan, Nico, Eniko, and Jimmy, thank you for bringing positivity to my daily work and for all the discussions we've had! Mathieu, you also brought a lot of positivity, and I truly appreciated your support, especially during the "grey Parisian days". A special thank you to all the doctoral students at IFPEN whom I had the pleasure of meeting at the start of this thesis and who have been there throughout this journey: Eddy, Jo, Camille, Kyan, Kapio, Vico, you made these three years so much more enjoyable, and I'm so happy to have met you and I can't wait for more soirées together! I also want to thank my friends outside of IFPEN for their support over these three years

and for always being there to listen during the tougher moments: Ilana, Prunellia, Eno, Rémi, Maria, Martin, Abdou and Matis. Finally, to my loved ones: Edou, thank you for being there every step of the way, for your constant support, for making me laugh every day, and for making everything brighter! To my parents and my sister, thank you for always encouraging me, listening to me talk about my thesis, and for the advice, distractions, and support that made these years so much more enjoyable. I am truly grateful to have had you by my side throughout this journey.

Titre : Modélisation de l'aéro-élasticité en grandes transformations par couplage partitionné : application aux grandes éoliennes.

Résumé : L'augmentation de la taille et de la complexité des éoliennes modernes, notamment celles installées en mer, pose des défis importants pour leur modélisation aéroélastique. Ces éoliennes sont caractérisées par des pales longues, élancées et flexibles, qui se déforment sous l'effet de charges aérodynamiques. Cette flexibilité induit des conditions d'écoulement instationnaires dues aux mouvements des pales. Le comportement aérodynamique des éoliennes est souvent modélisé à l'aide de la théorie Blade Element Momentum (BEM), qui est efficace en termes de temps de calcul mais considérée comme une méthode de basse fidélité, nécessitant des corrections empiriques pour prendre en compte les conditions instationnaires. Dans cette thèse, une approche de fidélité intermédiaire, la méthode Free Vortex Wake (FVW), est choisie pour les simulations aéroélastiques. Contrairement à la BEM, la méthode FVW prend directement en compte des phénomènes complexes tels que les écoulements non-uniformes, le comportement instationnaire des éléments de pales, et les interactions entre les pales et le sillage, permettant une représentation plus précise des forces aérodynamiques. Cependant, en raison de ses temps de calcul élevés, la méthode FVW est moins adaptée que la BEM en tant qu'outil de conception pratique. Pour rendre abordable l'utilisation des méthodes FVW dans les simulations aéroélastiques, ce travail analyse différentes techniques de couplage partitionné. La modélisation aéroélastique des éoliennes est ici effectuée en couplant un code structure d'éléments finis (FE) avec un code FVW en utilisant des approches de couplage partitionné telles que le schéma Conventional Serial Staggered (CSS) et une technique de sous-cyclage. Le schéma CSS permet à chaque solveur de résoudre le problème relatif à sa physique de manière indépendante et d'échanger des données à chaque pas de temps, mais il peut devenir contraignant lorsque des discrétisations temporelles différentes sont nécessaires pour les solveurs fluides et structure. En revanche, la technique de sous-cyclage permet à chaque solveur d'utiliser des pas de temps différents, avec un échange de données ne se produisant qu'aux intervalles du plus grand pas de temps, offrant ainsi une approche plus flexible. Ce travail compare ces techniques de couplage pour différents problèmes modèles afin d'évaluer leurs précision, stabilité et coût de calcul dans diverses conditions. Tout d'abord, un problème linéaire simplifié est utilisé, dans lequel une équation d'oscillateur structurel est couplée à la dynamique du sillage modélisée par une équation de Van der Pol. Cela permet une première évaluation des propriétés numériques de chaque méthode de couplage. Ensuite, un problème de type "pitch-heave", plus complexe, est introduit, en utilisant un modèle simplifié par éléments finis pour la structure et un code FVW basé sur la théorie de la ligne portante à discrétisation en filament pour l'aérodynamique. Ce code Python a été spécifiquement implémenté pour cette thèse, offrant un outil simplifié pour modéliser des problèmes aérodynamiques ou d'interactions fluide-structure. Enfin, ces approches de couplage partitionné ont été implémentées dans un outil de modélisation aéro-servo-élastique avancé pour analyser leurs effets sur un modèle d'éolienne de grande échelle. L'analyse montre comment les différentes méthodes de couplage influencent l'efficacité des simula-

tions aéroélastiques, démontrant que le sous-cyclage peut réduire significativement les temps de calcul tout en préservant la précision lors de l'utilisation des méthodes FVW pour la modélisation aérodynamique.

Mots clés : Modélisation, aéroélasticité, éoliennes, couplage partitionné

Title : Modelling of aeroelasticity in large transformations by partitioned coupling: application to large wind turbines.

Abstract: The increasing size and complexity of modern wind turbines, particularly those in offshore settings, present significant challenges for aeroelastic modelling. These turbines are characterised by long and slender blades that are highly flexible, making them sensitive to deformation under varying aerodynamic loads. This flexibility introduces unsteady inflow conditions due to the blades' motions. The aerodynamic behaviour of wind turbines is often modelled using the Blade Element Momentum (BEM) theory, which is computationally efficient but considered as a low-fidelity approach, requiring empirical corrections to account for unsteady conditions. In this thesis, a medium-fidelity approach, the Free Vortex Wake (FVW) method, is chosen for aeroelastic simulations. Unlike BEM, the FVW method directly captures complex phenomena such as non-uniform inflow, unsteady blade-element behaviour, and blade-wake interactions, providing a more accurate representation of the aerodynamic forces. However, due to its computational intensity, the FVW method is not yet suitable as a practical design tool compared to the more efficient BEM approach. To optimise the use of FVW methods in aeroelastic simulations, this work explores different partitioned coupling techniques. Aeroelastic modelling of wind turbines is here achieved by coupling a structural Finite Element (FE) solver with a FVW solver using partitioned time integration coupling approaches like the Conventional Serial Staggered (CSS) scheme and a subcycling technique. The CSS scheme allows each solver to operate independently and exchange data at every time step, but it can become restrictive when different time discretisations are needed for the fluid and structural solvers. In contrast, the subcycling technique allows each solver to use different time steps, with data exchange occurring only at the larger time step intervals, offering a more flexible approach. This work compares these coupling techniques across different model problems to assess their accuracy, stability, and computational cost under various conditions. First, a simplified linear problem is used, where a structural oscillator equation is coupled with wake dynamics modelled by a Van der Pol equation. This allows for an initial evaluation of the numerical properties of each coupling method. Next, a more complex pitch-heave problem is introduced, using a simplified finite element model for the structure and a Python-based, lifting-line filament FVW code for the aerodynamics. This python code was specifically developed for this thesis, providing

a user-friendly solver for testing aerodynamic or fluid-structure interactions problems. Finally, these partitioned approaches were implemented within an advanced aero-servo-elastic modelling tool to analyse their effects on a large-scale wind turbine problem. The analysis reveals how different coupling methods impact the computational efficiency of aeroelastic simulations, demonstrating that subcycling can significantly reduce computation times while preserving accuracy when using FVW methods for aerodynamic modelling.

Keywords : Modelling, aeroelasticity, wind turbines, partitionned coupling

---

Unité de recherche

[Institut de Mathématiques de Bordeaux, INRIA centre de l'Université de Bordeaux, équipe Memphis, 351, cours de la Libération - F 33 405 TALENCE]

Résumé étendu en français

Introduction

La décarbonation du secteur énergétique est un enjeu majeur pour atteindre les objectifs climatiques définis par l'Agence Internationale pour les Énergies Renouvelables (IRENA). D'après l'IRENA, d'ici 2050, l'énergie éolienne pourrait représenter plus d'un tiers de la production mondiale d'électricité, contribuant ainsi de manière significative à la limitation du réchauffement climatique. Pour y parvenir, une expansion massive des capacités éoliennes est nécessaire, avec une augmentation significative des installations tant terrestres qu'offshore. Cette croissance s'accompagne de nombreux défis techniques, parmi lesquels la modélisation aéroélastique des nouveaux modèles d'éoliennes. Des outils numériques à la fois précis et rapides sont indispensables pour répondre aux exigences des nouveaux designs. Les modèles récents d'éoliennes, en particulier les éoliennes flottantes offshore, sont soumis à des conditions de vent instationnaires. Dans ce contexte, les méthodes de modélisation aérodynamique utilisées actuellement, comme la méthode Blade Element Momentum (BEM), montrent leurs limites en termes de précision. La méthode BEM est largement utilisée pour modéliser le comportement aérodynamique des éoliennes en raison de son faible coût en terme de temps de calcul, mais celle-ci repose sur des hypothèses simplificatrices, telles que des conditions d'écoulement stationnaire, la rendant moins adaptée pour la modélisation aérodynamique des nouveaux designs. Dans cette thèse, la méthode Free Vortex Wake (FVW) a été choisie comme alternative à plus haute fidélité pour les simulations aéroélastiques. Contrairement à la BEM, les méthodes FVW permettent de mieux capturer des phénomènes complexes tels que les écoulements non uniformes, le comportement instationnaire des pales, et les interactions entre les pales et le sillage. Cependant, cette méthode reste beaucoup plus coûteuse en termes de temps calculs, et elle n'est pas encore adaptée pour être utilisée comme outil de design standard pour les simulations aéroélastiques. L'objectif principal de cette thèse est donc d'explorer différentes approches de couplage partitionné afin de réduire les temps de calcul des simulations aéroélastiques utilisant une méthode FVW tout en maintenant un haut niveau de précision. Le couplage partitionné permet de coupler plusieurs solveurs, chacun modélisant une physique spécifique (l'aérodynamique et la structure dans le cas aéroélastique), offrant ainsi une plus grande flexibilité pour adapter les codes à de nouveaux modèles. Contrairement aux méthodes monolithiques, où un seul solveur résout simultanément plusieurs physiques, les méthodes partitionnées permettent d'optimiser chaque solveur indépendamment, rendant le processus de simulation plus efficace et adaptable aux évolutions technologiques.

État de l'art et approches

Dans un premier temps, une analyse des différentes méthodes de modélisation utilisées pour la simulation aérodynamique des éoliennes a été réalisée. Parmi les méthodes les plus répandues, la méthode Blade Element Momentum (BEM) se distingue par son efficacité en termes de temps de calcul. Toutefois, cette approche est lim-

itée car elle nécessite l'utilisation de nombreux modèles de correction pour intégrer certains phénomènes physiques. Les méthodes telles que Free Vortex Wake (FVW), Actuator Disk (AD) et Actuator Line (AL) offrent une fidélité intermédiaire, permettant une meilleure représentation des phénomènes physiques tout en restant moins coûteuses en calcul que les méthodes de type Computational Fluid Dynamics (CFD), qui, bien que très précises, présentent de coûts de calculs prohibitifs, surtout lorsqu'elles sont appliquées à des simulations aéroélastiques dans un cadre de design. Dans cette étude, la méthode FVW a été utilisée. Plus précisément, il s'agit de la méthode FVW à ligne portante avec une discrétisation du sillage sous forme de filaments vortex, représentée dans le schéma ci-dessous.

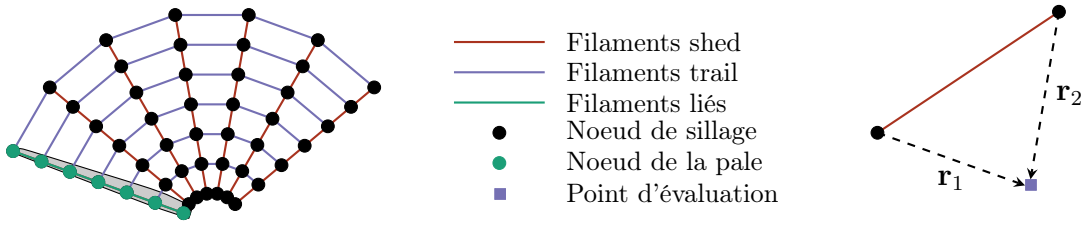


Figure 1: Représentation du sillage et notations pour le calcul d'induction avec $\mathbf{r}_1$, $\mathbf{r}_2$ les vecteurs partant des extrémités d'un filament vers le point d'évaluation. [1]

Dans cette approche, les pales de l'éolienne sont modélisées comme des sources de circulation. Chaque pale est représentée par une ligne portante située au quart de la corde de chaque section, à laquelle est associée une circulation liée, Γ_{bound}. Cette circulation est reliée à la portance générée par la pale via le théorème de Kutta-Joukowsky : $L = \rho|\vec{u}_{eff}|\Gamma_{bound}$. Le sillage est discrétisé en filaments vortex, notamment des filaments shed et trail, auxquels sont attribuées respectivement des circulations shed et trail. Ces circulations peuvent être déterminées grâce au théorème de Kelvin, qui stipule que, dans un fluide non visqueux, la circulation totale d'un système reste constante au cours du temps. Ainsi, toute variation de la circulation liée de la pale est compensée par l'ajout d'une nouvelle rangée de filaments vortex dans le sillage. De plus, pour évaluer la portance, le théorème de Kutta-Joukowsky nécessite de connaître la vitesse effective perçue par un élément de pale. Celle-ci est définie comme la somme des vitesses du vent, des contributions liées à la géométrie de la pale, ainsi que des vitesses induites par les filaments vortex dans le sillage. Ces dernières sont calculées à partir de la loi de Biot-Savart, qui, en discrétisation filamentaire, peut être définie comme : $\vec{u}_{induced}(\vec{x}_p) = \dfrac{\Gamma}{4\pi}\dfrac{(|\vec{r_1}| + |\vec{r_2}|)(\vec{r_1} \times \vec{r_2})}{|\vec{r_1}||\vec{r_2}|(|\vec{r_1}||\vec{r_2}| + \vec{r_1} \cdot \vec{r_2})}$, où $\vec{x}_p$ est un point d'évaluation, et $\vec{r_1}$, $\vec{r_2}$ sont représentés sur le schéma ci-dessus. La loi de Biot-Savart présente un comportement singulier, nécessitant des techniques de désingularisation telles que la méthode de Vatistas et la méthode Offset, dont l'impact sur le calcul des vitesses induites est étudié. Les méthodes vortex dépendent également de plusieurs paramètres clés, comme la discrétisation des filaments vortex, la longueur du sillage, et les approches de désingularisation, tous analysés dans cette étude. Afin de réaliser cette analyse, le solveur C++ CASTOR, basé sur la méthode FVW à ligne portante, a été utilisé. Par ailleurs, un solveur Python reprenant cette même approche a été développé dans le cadre de cette thèse afin de disposer d'un outil plus accessible permettant une meilleure compréhension des

différents paramètres influençant l'évaluation précise des forces aérodynamiques. Pour le calcul aéroélastique le solver DeepLines Wind™ a été utilisé, couplant le code Elements Finis DeepLines au solver FVW CASTOR. Le couplage temporel entre ces deux codes est basé sur une méthode partitionnée, appelée le schéma "Conventional Serial Staggered" (CSS), décrit ci-dessous.

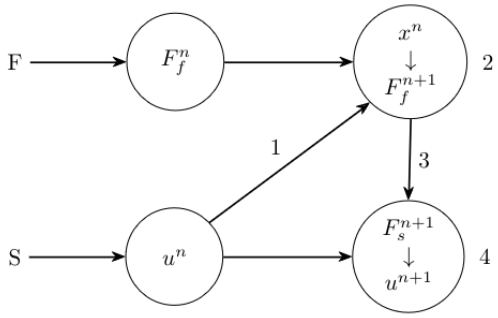


Figure 2: Schéma "Conventional Serial Staggered" (CSS) [2]

Cette méthode permet de séparer l'intégration temporelle des deux physiques. Un pas de temps unique est utilisé pour le couplage, définissant à la fois le pas d'intégration pour le problème fluide, celui du problème structure et le pas de temps où les échanges de données entre les deux solveurs sont effectués. Le couplage aéroélastique suit donc les étapes suivantes: dans un premier temps, les déplacements et vitesses calculés par le solveur structure sont transmis au solveur aérodynamique. Celui-ci résout ensuite le problème fluide, en calculant les forces aérodynamiques induites par les déplacements structuraux. Ces forces sont ensuite renvoyées au solveur structure, qui procède à l'intégration du problème pour calculer les déplacements induits par les forces aérodynamiques. Bien que ce schéma de couplage soit l'un des plus couramment utilisés parmi les méthodes partitionnées, l'utilisation d'un pas de temps unique peut s'avérer restrictive lorsque les différentes physiques impliquent des échelles temporelles caractéristiques distinctes. Pour surmonter cette limitation, un autre schéma couramment utilisé est le sous-cyclage, illustré ci-dessous.

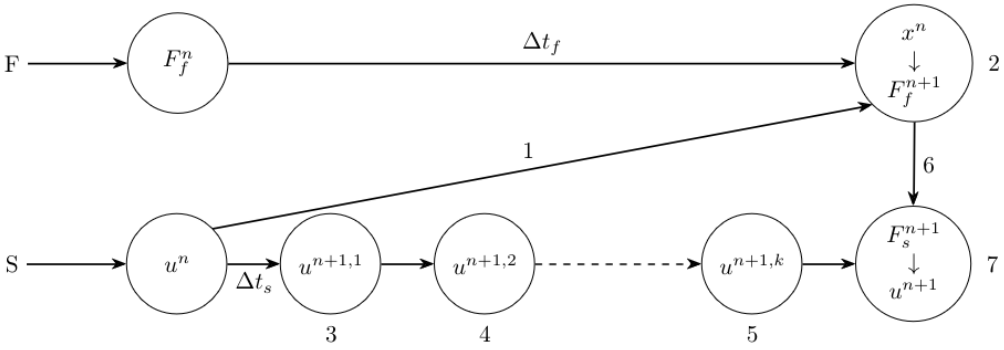


Figure 3: Schéma de sous-cyclage [2]

Ce schéma a été adapté pour son implémentation dans l'outil DeepLines Wind™, afin de contourner la contrainte d'un pas de temps très restrictif imposé par le

contrôleur, qui entraîne une sur-résolution du problème fluide, rendant la méthode FVW particulièrement coûteuse dans les calculs aéroélastiques. L'adaptation du schéma sous-cyclé à l'architecture de DeepLines Wind™ est présentée ci-dessous. L'architecture numérique a été modifiée pour permettre au solveur structure de communiquer avec deux librairies aérodynamiques : une librairie "haute fidélité", représentée par un solveur aérodynamique comme CASTOR et notée "F" dans le schéma, et une librairie "basse fidélité", notée "M" pour méta-modèle, qui agit comme un prédicteur fournissant des données sur les forces aérodynamiques au solveur structure. Ainsi, le solveur aérodynamique peut résoudre le problème fluide avec un pas de temps plus large que celui de la structure, pendant que le méta-modèle, à la fréquence de résolution de la structure, extrapole les forces calculées par le solveur lors des itérations précédentes et les transmets au solveur structure.Les méthodes d'extrapolation étudiées dans cette thèse incluent des régressions constantes, linéaires et quadratiques, mais l'architecture développée offre la flexibilité nécessaire pour l'implémentation de méthodes supplémentaires à l'avenir.

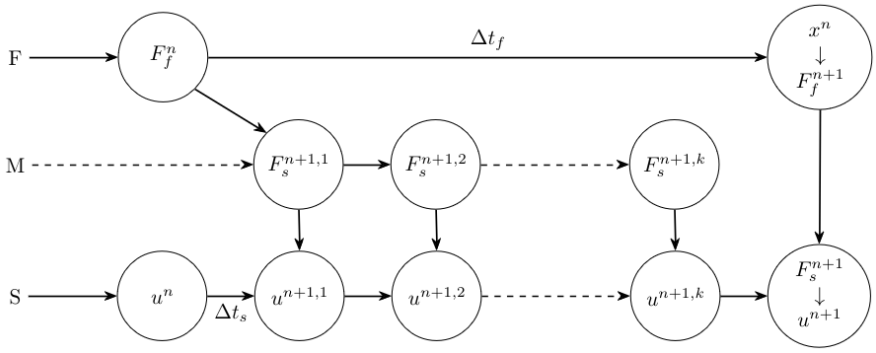


Figure 4: Schéma de sous-cyclage pour DeepLines Wind™ [2]

Les propriétés numériques des différents schémas de couplage ont d'abord été étudiées sur un modèle linéaire simplifié, où un oscillateur structure est couplé à une équation de Van der Pol représentant le fluide. Cela a permis d'observer les effets des différents schémas en termes de stabilité et de précision. Un cas plus réaliste, celui de l'éolienne IEA 15MW, a ensuite été analysé. Bien que ce problème soit plus complexe, certaines observations parallèles ont pu être établies avec le modèle simplifié, même si des liens directs ne peuvent pas être tirés. Dans ce second cas, une analyse de temps de calcul a également été réalisée. Les résultats obtenus pour ces deux modèles sont présentés ci-après.

Résultats

Le modèle linéaire a permis de mettre en évidence plusieurs observations. Tout d'abord, il a servi à caractériser différents types et degrés de couplage, distinguant ceux dominés par des effets d'amortissement de ceux présentant des effets de couplage en fréquence. Ces effets peuvent être faiblement ou fortement marqués selon le type de couplage. Les résultats de l'analyse de stabilité et de précision pour un couplage avec des effets d'amortissement prépondérants sont présentés ci-après.

Cette analyse montre que, sous le schéma CSS, la zone de stabilité est limitée par l'oscillateur ayant le pas de temps critique le plus restrictif. Si le pas de temps critique de l'oscillateur structurel est inférieur à celui de l'oscillateur fluide, le couplage via CSS impose un pas de temps proche de celui de l'oscillateur structure. En revanche, avec le schéma de sous-cyclage, où la structure est sous-cyclée, deux pas de temps peuvent être utilisés : un pas plus fin pour la structure et un pas plus large pour le fluide, qui devient le pas de temps du couplage. Ce schéma étend la zone de stabilité par rapport au CSS, et celle-ci peut même dépasser la zone de stabilité de l'oscillateur le moins restrictif, permettant ainsi l'utilisation de pas de temps plus grands pour le couplage. En ce qui concerne la précision, les erreurs des schémas CSS et sous-cyclage, en fonction du type de régression et du nombre de sous-itérations, ont été calculées par rapport à une solution monolithique. Il a été observé que l'utilisation du sous-cyclage ne dégrade pas l'ordre de précision par rapport au CSS, quel que soit le nombre de sous-cyclages ou le type de régression employé.

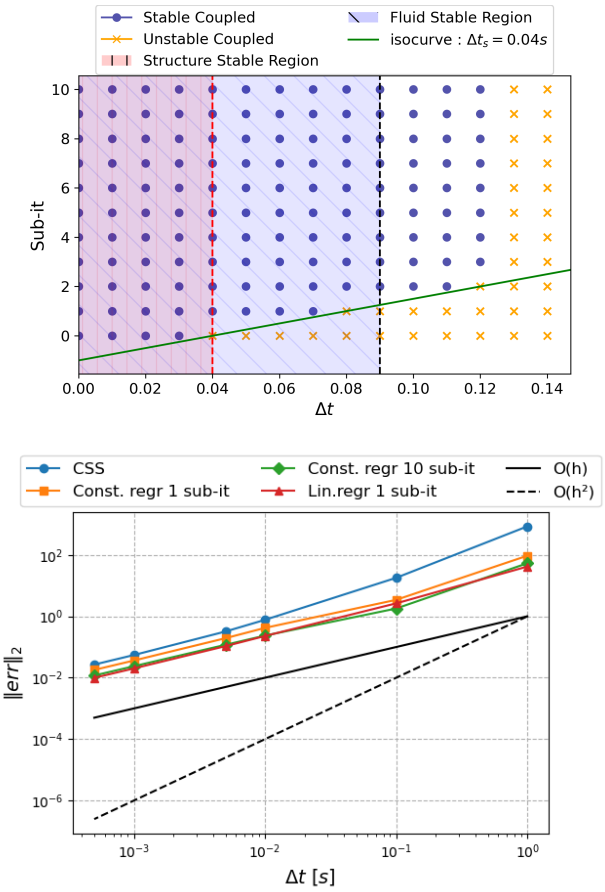


Figure 5: Analyse de stabilité (haut) et de précision (bas) du schéma de couplage CSS et du sous-cyclage.

Par la suite, l'effet du sous-cyclage a été étudié sur le problème aéroélastique plus réaliste de l'éolienne IEA 15MW, soumise à deux conditions de vent. D'abord, un vent constant de 10.6 m/s est imposé, puis un vent turbulent avec une vitesse moyenne de 8 m/s et une intensité de turbulence de 8%. Les résultats montrent

l'évolution des forces aérodynamiques sur une section pour différents nombres de sous-cycles, comparés à la solution de référence obtenue avec un couplage CSS. Les temps de calcul pour chaque simulation sont présentés dans le tableau ci-dessous. Les résultats montrent que l'utilisation du sous-cyclage permet de calculer les forces aérodynamiques avec une précision comparable à celle obtenue avec le couplage CSS. Toutefois, au-delà d'un certain nombre de sous-iterations, correspondant à un pas de temps de couplage critique, la simulation devient instable et divergente. Le sous-cyclage appliqué à la structure permet d'augmenter le pas de temps utilisé pour la résolution aérodynamique, réduisant ainsi la fréquence de résolution du solveur FVW. Cela se traduit directement par une réduction significative des temps de calcul. Une simulation de 200 secondes avec un schéma CSS nécessite 10 jours et 16 heures de temps de calcul, tandis qu'avec 10 itérations de sous-cyclage, ce temps est réduit à 36 minutes, soit une simulation 400 fois plus rapide. Ce gain de performance est directement lié à la méthode FVW : dans le schéma CSS, un filament est généré à chaque pas de temps de couplage, rendant le sillage dense en filaments. Avec le sous-cyclage la fréquence de génération des filaments est réduite, ce qui allège la densité du sillage et réduit considérablement les temps de calcul.

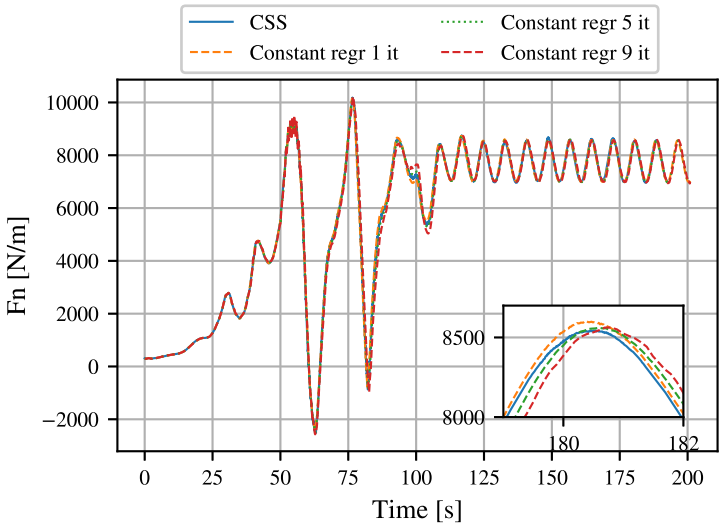


Figure 6: Comparaison des forces aérodynamiques de section pour la méthode CSS et le schéma de sous-cyclage utilisant une régression constante à la position radiale $r = 70, \mathrm{m}$.

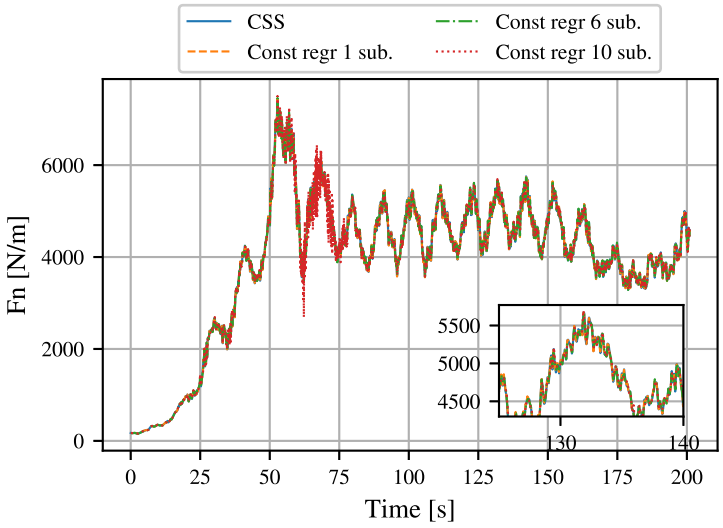


Figure 7: Comparaison des forces aérodynamiques de section pour la méthode CSS et le schéma de sous-cyclage utilisant une régression constante à la position radiale $r = 70,\mathrm{m}$ dans un cas de vent turbulent.

Method	CSS	1 sub-it	2 sub-it	4 sub-it	7 sub-it	9 sub-it
Total comp. time	10d16h	35h06min	11h24min	2h55min	57min	36min

Table 1: Total computational times for CSS and subcycled schemes.

Conclusion et perspectives

Dans cette thèse, l'objectif était de réduire les temps de calcul des simulations aéroélastiques réalisées avec l'outil DeepLines Wind™ tout en maintenant une précision adéquate. La méthode Free Vortex Wake (FVW) a été choisie pour modéliser l'aérodynamique en raison de sa fidélité supérieure à la méthode Blade Element Momentum (BEM), bien qu'elle soit nettement plus coûteuse en termes de calcul lorsqu'elle est utilisée dans des simulations aéroélastiques. Afin de limiter ces coûts de calcul élevés, des stratégies de couplage partitionné, telles que le sous-cyclage, ont été implémentées dans DeepLines Wind™ pour comparer leurs effets au schéma de couplage partitionné initial, le Conventional Serial Staggered (CSS). Le sous-cyclage permet de dissocier les pas de temps entre les solveurs aérodynamique et structure, à l'inverse du schéma CSS qui imposait un unique pas de temps de résolution, entraînant une sur-résolution du problème aérodynamique et des coûts de calcul élevés. L'application du sous-cyclage à une simulation aéroélastique de l'éolienne IEA 15MW, soumise à des conditions de vent constant et turbulent, a montré que le temps de calcul pour une simulation de 200 secondes pouvait être réduit de 10 jours et 16 heures à seulement 36 minutes, soit un gain de performance de plus de 400 fois. Les résultats montrent également qu'une précision comparable à celle obtenue avec le schéma CSS est maintenue. Ces résultats sont prometteurs pour

rendre l'utilisation des méthodes vortex plus viable dans les simulations aéroélastiques destinées au design des futurs éoliennes. Ces travaux pourront être approfondis par l'intégration de ces techniques dans des simulations aéro-hydro-servo-élastiques complètes, ainsi que par l'application simultanée de techniques de "coarsening" de sillage dans les méthodes FVW pour réduire encore davantage les temps de calcul. Par ailleurs, l'implémentation de méthodes de régression plus avancées, telles que les modèles ARIMA ou des techniques de machine learning, pourrait renforcer la stabilité du schéma de couplage et permettre l'utilisation d'un nombre encore plus élevé de sous-itérations, contribuant ainsi à une réduction supplémentaire des temps de calcul.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

As wind turbines evolve, the trend toward larger rotor sizes and offshore configurations is becoming increasingly prominent. However, each new design iteration introduces challenges that require designers to carefully balance performance with cost considerations [8, 9, 10]. One of the key challenges is the significant increase in turbine size, which creates a dual constraint between mass and structural stiffness for a given set of materials. Blades designed to be rigid tend to have higher mass, which increases costs and places greater loads on the system. On the other hand, lighter blades reduce costs and structural loads but introduce greater flexibility, making them more susceptible to aeroelastic instabilities, such as flutter. Studies have shown a link between blade size, specific design choices, and aeroelastic instabilities. For example, Resor et al. (2012) [11] demonstrated that as blade length increase, flutter speeds approach the turbine's operating speeds, leading to lower flutter margins and a higher risk of instability. In smaller turbines, larger flutter margins were found, explaining why this phenomenon has historically not been a significant design constraint. Additionally, Kelley and Paquette (2020) [12] found that while there is not always a direct correlation between increasing blade length and decreasing flutter margin, the reduction in flutter margin is often tied to specific design choices made for larger blades. In this case, design decisions related to the transportability of blades, outlined by Carron et al. (2020) [13], required blades to be more flexible in order to withstand transportation strains without exceeding material limits. This increased flexibility, however, made the blades more susceptible to aeroelastic instabilities. As larger wind turbine models are developed to meet specific design requirements, accurately modelling their aeroelastic behaviour is becoming increasingly important. This requires high-performance simulation tools capable of capturing aeroelastic effects with sufficient precision while maintaining computational efficiency to make them practical for design purposes. In wind turbine engineering, many simulation tools are available to solve single-physics problems (e.g., mechanical, aerodynamics, hydrodynamics) with varying degrees of fidelity. These solvers can be coupled to simulate the interactions between different physics, such as fluid-structure interactions in this case. To accurately capture aeroelastic phenomena for design purposes, it is essential to use solvers with sufficient fidelity while avoiding high computational costs. This can be achieved either by selecting higher fidelity codes, which come with increased computational costs, or by main-

taining these high-fidelity methods and adapting the coupling strategy to minimize the computational burden. The latter approach is adopted in this work, where partitioned coupling techniques are analysed on various model problems to assess their impact on fluid/structure modelling. This first chapter will serve as an introduction to the work, where an overview on wind turbines and aeroelastic modelling will be presented. It begins with an overview of wind turbines, covering their components and providing examples of reference turbines that reflect the latest trends in larger turbine designs. The second section explores different modelling strategies with varying fidelity methods used for single-physics simulations, leading to the presentation of the simulation tool used for aeroelastic analysis in this work, along with the specific methods applied. It also include a presentation of aeroelastic instabilities observed in wind turbines of various sizes, showing the conditions under which these instabilities occur. Finally, the discussion on aerodynamic and structural time scales in the wind turbine framework will set the stage for understanding the objectives of this thesis.

1.1 Wind turbines: an overview

The decarbonization of the energy sector is central to achieving climate goals, as outlined by the International Renewable Energy Agency (IRENA). In its roadmap for a sustainable energy future [3], IRENA proposes pathways to reduce carbon emissions through the rapid deployment of renewable technologies, particularly wind power, highlighting that, by 2050, wind energy could account for over a third of global electricity generation, playing a crucial role in limiting global warming to 1.5°C above pre-industrial levels. In order to meet these targets, a massive expansion in wind capacity is needed, with global annual deployment predictions of onshore and offshore systems presented in fig. 1.1 and fig. 1.2. It is shown that onshore wind capacity will need to triple by 2030 and increase nine-fold by 2050, while offshore wind capacity must grow nearly ten-fold by 2030. This unprecedented growth will require robust technological advancements and significant investments. Efficient design tools and innovations, such as larger turbines, improved foundations for offshore wind, and optimized grid integration, are essential to support the wind industry's rapid scaling and ensure that it can meet future energy demands. This growth in wind capacity necessitates the development of highly efficient design tools to handle the increasing scale and complexity of wind power deployments.

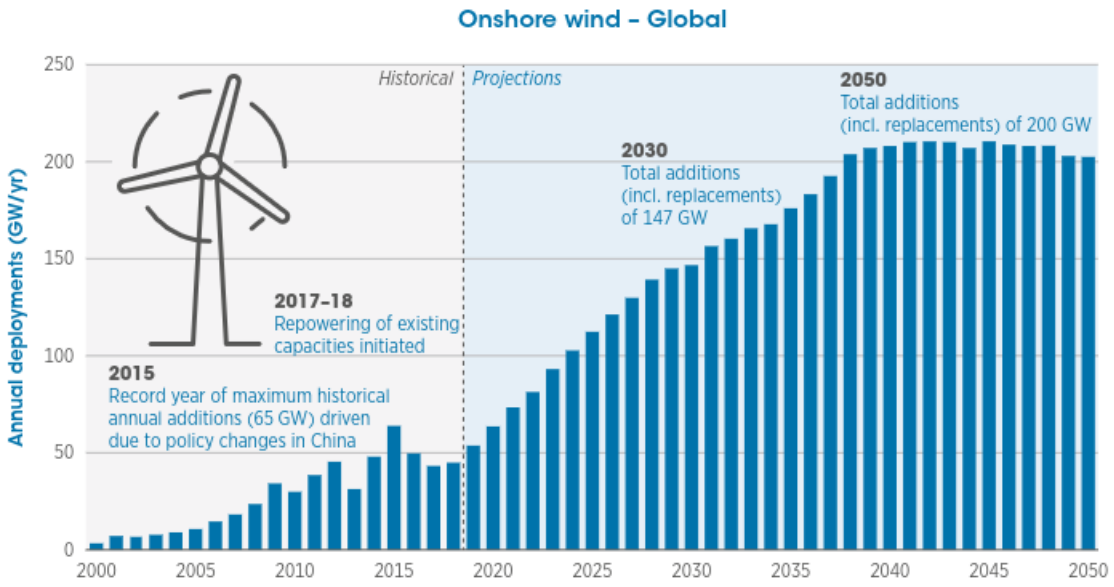


Figure 1.1: Projections of global onshore windpower additions to meet 2050 climate goals, from [3].

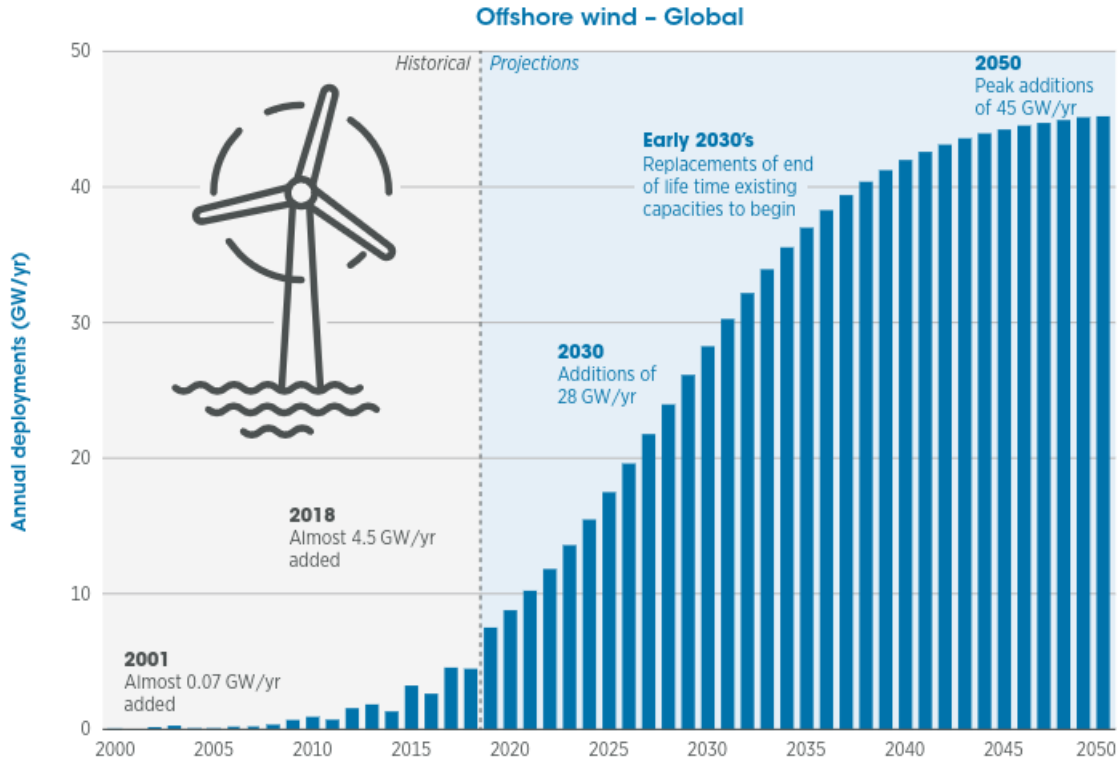


Figure 1.2: Projections of global offshore windpower additions to meet 2050 climate goals, from [3].

1.1.1 HAWT components and characteristic motions

In general terms, the main components of a wind turbine, particularly for Horizontal Axis Wind Turbines (HAWT) [14], can be divided into four key parts which are presented in fig. 1.3. First, the tower provides structural support. It is fixed to the ground in onshore applications and either bottom-fixed or attached to a floating platform in offshore settings. These floating platforms can include various designs, such as spar buoys, tension leg platforms (TLPs), barges, or semi-submersibles [15]. Above the tower is the nacelle, which houses critical components like the gearbox and generator that convert mechanical energy into electrical energy. Attached to the nacelle is the hub on which the blades are connected. The blades rotate the hub in an azimuthal motion at a rotational speed denoted by Ω. The nacelle and rotor can move in two primary directions relative to the tower: a longitudinal (or fore-aft) motion, which refers to movement in the wind direction, and a lateral (or sidewards) motion, which refers to movement transverse to the wind direction. Additionally, the nacelle and rotor can rotate in several ways: in yaw, which corresponds to the rotation of the nacelle around a vertical axis, aligning the rotor with the wind, and tilt and roll, which describe rotational motion of the nacelle and rotor relative to the tower's vertical axis. The blades themselves are subject to several characteristic motions. These include pitch rotation, where the blades rotate about their spanwise axis to optimise aerodynamic performance, and two types of bending: flapwise displacement, where the blades deflect perpendicular to the rotor plane, and edgewise displacement,

where the deflection occurs within the rotor plane. The blades can also undergo torsional deformation, in which the airfoil cross-sections twist around the blade's longitudinal axis. For a more detailed description of wind turbine motions, refer to [16].

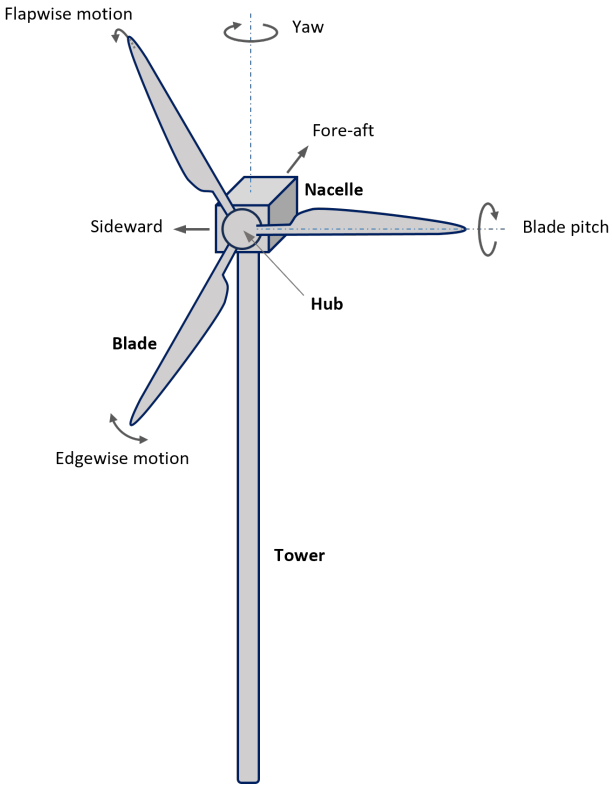


Figure 1.3: Degrees of freedom of wind turbines

1.1.2 Reference wind turbine models

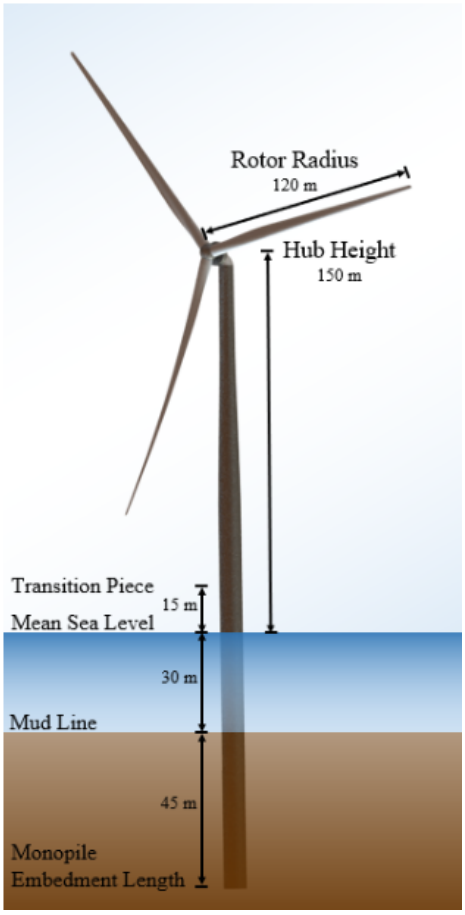


Figure 1.4: IEA 15 MW reference wind turbine from [4]

Reference wind turbines are essential tools for the international research community, enabling the study of standardized, open-access models free from confidentiality constraints. These turbines serve as benchmarks for the development of simulation methodologies, performance evaluations, and aerodynamic and structural analyses. Among the widely used reference turbines are the 5MW wind turbine developed by the National Renewable Energy Laboratory (NREL) [17], and the 10MW model released by the Technical University of Denmark (DTU) [18]. More recently, the IEA 15MW wind turbine [4], illustrated in Figure 1.4, was introduced as part of an international collaboration between NREL, DTU, and the University of Maine. Very recently, the IEA 22MW turbine [19] was released as a collaborative effort between NREL and DTU, further expanding the scope for research in large-scale wind turbine design and analysis.
Each of these reference turbines represents a different stage in the evolution of wind turbine technology, with distinct design features suited to different scales and purposes. The NREL 5MW turbine, with a rotor diameter of 126 meters and a hub height of 90 meters, features blades made of fiberglass, designed for moderate-scale offshore deployment. The DTU 10MW turbine, with a 178-meter rotor and 119-meter hub height, introduces more advanced composite materials such as carbon

fiber for the blades, improving stiffness and reducing weight. The IEA 15MW turbine, with its 117-meter blades and 150-meter tower, represents a state-of-the-art design and was chosen for aeroelastic simulations in this work due to its relevance to large-scale wind turbine research. Most recently, the IEA 22MW turbine, featuring a rotor diameter of 284 meters and a 170-meter tower, is designed for next-generation offshore wind farms. This latest development represents the cutting edge of wind turbine design, providing an essential benchmark for future ultra-large offshore turbines.

1.2 Wind turbines as coupled systems

It was previously observed that efficient numerical methods are needed for design purposes in order to address the growing scale and complexity of wind power deployments. For instance, these methods must be capable of modelling the coupling of fluid and structural dynamics, as well as various aeroelastic phenomena. This section explores the numerical methods used to model wind turbines as fully coupled systems, introducing the DeepLines Wind™ aero-servo-elastic coupling tool.
The section begins by defining aeroelasticity and presenting the aeroelastic phenomena that may occur in large reference wind turbines, along with the conditions under which these instabilities arise. Following this, aerodynamic modelling methods are reviewed, addressing the limitations of each. An overview of structural modelling techniques is also presented, with a discussion on how control systems manage wind turbine performance. The section concludes with an explanation of the coupling process between aerodynamic, structural, and control systems, including the data exchange mechanisms, and introduces the specific aeroelastic solver used in this work.

1.2.1 Aeroelasticity definition

Aeroelasticity as a field emerged primarily in response to challenges in the aeronautical industry, where it played a crucial role in addressing specific design problems related to aircraft performance [20, 21]. The term "aeroelasticity" can be defined as the study of the dynamic behaviour of structures when subjected to aerodynamic flows. The Collar diagram, shown in fig. 1.5, illustrates how it was first defined in [22], describing it as "the study of the mutual interaction that takes place within the triangle of the inertial, elastic and aerodynamic forces acting on structural members exposed to an airstream, and the influence of this study on design".

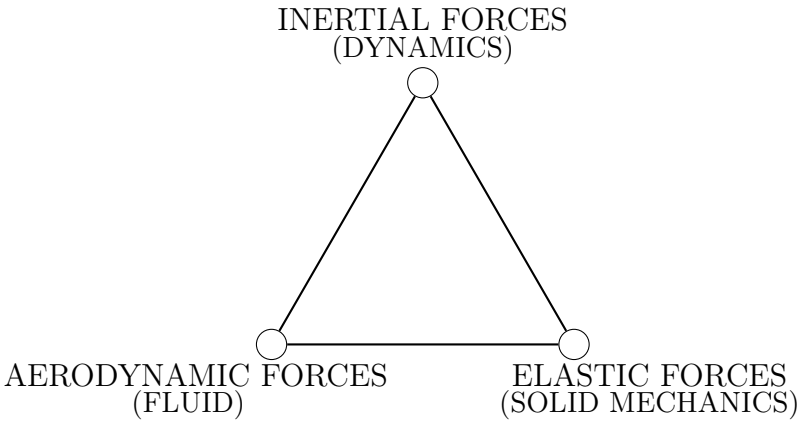


Figure 1.5: Collar diagram

In wind turbines, studying aeroelasticity is essential for understanding how the blades deform under aerodynamic loads. By analysing these deformations, both the turbine's long-term structural durability and its efficiency in capturing wind energy can be evaluated. Aeroelastic analysis also identifies potential instabilities, such as flutter, which may arise under specific operating conditions. This will be the focus of the next section. Addressing these challenges early in the design phase allows for the development of safer, more reliable, and longer-lasting wind turbines, which are crucial for supporting the future growth of wind energy capacity.

1.2.2 Unsteady phenomena related to aeroelasticity in wind turbines

This section will present the aeroelastic instabilities that can affect wind turbines and are critical for certification test validation. The three main instabilities covered are classical flutter (or coupled mode flutter), stall-induced flutter, and vortex-induced vibrations. Reviews of possible instabilities in wind turbines can be found in [23, 16], providing further insight into these phenomena.

1.2.2.1 Classical flutter

Classical flutter is a severe form of aeroelastic instability that can lead to structural failure. It occurs when two structural modes, typically flexion and torsion, couple through the aerodynamic flow, resulting in oscillations once a critical velocity is exceeded. In the NREL 5MW and DTU 10MW wind turbines, flutter was found to occur at speeds well above the operational range. For the NREL 5MW turbine, studies found flutter speeds around 20 rpm in [24, 25] and 24 rpm in [16], nearly double the rated speed of 12.1 rpm. Similarly, Hua et al. [26] identified flutter speeds for the DTU 10MW turbine well above the rated speed, with a flutter margin between 1.4 and 1.7, indicating a significant safety margin. However, for the larger IEA 15MW model, Escalera et al. [27] found that flutter could occur within the operational range, at 6.87 rpm, which is below the rated speed of 7.55 rpm for that

specific model. This suggests that flutter could pose a risk during normal operation in larger wind turbines.

1.2.2.2 Vortex induced vibrations

Vortex-induced vibration (VIV) is an aeroelastic instability induced by flow motion. In VIV, the triggering fluid motion is vortex shedding. When wind turbine blades are subjected to high angles of attack, the flow behind the blades becomes highly separated, potentially forming vortex shedding, which can lead to periodic variations in aerodynamic loads. If the frequency of the vortex shedding approaches the blade's natural frequency and the blade starts moving due to fluctuating loads, the blade may begin to vibrate [28]. Such instabilities often occur when wind turbine blades are subjected to high angles of attack, which can happen under various circumstances. For example, during extreme wind events or maintenance operations, blades may be positioned at standstill with a 90° angle to aerodynamically brake the rotor. In the event of a yaw system failure, the blades may also experience high angles of attack. Several studies have numerically modelled VIV on various reference turbines, including the NREL 5MW turbine [29] and the DTU 10MW configuration [28, 30, 31]. Dose et al. [29] simulated VIV for a 90° inflow angle and wind speeds of 40 m/s and 50 m/s. In the DTU 10MW turbine studies, wind speeds as low as 16 m/s were sufficient to trigger vibrations, with more severe effects observed at speeds above 22 m/s. At these higher velocities, edgewise blade vibrations reached peak-to-peak amplitudes exceeding 5, m, posing a significant risk to blade integrity.

1.2.2.3 Stall induced vibrations

Stall-induced vibrations are another form of aeroelastic instability that can occur in blades at high angles of attack passed a critical wind speed. Unlike classical flutter, which requires multiple degrees of freedom, this instability can develop even when the structural motion is constrained to a single degree of freedom. At high angles of attack, dynamic stall can occur, where the airflow around the blade separates from the surface, resulting in unsteady aerodynamic loads that drive blade oscillations. During these oscillations, the flow alternates between separating and partially or fully reattaching, leading to abrupt changes in lift and drag forces [32]. These abrupt variations in aerodynamic forces can induce in large, unsteady structural motions, which can impose significant, potentially damaging loads on the turbine. These instabilities have been simulated on all three reference wind turbines [33, 34, 35] and are observed at much lower yaw misalignment angles than those for vortex-induced vibrations (VIV), but at higher inflow velocities, typically around 20 ° yaw and wind speeds near 45 m/s.

1.2.3 Aerodynamics modelling

Accurate aerodynamic modelling is essential for predicting wind turbine performance and aerodynamic load distribution on the blades. Several aerodynamic modelling approaches are available [36], each varying in complexity, fidelity, and computational cost. The choice of method depends on the specific goals of the analysis and the available computational resources. This section provides an overview of the primary aerodynamic modelling methods used in wind turbine simulations, ranging from low-fidelity, low-computational-cost models to higher-fidelity, computationally intensive techniques. The Blade Element Momentum (BEM) theory will be introduced first as the simplest and most computationally efficient approach. BEM is widely used in industry but has limitations in handling unsteady or complex flow conditions. Next, the Free Vortex Wake (FVW) method will be discussed, which offers a more detailed representation of wake dynamics by explicitly modelling the vortex structure in the wake. Following this, the Actuator Disk (AD) and Actuator Line (AL) models will be presented, both of which resolve the flow on a mesh using Computational Fluid Dynamics (CFD) techniques. AD simplifies the rotor by representing it as a disk, while AL offers higher accuracy by accounting for the positions of individual blades as the rotor rotates. Finally, the highest fidelity method, blade-resolved CFD, will be introduced. This method captures the full blade geometry and detailed flow features, providing the most precise results but at a significantly higher computational cost.

1.2.3.1 Blade Element Momentum (BEM) method

The Blade Element Momentum (BEM) method [37, 38, 39] is one of the most widely used techniques for wind turbine aerodynamic design due to its low computational cost. This method focuses solely on rotor modelling and combines two primary theories: the conservation of momentum and blade element sectional analysis. The combination of these approaches results in a system of two equations, forming a fixed-point problem. A more detailed development of this can be found in [40], and a brief overview will be presented here.

To introduce the momentum theory, several assumptions and approximations are made. The fluid is assumed to be in a steady state and incompressible. Momentum theory evaluates the variation of fluid momentum as it passes through the rotor disk, as illustrated in fig. 1.6. The expansion of the stream tube requires the conservation of flow rate.

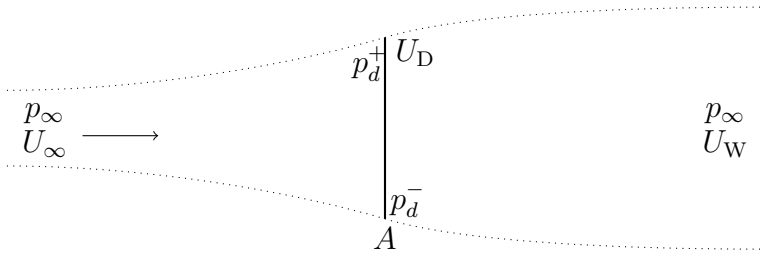


Figure 1.6: Stream tube passing through an actuator disk.

The stream tube can be characterised by the far-field pressure p_∞, the inflow velocity

U_∞, the far wake velocity U_W, the velocity at the rotor disk U_D, and the pressures just upstream and downstream of the rotor disk, p_d^+ and p_d^-. The rotor disk has a circular area $A = \pi R^2$, where R is the disk radius. By applying Bernoulli's theorem, the thrust acting on the rotor can be expressed as:

$$T = \frac{1}{2}\rho A \left(U_\infty^2 - U_W^2\right), \tag{1.1}$$

where ρ is the air density. The velocities at the rotor disk U_D and in the wake U_W are initially unknown. These can be defined by introducing an axial induction factor a, which represents the axial velocity deficit at the rotor. This leads to:

$$U_D = U_\infty(1 - a). \tag{1.2}$$

By considering the balance of momentum between the upstream and downstream sides of the rotor, a new expression for thrust at the rotor disk can be derived:

$$T = \rho A U_D \left(U_\infty - U_W\right), \tag{1.3}$$

From equations (1.1) and (1.3), both the velocity in the wake and the thrust can be expressed in terms of the inflow velocity and the axial induction factor a as:

$$U_W = U_\infty(1 - 2a), \quad T = \frac{1}{2}\rho A U_\infty^2 4a(1 - 2a). \tag{1.4}$$

Similar to the axial induction factor a, the actuator disk, rotating at an angular velocity Ω, induces an angular velocity ω in the flow downstream. These are related by a tangential induction factor a' through $\omega = a'\Omega$. The momentum balance between upstream and downstream flow results in the following expression for the torque Q:

$$Q = \pi r^4 \rho U_\infty \Omega(1 - a)a'. \tag{1.5}$$

To obtain localized forces on the blade, the actuator disk is discretized into annular sections of radius r and size dr, where the area of each section is $dA = 2\pi r dr$. The thrust and torque on each blade element can then be expressed as:

$$dT = 4\pi r \rho U_\infty^2 (1 - a)a \, dr, \tag{1.6}$$

$$dQ = 4\pi r^3 \rho U_\infty \Omega(1 - a)a' \, dr. \tag{1.7}$$

Each blade element, represented as a 2D airfoil (as shown in fig. 1.7), experiences a relative velocity W, and the inflow angle φ is defined as the sum of the angle of attack α and the blade's twist and pitch angles, collectively denoted as β. The lift and drag generated by the blade element can be projected onto the axial and tangential directions to yield new expressions for elemental thrust and torque:

$$dT = \frac{1}{2} n_b \rho c W^2 \left(C_L \cos(\varphi) + C_D \sin(\varphi)\right) dr, \tag{1.8}$$

$$dQ = \frac{1}{2} n_b \rho c W^2 \left(C_L \sin(\varphi) - C_D \cos(\varphi)\right) dr, \tag{1.9}$$

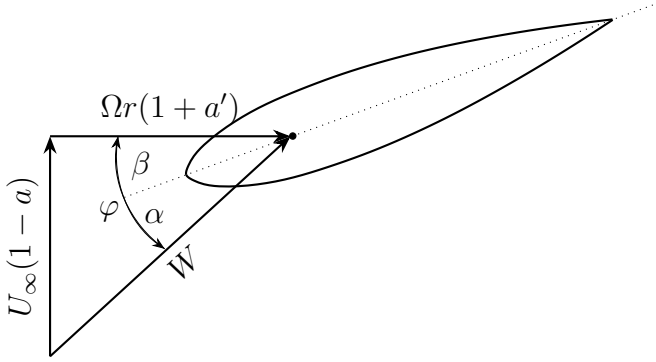


Figure 1.7: Relative velocity W and inflow angle φ of a wind turbine airfoil.

where n_b is the number of blades, c is the blade chord, and C_L and C_D are the lift and drag coefficients.
By equating the momentum theory-derived equations (1.6) and (1.7) with the blade element-derived equations (1.8) and (1.9), the axial and tangential induction factors can be determined as:

$$\begin{cases} a = \frac{1}{2}\left(1 - \sqrt{1 - \sigma\frac{C_L\cos(\varphi)+C_D\sin(\varphi)}{\sin^2(\varphi)}}\right), \\ a' = \frac{\sigma}{4a}\frac{a'(1+a')C_L\sin(\varphi)-C_D\cos(\varphi)}{\sin^2(\varphi)}. \end{cases} \tag{1.10}$$

These induction factors are computed iteratively, starting from an initial guess. A detailed algorithm for this iterative process can be found in [41, 42].
Although the BEM method is widely used for computing aerodynamic forces in wind turbines and is particularly well-suited for steady-state conditions, in more realistic scenarios, correction models are necessary [41]. Among these are models accounting for tip and hub losses [37, 43, 44], which address reduced circulation near the blade tips and hub. The turbulent wake state correction [45, 38] adjusts the axial induction factor under high-load conditions, where increased turbulence affects the wake. Corrections for 3D effects modify the predictions of lift and drag to account for instance for the spanwise flow that is neglected in the 2D blade element theory. Dynamic stall models [46, 47] capture the unsteady flow separation and stall effects that occur during rapid changes in the angle of attack. Dynamic inflow corrections [48, 49] account for the time lag in the wake's response to changing conditions. Lastly, yawed inflow models [50, 51, 52] correct for the effects of non-axial wind flow. A detailed description of these correction models can be found in [40].

1.2.3.2 Free Vortex Wake (FVW) method

Unlike BEM, which relies on empirical corrections to approximate effects like unsteady flow or yaw misalignment, the Free Vortex Wake (FVW) method [53, 32, 54, 55, 56, 57] inherently captures the flow dynamics by directly tracking the vorticity shed by the turbine blades. A brief introduction to the method is provided here, while chapter 3 will offer a more detailed explanation of the FVW method and the solvers used in this work that are based on this approach.
The FVW method is a Lagrangian approach that models the vorticity in the wake

using discrete vortices, which can be represented in several ways, such as vortex particles, vortex filaments, or others [58]. Additionally, the geometry of the wind turbine blades can be discretised using different techniques, including panel methods, lifting surface methods, or lifting line methods [5]. For wind turbine modelling, a common approach is to use a lifting line model combined with a filament-discretised wake, which is the technique adopted in this work and which can be represented by the following scheme from [5]:

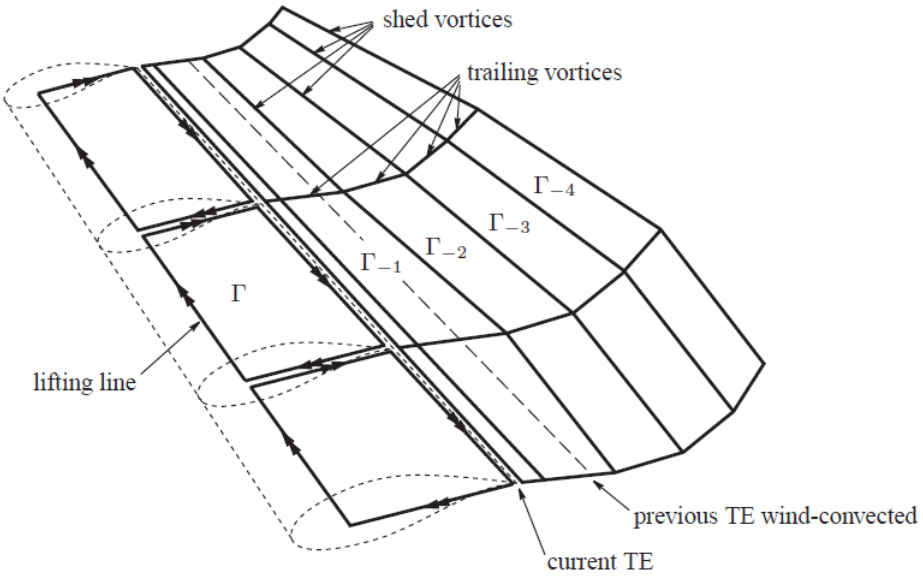


Figure 1.8: Blade and wake discretisations, from [5].

A potential flow is considered here, where vorticity is carried by the vortex filaments. As shown above, the free wake is discretised into shed and trailing filaments, each characterised by their respective circulations Γ, while the lifting line is discretised using straight bound filaments, associated with bound circulation representing the circulation around the airfoil that generates lift. According to Kelvin's theorem [59], which states that total circulation in an inviscid system remains constant over time, any change in bound circulation along the blade must be offset by the shedding of new filaments with equal and opposite circulation into the wake. Shed filaments handle these temporal variations, while trailing filaments account for spanwise variations in circulation along the blade. The position of each filament in the wake is updated by calculating the induced velocity, which represents the velocity contribution from all other filaments, using the Biot-Savart law [53]. This law determines how vortex filaments are convected downstream into the wake, simulating the flow field behind the turbine blades. To compute the lift on the blade, the Kutta-Joukowski theorem is applied, which links the circulation around the airfoil to the lift force generated by it. Similar to the BEM method, the FVW method follows an iterative procedure to calculate aerodynamic forces and update the wake. A detailed explanation of the iterative process is also provided in chapter 3.

1.2.3.3 Actuator disk, actuator line and blade resolved CFD methods

Other methods for wind turbine modelling include the Actuator Disc (AD)[60, 61], Actuator Line (AL)[52] and fully blade-resolved CFD methods [62, 63], listed in increasing order of computational cost and accuracy. These methods differ primarily in how they represent the wind turbine's rotor. Both the actuator disc and actuator line models compute source terms representing aerodynamic forces, which are then applied as body forces in a fluid solver. The key difference lies in how they model the rotor blades, which are not directly meshed in either method [40]. A comprehensive description of both methods is available in [64]. The AD method simplifies the rotor's influence by representing it as body forces distributed uniformly across the rotor disc (for a non-rotating AD method) and non-uniformly (for a rotating AD method). This method is well-suited for studying wake dynamics, though it does not capture detailed blade effects, such as tip loss. The AL method offers a more detailed approach by applying forces directly at the precise positions of the blades within the computational mesh, capturing blade loads more accurately than the AD method. This makes the AL method better suited for studying the aerodynamic loads on the blades themselves, while still being computationally more efficient than fully resolved blade models. For the highest fidelity, fully blade-resolved CFD methods use a body-fitted mesh to model the flow around each blade in great detail. Although these methods are computationally intensive, they allow for the most accurate representation of the flow around the blades, making them essential for detailed studies of blade aerodynamics and complex flow interactions around the rotor.

1.2.4 Structure modelling

In structural modelling, two primary axes define the approach: geometrical and kinematic approximations. These axes govern how a structure is represented and how it is assumed to deform. Geometrical approximations can be divided into several categories based on the dimension and complexity of the model. The most common approaches include beam models, shell models, and 3D finite element methods (FEM), each offering different levels of accuracy and computational cost. Beam models are the most computationally efficient as they reduce the structural system to a one-dimensional line. This is especially suitable for modelling long, slender components like wind turbine blades, where the primary interest is in global structural behaviour rather than localised stress concentrations. These models are ideal for design stages where fast computation is essential. However, due to their simplicity, beam models lack the ability to capture detailed local phenomena. Shell methods [65, 66] provide a middle ground by reducing the structure to two-dimensional surfaces. These models retain more geometric fidelity than beam models while remaining computationally feasible for larger-scale analyses. They allow for a more accurate representation of local structural behaviour without overly increasing simulation time, making them a practical choice for analysing phenomena that beam models cannot capture. For cases that require even higher detail, 3D FEM is used, where the full three-dimensional geometry of the structure is preserved. This method is necessary for capturing local effects with high precision, such as stress concentrations or material failure at critical points. While 3D FEM provides the highest accuracy in terms

of stress and deformation distribution, the computational cost is significantly higher making it generally reserved for localised analysis of specific areas, rather than for full-scale simulations during design phases. Kinematic approximations, which dictate how structural deformations and displacements are handled, range from linear to non-linear models. Linear kinematic models assume small deformations and are often paired with modal decomposition techniques [67, 68, 69], which simplify the system by reducing the number of degrees of freedom. Modal decomposition is especially valuable for dynamic analyses, where a structure's vibrational modes are of interest. By expressing the structural response in terms of its natural modes, linear models allow for efficient computations while retaining the essential characteristics of the dynamic behaviour. This method is often employed in cases where the structure's global behaviour under small perturbations is the focus. For cases involving large deformations, non-linear kinematic models are used. These models account for geometric changes during deformation, providing a more accurate prediction of structural behaviour under more extreme conditions but requiring significantly more computational resources.

1.2.5 Wind turbine control

Two open-source wind turbine controllers used by the research community are the ROSCO controller, developed by NREL [70], and the DTU baseline controller [6]. This section provides an overview of the general actuation methods used in reference wind turbines. The baseline control systems of a wind turbine typically employ a conventional variable-speed and variable blade-pitch configuration. This system can be simplified into two key components: the generator torque controller, which regulates the generator's torque to optimise power capture at different wind speeds, and the rotor-collective blade-pitch controller, which adjusts the pitch angle of the blades collectively to manage the aerodynamic loads and maintain optimal rotor speed. Both the generator torque and blade-pitch controllers typically rely on the generator speed as their primary input. Based on the generator speed, the system generates feedback to adjust the pitch angle and torque, maintaining optimal performance. For the generator torque controller, the generator speed is divided into five distinct operating regions, which define how the control strategy is applied across different wind conditions. These regions are depicted in fig. 1.9.

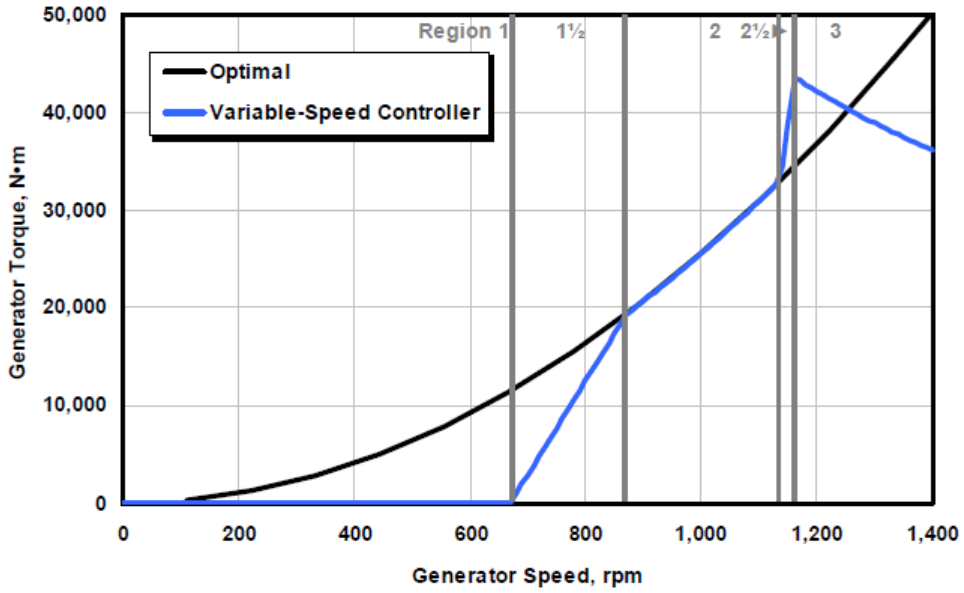


Figure 1.9: Generator speed control curve from [6].

The five operation regions are each tailored to different wind conditions. In region 1, before the cut-in wind speed, the rotor accelerates without generating power. Once the wind reaches the cut-in speed, the turbine enters region 2, where power capture is maximised by keeping the tip-speed ratio constant through a generator torque proportional to the square of the generator speed. As wind speeds rise above the rated level, region 3 maintains constant power by making the generator torque inversely proportional to the generator speed. Regions $1\frac{1}{2}$ and $2\frac{1}{2}$ serve as smooth transitions between these main regions, limiting operational speed and controlling tip speed to ensure safe and efficient turbine operation.

1.2.6 DeepLines Wind™ : a multi-physics modelling tool

A schematic representation of the DeepLines Wind™ solver [7], developed by Principia and used at IFPEN is shown in fig. 1.10, illustrating how it enables aero-hydro-servo-elastic modelling. Fluid-structure coupling is achieved using functionalities from this code, with DeepLines primarily handling structural analysis and AeroDeeP [71] or CASTOR [1] managing the aerodynamic computations.

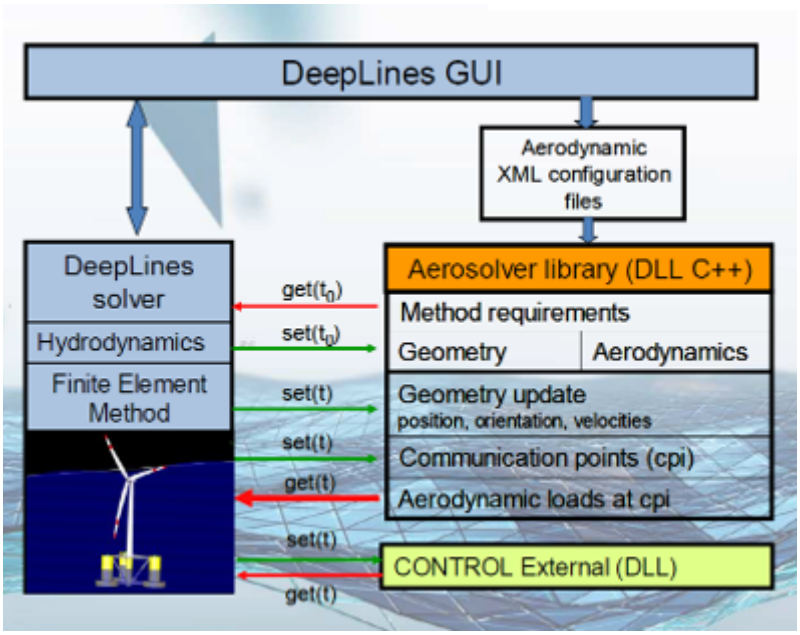


Figure 1.10: DeepLines Wind™ coupling architecture, from [7].

DeepLines handles structural modelling using beam theory, which accounts for large displacements and rotations, while also enabling hydrodynamic modelling through a monolithic coupling method, though the hydrodynamics will not be considered here. AeroDeep and CASTOR, which use Blade Element Momentum theory and Free Vortex Wake methods respectively, compute the aerodynamic forces based on the displacements and velocities provided by the structural solver. A control system is coupled with the structural solver, allowing for blade orientation and generator torque adjustments according to the generator speed provided by DeepLines. Communication between the solvers is illustrated in fig. 1.11, where coupling is achieved using baseline techniques without convergence sub-iterations. Data exchange occurs at each time step, which can lead to higher computational costs if the solvers operate on different time scales. This issue will be addressed in the following section.

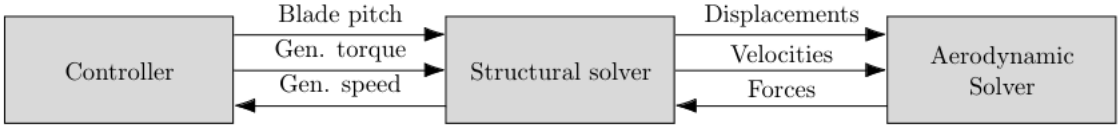


Figure 1.11: Aero-servo-elastic solver data transfer.

1.3 Fluid and structural time scales for a wind turbine in operation

As previously shown, DeepLines Wind™ allows for the coupling of the structural solver to an aerodynamic library, where the numerical architecture imposes the resolution of both the fluid and structural problems using the same time step. However, in some cases, the need for different time steps between these two solvers may have to be considered due to the distinct physical phenomena they resolve. This can be relevant when simulating large wind turbines, where the structural dynamics and the unsteady aerodynamics operate on different characteristic timescales. For the structural problem, a characteristic time can be determined by analysing the natural frequencies of the wind turbine's modes. Depending on how many modes one intends to model, different characteristic times can be considered. For instance, when studying the IEA 15MW wind turbine and focusing on the first modes of interest, a characteristic time can be associated with the highest frequency. Taking in this case the first torsional mode of the blade, which has a frequency of approximately $f = 4.5\,\text{Hz}$, the characteristic time for the structural dynamics is $T_s = \frac{1}{f} \approx 0.2\,\text{s}$. For the fluid problem, on the other hand, a characteristic time can be defined based on unsteady dynamics, where the aerodynamic forces exhibit a lagged response to structural perturbations. This problem was addressed by Theodorsen [72], who derived an analytical solution for the unsteady aerodynamic load on an oscillating airfoil in inviscid, incompressible flow, subject to small disturbances in the frequency domain, and later adapted to the time domain using Wagner's solution [73, 74]. The dynamic aerodynamic response is given in [75] and can be described by:

$$\phi(t^*) = 1 - 0.165e^{b_1 t^*} - 0.355e^{b_2 t^*} = 1 - 0.165e^{-0.0455t^*} - 0.355e^{-0.3t^*}, \tag{1.11}$$

where b_1 and b_2 are coefficients obtained in [76], and $t^* = \frac{2U_t}{c}$ is the non-dimensional time, with U_t representing a velocity term and c a chord length. Taking the decay term of the response function of a first-order system to a perturbation, $e^{\frac{-t}{\tau}}$, which characterizes how fast the system approaches its steady-state value, where t is time and τ is a time constant characteristic of the system's response, and assuming that 95% of the final solution is reached for 3τ, gives $3\tau = \frac{3c}{2U_t b}$. Using the coefficient associated with the slower decaying term $b_1 = 0.0455$, considering the chord length of the IEA 15MW turbine at 75% of the blade's span as $c \approx 3\,\text{m}$, and taking $U_t \approx 60\,\text{m/s}$, the characteristic time for the fluid problem is given by $T_f \approx 1.6\,\text{s}$. In this case, the time step that is used to solve the global probem will be defined by the structure rather than by the aerodynamic problem, in order to be able to capture the wanted structural phenomenon. This shows that depending on the effects that need to be modelled in either physics, different resolution times can be necessary in order to both capture the effects of the most restrictive physics and not over resolve the other problem that might need a coarser time step resolution. In this case, the time step required to solve the global problem will be dictated by the structural dynamics rather than the aerodynamic response, to accurately capture the relevant structural phenomena. However, if the faster aerodynamic response were the main

focus of the analysis, a characteristic time step could be chosen accordingly. This demonstrates that depending on the dominant phenomena needing to be captured, different time steps may be selected. Care must be taken to balance the resolution between the physics being modelled to avoid unnecessarily high computation costs while ensuring the phenomena of interest are accurately captured. This highlights that different time step resolutions may be needed for each domain, to avoid over-resolving one aspect of the physics or neglecting the other. Addressing this issue was a key focus of this work, as will be discussed in the following section.

1.4 Thesis objectives and approaches

This thesis has several key objectives. One of the primary aims is to perform aeroelastic modelling of a large wind turbine under operational conditions, using an alternative to the low-fidelity Blade Element Momentum method. The focus is on employing a Free Vortex Wake method, which offers higher fidelity but comes with significantly greater computational costs, particularly in aeroelastic simulations. The numerical architecture of the aero-servo-elastic solver used in this work, DeepLines Wind™, developed by Principia and used at IFPEN, is designed such that a single time step is applied to both the servo-elastic and fluid resolution. This results in the aerodynamic problem being overly resolved, significantly increasing computational costs compared to BEM. The goal is, therefore, to explore different partitioned coupling approaches to reduce computational time without compromising numerical stability or accuracy.

Initially, the method used to couple the DeepLines and CASTOR codes employs a Conventional Serial Staggered (CSS) scheme, which enforces the single time step resolution. To evaluate the impact of different coupling schemes on aeroelastic simulations using vortex methods, this thesis first reviews a range of coupling techniques, focusing on their numerical properties. A simplified linear fluid/structure coupled model is implemented, where a structural oscillator is coupled with a Van der Pol equation representing the aerodynamic flow. This allows for a detailed assessment of the numerical behaviour of coupling schemes such as CSS and subcycling, comparing their properties.

Following this, the focus shifts to the Free Vortex Wake (FVW) method, which is intended for the aerodynamic modelling component. A detailed description of the method and the numerical codes is presented, including a fully developed GPU-accelerated Free Vortex Wake code in Python. This code is implemented to facilitate ease of use and improve the understanding of key parameters critical for aeroelastic simulations. The analysis highlights how the discretisation of the wake filaments, dictated by the aerodynamic time step, plays a pivotal role in both computational cost and accuracy when coupled with structural models.

In the final part of the thesis, the developed FVW code is coupled with a simplified Finite Element Method (FEM) to examine an initial coupling problem using a single blade model. A new architecture is also implemented in the DeepLines Wind™ code to enable alternative partitioned coupling, with the goal of reducing computational costs while maintaining accuracy and stability in aeroelastic simulations. The results from this new architecture, tested under both constant and turbulent

wind conditions, are compared with those from the original architecture to evaluate whether significant computational savings can be achieved without compromising performance when using a Free Vortex Wake method code for aerodynamic computations.

Chapter 2

Coupling methods for aero-elasticity simulations

In fluid-structure interaction (FSI) problems, two primary approaches are used to couple different physical domains: partitioned [77, 78, 79, 80, 81, 82] and monolithic techniques [83, 84, 85]. This chapter focuses on these coupling methods and their numerical properties, providing a detailed examination of their application and impact on FSI simulations. Monolithic techniques involve solving the entire coupled problem as a single system. While effective in certain scenarios, monolithic approaches are often limited by their computational cost and complexity, making them impractical for large-scale industrial applications. These methods tend to be highly case-dependent, requiring careful tuning and significant computational resources. In contrast, partitioned coupling methods, such as those described by [86] and [87], offer substantial advantages by allowing the sequential resolution of each physical domain using distinct solvers. This approach enables solvers to evolve independently, providing greater flexibility to adapt to varying modelling scenarios. Among the various partitioned techniques, different coupling schemes can significantly influence the efficiency and accuracy of simulations. One of the most commonly employed partitioned methods is the Conventional Serial Staggered (CSS) scheme, as described by [80]. This method enhances flexibility and adaptability by allowing the sequential resolution of physical domains with different solvers. Another widely used technique is subcycling, which allows physical domains to evolve with different time steps. This method is particularly advantageous in FSI problems where aerodynamic and structural dynamics operate on distinct timescales. The numerical properties of partitioned coupling schemes are critical in determining the overall performance of simulations. Understanding these properties is essential for optimizing computational models, particularly in industrial applications where balancing high accuracy with manageable computational costs is necessary. This chapter aims to provide an in-depth examination of the numerical properties of several coupling techniques used in FSI problems. These properties will be derived through the application of coupling schemes to a simplified model of two coupled linear oscillators. The first section of the chapter introduces the physical model and its properties, examining how each parameter influences coupling and defining the types of coupling encountered. Subsequently, the coupling strategies will be presented, and the integration of some of

these strategies into the simplified model will be analysed analytically.
This analysis leads to a detailed examination of the numerical properties of these coupling techniques in the third part, allowing for a comparison of their stability and accuracy. The goal is to establish a solid foundation for understanding the numerical characteristics of such schemes and to draw parallels with how these properties might manifest in more complex, realistic problems. Implementation details of these techniques for larger-scale problems will be discussed at the end of the chapter, with results presented in chapter 4.

2.1 Model problem: two coupled linear oscillators

In this section, the physical problem of two coupled oscillators is introduced as a foundational example to understand the effects of various system parameters. The structural oscillator represents a vibrating structure, while the wake dynamics, modelled by a Van der Pol equation, approximate oscillating forces similar to those generated by a von Karman vortex street. The governing equations for the coupled system are presented first. This setup provides a framework for evaluating how each parameter influences the system, particularly in terms of its frequency and damping characteristics. This analysis leads to the identification and classification of different types and degrees of coupling, highlighting configurations that parallel the wind turbine scenario discussed in chapter 4. The presentation of such model problem will also allow the reader to get a concrete example of how coupling schemes can be integrated into a simplified FSI model, offering insights into how numerical properties can be analytically analysed in a straightforward manner.

2.1.1 Physical problem description

This section discusses the model from [88], which describes the motion of a cylinder oscillating in a uniform flow with free stream velocity U. The dimensional displacement $Y(T)$ (where T is the time variable) of the cylinder is governed by the linear oscillator equation:

$$m\ddot{Y} + r\dot{Y} + hY = S, \tag{2.1}$$

where dots represent a derivative with respect to dimensional time and where m and r account for the mass and damping including contributions from both the structure (m_s and r_s) and the fluid's added mass and damping (m_f and r_f) such that $m = m_s + m_f$ and $r = r_s + r_f$. Parameter h denotes the structural stiffness, and S represents the forcing term due to the oscillating wake.
The dynamics of the wake is itself described by the following Van der Pol equation:

$$\ddot{q} + \epsilon\Omega_f(q^2 - 1)\dot{q} + \Omega_f^2 q = F, \tag{2.2}$$

where Ω_f is the wake's oscillation frequency, q is the dimensionless wake variable and F represents the effects of the cylinder motion on the near wake.
By defining dimensionless time $t = T\Omega_f$ and space coordinate $y = Y/D$ as done in [88], the coupled system can be rewritten as:

$$\ddot{y} + \left(2\xi\delta + \frac{\gamma}{\mu}\right)\dot{y} + \delta^2 y = s, \quad \ddot{q} + \epsilon(q^2 - 1)\dot{q} + q = f. \tag{2.3}$$

Following [89], the nonlinear term in the Van der Pol equation, $\epsilon q^2 \dot{q}$ is neglected while the negative damping term $-\epsilon\dot{q}$ in the wake as well as the cylinder's damping term $\left(2\xi\delta + \frac{\gamma}{\mu}\right)\dot{y}$ are retained in this case. This simplification yields the linear coupled model equations:

$$\ddot{y} + d_s\dot{y} + \delta^2 y = Mq, \quad \ddot{q} + d_a\dot{q} + q = A_0 y + A_1\dot{y}, \tag{2.4}$$

where $d_s = \left(2\xi\delta + \frac{\gamma}{\mu}\right)\dot{y}$ and $d_a = -\epsilon$ represent the structural and fluid damping terms respectively, A_0 and M represent the added coupling stiffnesses and A_1 is an added coupling damping term.

2.1.2 Mode coupling analysis - effects of the system's parameters

The effects of the coupled system's parameters will be studied in this section. For that matter, by setting $X = \begin{bmatrix} y \\ q \end{bmatrix}$, eq. (2.4) will be rewritten as follows:

$$\ddot{X} + \begin{bmatrix} d_s & 0 \\ -A_1 & d_a \end{bmatrix}\dot{X} + \begin{bmatrix} \delta^2 & -M \\ -A_0 & 1 \end{bmatrix} X = 0. \tag{2.5}$$

In order to understand the effects of each parameter, the system's frequency will be studied first by omitting all damping effects. In a second part, the damping terms will be considered, and all of the combined effects will be analysed to help define the different types and degrees of coupling one can encounter which will be the focus of the third part.

2.1.2.1 Without damping terms

In this section, the effects of the structural, aerodynamic and coupling damping parameters (d_s, A_1 and d_a) are supposed to not play a role in the system's resulting frequency evolution and are all set to zero. By omitting the damping terms in eq. (2.5), the coupled system reduces to:

$$\ddot{X} + \begin{bmatrix} \delta^2 & -M \\ -A_0 & 1 \end{bmatrix} X = 0. \tag{2.6}$$

Looking for undamped vibrations of type $X = X_0 e^{i\omega t}$ in the above equation, non-trivial solutions are given by

$$det \begin{bmatrix} \delta^2 - \omega^2 & -M \\ -A_0 & 1 - \omega^2 \end{bmatrix} = 0, \tag{2.7}$$

leading to the following dispersion equation:

$$D(\omega) = \omega^4 - (\delta^2 + 1)\omega^2 + \delta^2 - A_0 M = 0, \tag{2.8}$$

for which solutions are given by:

$$\omega_{\pm}^2 = \frac{\delta^2 + 1 \pm \sqrt{\Delta}}{2}, \tag{2.9}$$

where $\Delta = (\delta^2 + 1)^2 - 4(\delta^2 - A_0 M)$.
For a given value of $A_0 M$, which will further be referred to as the combined coupling stiffness parameter, the system's stability can vary from neutrally stable to unstable depending on the frequency of the structural oscillator, δ. Figure 2.1 illustrates the evolution of the real part of the system's eigenfrequencies relative to δ. In this case, the combined coupling stiffness parameter is chosen to be small compared to the structural oscillator's frequency, specifically $A_0 M = -0.5$.
When $\sqrt{\Delta} < (\delta^2 + 1)$ and $\Delta > 0$, the solutions derived from eq. (2.9) are real, indicating the presence of two neutrally stable modes. These modes are evident on both ends of the graph, where two uncoupled frequencies are observed.
Conversely, when $\sqrt{\Delta} > (\delta^2 + 1)$ or $\Delta > 0$, the system's eigenfrequencies include an imaginary component. This scenario is known as "coupled mode flutter" [89], where two neutral modes merge into a single frequency, as depicted in the center of fig. 2.1 (left). Coupled mode flutter corresponds to a physically unstable case where vibration amplitudes grow exponentially with time. As shown in fig. 2.1 (right), one mode exhibits a negative imaginary part, indicating negative damping and thus instability.

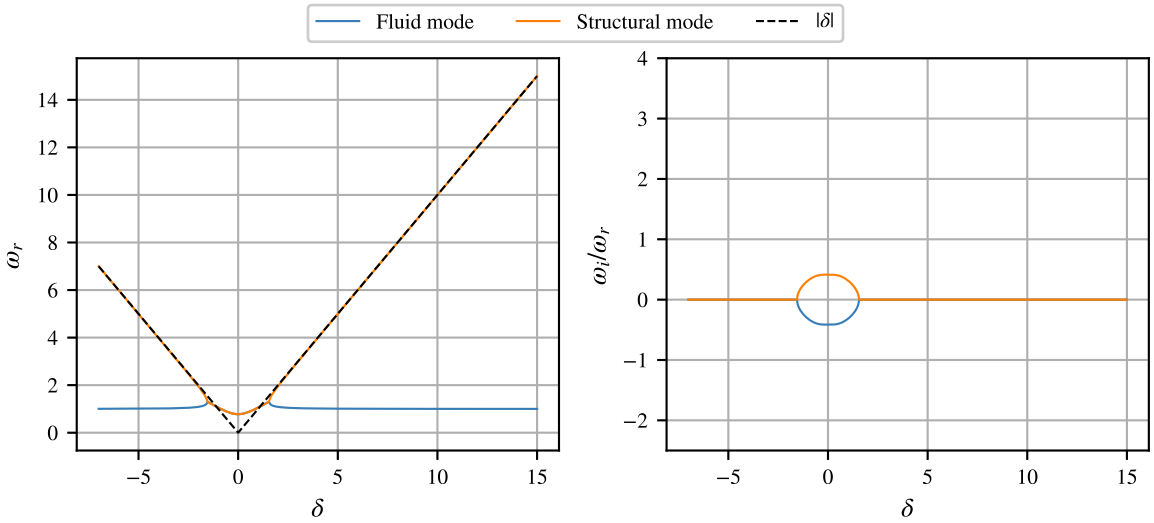


Figure 2.1: Effect of the structural's oscillator frequency on the modes of the coupled system (left) frequency (right) growth rate

Figure 2.2 illustrates the influence of $A_0 M$ on both the real and imaginary parts of the eigenfrequencies. It can be seen that as $A_0 M$ decreases, the region where modes exhibit coupling expands, consequently enlarging the potential zone of instabilities. Additionally, the growth rate increases with decreasing $A_0 M$, further contributing to the potential destabilization of the coupling in this expanded region.

Throughout the remainder of this chapter, oscillation frequencies will be standardized to $\Omega_f = 1$ and $\delta = 10$ (unless stated otherwise), ensuring the presence of two neutrally stable modes regardless of the specific value of $A_0 M$.

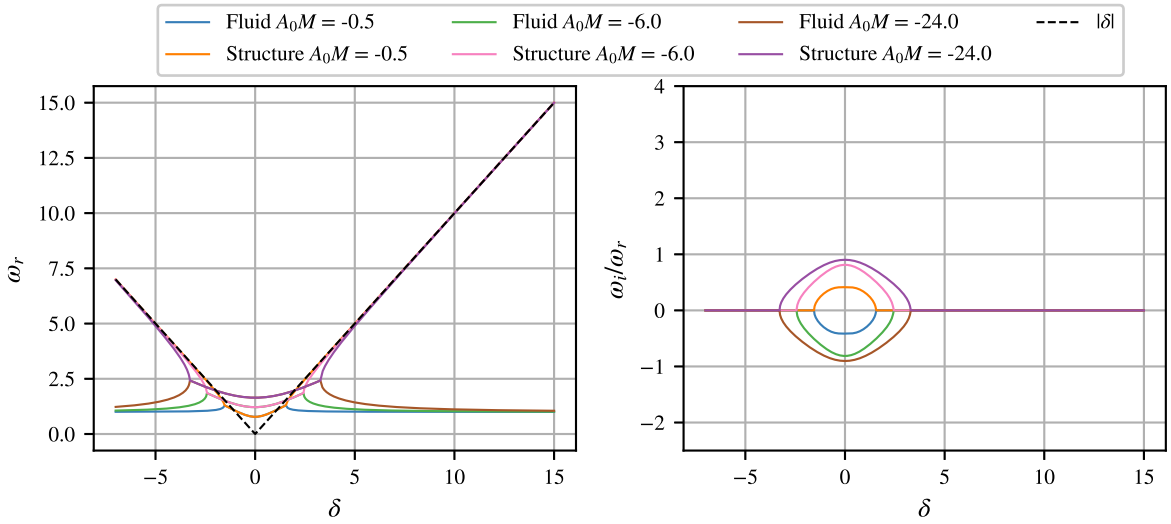


Figure 2.2: Effect of the structural's oscillator frequency on the modes of the coupled system (left) frequency (right) growth rate.

2.1.2.2 With damping terms

Taking into account the damping terms, the modes which were previously considered neutrally stable are now more realistically being either damped or unstable due to these parameters. The associated dispersion relation yields:

$$D(\omega; d_s, d_a, A_1) = (-\omega^2 + \delta^2 + (d_s\omega)i)(-\omega^2 + 1 + (d_a\omega)i) - A_0 M - (A_1 M\omega)i = 0. \tag{2.10}$$

The effects of parameters d_s, d_a, and A_1 on the system's modes are assessed by examining the variations of the eigenfrequencies ω with these parameters, first in the case when they are close to zero. By expanding ω as $\omega = \omega_0 + d_s\omega_s + d_a\omega_a + A_1\omega_c$, where ω_0 satisfies the frequency equation without considering d_s, d_a, and A_1 (i.e., $D(\omega_0; 0, 0, 0) = 0$), the expansion of eq. (2.10) in terms of the (assumed small frequencies) ω_s, ω_a and ω_c yields:

$$
\begin{aligned}
D(\omega_0 + d_s\omega_s + d_a\omega_a + A_1\omega_c; d_s, d_a, A_1) = D(\omega_0; 0, 0, 0) \\
&+ d_s\omega_s \left.\frac{\partial D}{\partial \omega}\right|_{(\omega_0,0,0,0)} + d_a\omega_a \left.\frac{\partial D}{\partial \omega}\right|_{(\omega_0,0,0,0)} \\
&+ A_1\omega_c \left.\frac{\partial D}{\partial \omega}\right|_{(\omega_0,0,0,0)} \\
&+ d_s \left.\frac{\partial D}{\partial d_s}\right|_{(\omega_0,0,0,0)} + d_a \left.\frac{\partial D}{\partial d_a}\right|_{(\omega_0,0,0,0)} \\
&+ A_1 \left.\frac{\partial D}{\partial A_1}\right|_{(\omega_0,0,0,0)} = 0.
\end{aligned}
\tag{2.11}
$$

Since this must be true whatever the small perturbation, this implies that:

$$
d_s\omega_s\frac{\partial D}{\partial \omega} + d_s\frac{\partial D}{\partial d_s} = 0 \tag{2.12}
$$

$$
\Rightarrow \quad \omega_s = -\frac{\partial D/\partial d_s}{\partial D/\partial \omega}, \tag{2.13}
$$

and in a similar manner that:

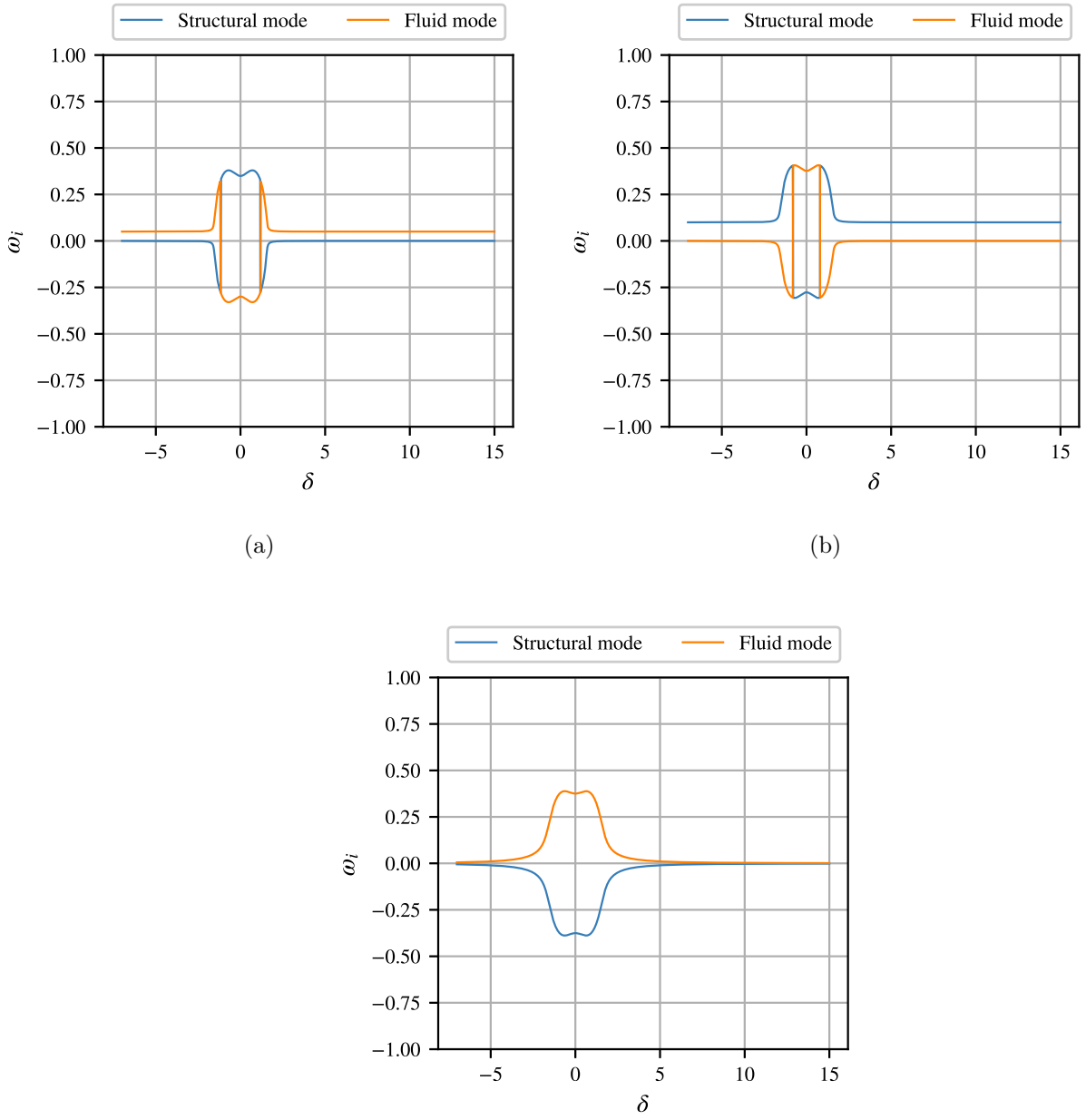
$$
\omega_a = -\frac{\partial D/\partial d_a}{\partial D/\partial \omega}, \quad \omega_c = -\frac{\partial D/\partial A_1}{\partial D/\partial \omega}. \tag{2.14}
$$

The above derivatives can be computed from eq. (2.10) and taken at the reference state ω_0. These can then be re-injected into the first order expansion in order to evaluate the effects of damping parameters on ω:

$$
\omega = \omega_0 + id_s\left[\frac{\omega_0(\omega_0^2 - 1)}{4\omega_0^3 - 2(1+\delta^2)\omega_0}\right] + id_a\left[\frac{\omega_0(\omega_0^2 - \delta^2)}{4\omega_0^3 - 2(1+\delta^2)\omega_0}\right] + iA_1\left[\frac{M\omega_0}{4\omega_0^3 - 2(1+\delta^2)\omega_0}\right]. \tag{2.15}
$$

Equation (2.15) analytically demonstrates that d_a, d_s, and A_1 each exert at first order an influence only on the damping of the coupled system. This influence is further illustrated in fig. 2.3, where eq. (2.10) is solved numerically and where the individual effects of these parameters on the system's damping are depicted, considering $A_0 M = -0.5$.

In the regions where neutrally stable modes were previously identified, fig. 2.3a and fig. 2.3b highlight the positive damping effects of parameters d_a and d_s on both the structural and fluid modes. Conversely, the impact of parameter A_1 appears comparatively minor, resulting in slight damping effects that are rather negligible.



(a)

(b)

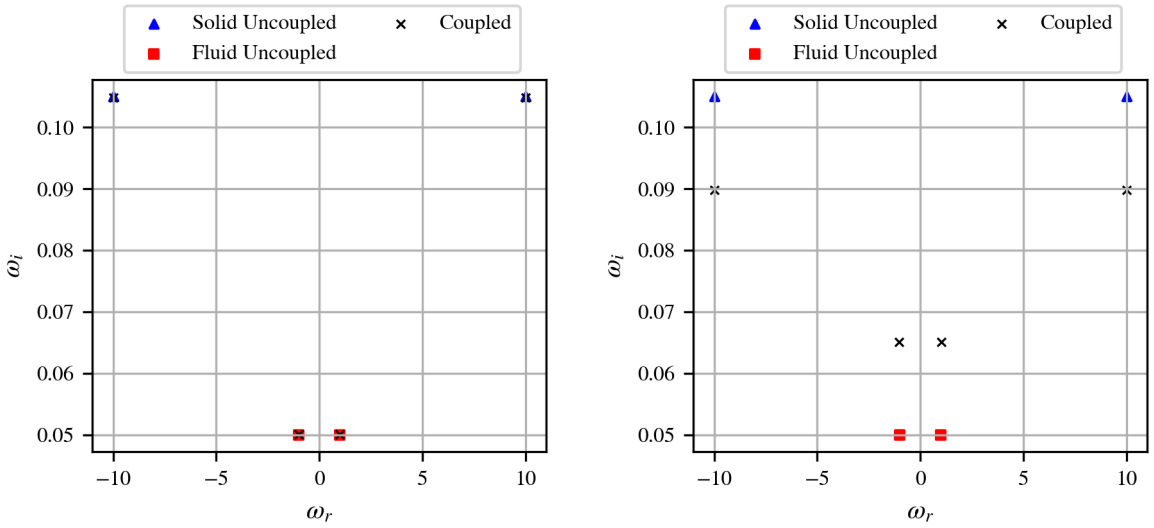
(c)

Figure 2.3: Effects of the individual damping terms d_a, d_s, and A_1 on the coupled system's damping: (a) $d_a = 0.1$, $d_s = 0.0$ and $A_1 = 0.0$, (b) $d_a = 0.0$, $d_s = 0.2$ and $A_1 = 0.0$, (c) $d_a = 0.0$, $d_s = 0.0$ and $A_1 = -0.5$.

2.1.2.3 Types/degrees of coupling

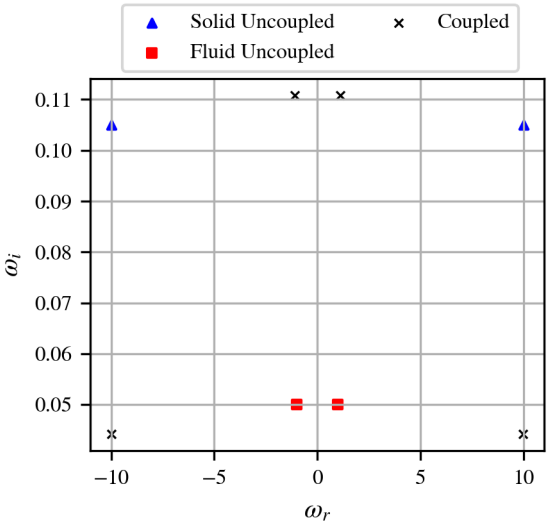
This section analyses different types and degrees of coupling by manipulating the parameters introduced in the previous sections. In fig. 2.4, the structural and fluid damping terms are fixed at $d_s = 0.2$ and $d_a = 0.1$ respectively, while the coupling parameters A_0M and A_1M vary. Specifically, A_0M ranges from -0.06 in fig. 2.4a to -24 in fig. 2.4c, and A_1M ranges from -0.03 to -12 across these figures. The

eigenvalues corresponding to the uncoupled oscillators are compared with those obtained for the coupled fluid/structure system, with the variation occurring only in their imaginary components as the parameters change. This configuration will, in this paper, be referred to as a damping coupling scenario, where the imaginary parts of the eigenvalues determine the system's positive or negative damping. Notably, all cases exhibit positive imaginary parts, indicating a stable and positively damped system.
In the "weakly" coupled scenario depicted in fig. 2.4a, the eigenvalues of the coupled system closely resemble those of the uncoupled system, suggesting minimal interaction between the oscillators. In contrast, in the "medium" and "strongly" coupled cases, imaginary parts of the eigenvalues deviate from those of the uncoupled oscillators. This trend shows the increasingly significant coupling effects between the oscillators.



(a) Weak

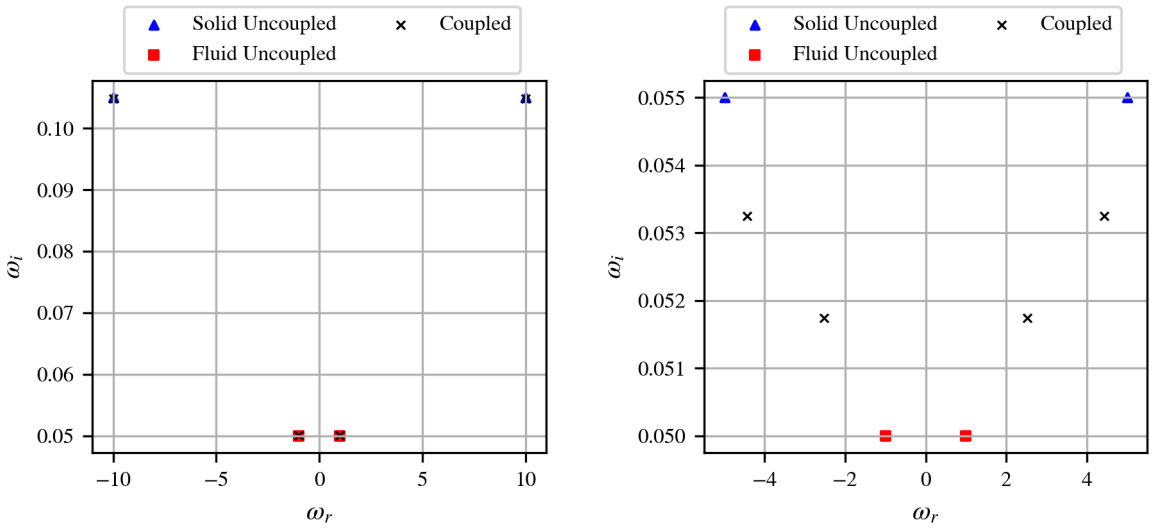
(b) Medium



(c) Strong

Figure 2.4: Eigenvalue comparison between uncoupled and "damping" coupled fluid/structure problems.

In fig. 2.4, the ratio of structural to fluid oscillation frequencies was maintained at 10 with $\delta = 10$ and $\Omega_f = 1$, where no significant impact on the real part of the eigenfrequencies is observed. Referring to fig. 2.2 and eq. (2.9), one can determine that the combined coupling stiffness parameter $(A_0 M)$ needs to be of comparable order to δ^2 to induce changes in the system's frequency. Figure 2.5 depicts the eigenfrequencies obtained with $A_0 M = 100$, where the structural oscillation frequency δ was reduced to 5, and $A_1 M = -0.1$. Here, both the real and imaginary parts of the eigenvalues deviate from those of the uncoupled system. This configuration will here be referred to as a frequency/damping coupling scenario.
Subsequent sections will focus on damping coupling cases, where parameters will be adjusted accordingly.



(a) Weak

(b) Medium

Figure 2.5: Eigenvalues comparison between uncoupled and "frequency/damping" coupled fluid/structure problem.

2.1.3 Outcomes

In this section, the first subsection introduced the linear coupled two-oscillator problem, examining how each parameter influences the system's coupling dynamics. It was observed that the combined coupling stiffness parameter $(A_0 M)$ plays a crucial role in varying the coupled system's frequency, while d_a, d_s, and A_1 affect its damping characteristics. Two main coupling scenarios were defined: a frequency/damping coupling and a straightforward damping coupling case, covering different degrees of coupling from weak to strong. The damping coupling, resembling fluid/structure interactions typical in wind turbine dynamics (as discussed in section 4.4.1), will be the focus in subsequent sections. The next part will address the numerical aspects of the coupled oscillator system, presenting the time discretisation methods for different coupling schemes, followed by a detailed numerical stability and precision analysis.

2.2 Coupling strategies

In computational simulations involving fluid-structure interaction problems, the choice between monolithic and partitioned methods is critical in determining computational efficiency and cost. Monolithic approaches, which solve the entire coupled problem as a unified system, involve inverting large matrix systems at each time step which is computationally intensive and can become prohibitively expensive, especially for large-scale problems. On the other hand, partitioned methods decompose the problem into separate physical domains, allowing each domain to be solved independently. This decomposition can significantly reduce computational costs by avoiding the need for complex matrix inversions and by leveraging specialised solvers for each domain.
This section will explore the advantages and disadvantages of monolithic versus partitioned methods, providing a comparison of their respective strengths and limitations. Following this discussion, an in-depth analytical development of the integration of partitioned coupling schemes into the model problem presented in the previous section will be provided. This analysis will serve as a foundation for the subsequent section, where the numerical properties of these partitioned schemes, specifically their stability and accuracy, will be thoroughly examined. For this purpose, time discretisation will be achieved through linear multistep time integrators [90, 91, 92, 93, 94, 95] which express the most recent step as a function of all previous states through linear relations. The matrix system corresponding to the coupled system will therefore be considered in the following form in the next sections:

$$\begin{bmatrix} \dot{Q} \\ \delta\dot{W} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} Q \\ \delta W \end{bmatrix}, \tag{2.16}$$

where $\delta W = \begin{bmatrix} X_f \\ \dot{X}_f \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix}$ represents the fluid state, $Q = \begin{bmatrix} X_s \\ \dot{X}_s \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$ is the structural state and where A, B, C and D are all (2x2) matrices defined as follows:

$$A = \begin{bmatrix} 0 & 1 \\ -\delta^2 & -d_s \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ M & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 \\ A_0 & A_1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 1 \\ -1 & -d_a \end{bmatrix}.$$

2.2.1 Monolithic vs partitioned coupling

Coupling refers to the mathematical representation of multi-physics problems across both time and space, where different physical domains interact. Various approaches can be employed to address such systems, as described by Felippa and Park in [96]:

- Monolithic Methods: In monolithic methods, the entire problem is treated as a single, unified system. To illustrate the resolution technique, let's discretise the physical eq. (2.16) in time using an implicit Euler scheme. This results in the following reformulation of eq. (2.16):

$$\begin{bmatrix} Q \\ \delta W \end{bmatrix}^{n+1} = \left(\underbrace{\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \Delta t \begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{\mathbf{M}} \right)^{-1} \begin{bmatrix} Q \\ \delta W \end{bmatrix}^{n}. \tag{2.17}$$

Equation (2.17) demonstrates that both the fluid and structural states at the next time step $n+1$ are determined through a single operation, which involves the inversion of matrix $\mathbf{M}$. This process can be computationally intensive, particularly for complex systems with many degrees of freedom. Since all variables of the system are updated simultaneously, this technique ensures a strongly coupled solution, where all interactions are accounted for at each time step. Although monolithic schemes are generally more stable and accurate than partitioned methods, their complexity results in tightly coupled code. This high degree of interdependence makes the code more challenging to maintain and less flexible for integrating modifications or improvements to specific components.

- Partitioned Methods: In contrast, partitioned methods decouple the various physical domains, allowing each to be solved independently. As will be shown in section 2.2.2.2, eq. (2.20) and eq. (2.22) illustrate the independent time discretization of each domain, where coupling terms model the interaction between the physics. This partitioned discretization allows each set of equations to be solved by a respective solver, which exchanges information with neighboring solvers about the state of the interacting physics. This flexibility enables each domain to use its own modeling technique (e.g., finite element for one domain and finite volume for another), discretization (including non-matching meshes), and time integration schemes (such as implicit for one domain and explicit for another). Additionally, this approach facilitates the reuse of existing codes and allows for independent development of the domains, irrespective of the coupling strategy.

  However, partitioned methods are not without drawbacks. As highlighted by Park and Felippa in [96], integrating the physical domains separately introduces a lag, which can lead to potential stability issues. This lag occurs because the first resolved system only receives information about the state of the neighbouring physics from the previous time step, leading to asynchronous integration that does not fully capture the actual action-reaction principle at play. Consequently, the system may not accurately reflect the instantaneous interactions between the physical domains, which can compromise the fidelity of the simulation.

In the next sections, various partitioned coupling schemes will be presented. Details regarding their integration into the previously presented model problem will be given for the coupling techniques that are used in chapter 4 for the resolution of the large wind turbine problem.

2.2.2 Conventional Serial Staggered (CSS) coupling scheme

The conventional serial staggered [97] scheme is one of the most commonly used amongst partitioned methods as it is one of the simplest configurations for sub-domain time integration separation. The different steps of such algorithm are depicted in fig. 2.6, where the evolution of aerodynamic and structural problems, as considered in this work (when DeepLines and CASTOR are coupled), are presented. In order to go from time steps n to $n+1$, four main steps are carried out:

1) structural displacements u^n and velocities $\dot{u}^n$ computed at time t_n are transferred from the structural to the aerodynamic solver.

2) time integration of the fluid problem is conducted and aerodynamic forces F_f^{n+1} induced from structural displacements are obtained at time t_{n+1}

3) aerodynamic forces are transferred to the structural solver F_s^{n+1}

4) time integration of the structural problem is carried out to obtain structural displacements u^{n+1} induced from aerodynamic forces at time t_{n+1}

In steps 2 and 4 of the above described coupling scheme, time integration is achieved through an explicit scheme on the aerodynamic's side and by implicit integration for the structure. This can vary from one problem to another, considering which physics is solved first and whether predictions are made on the first transmitted data's state at time t_{n+1}. In the present work, and more specifically to resemble the coupling between the FVW method code and the FEM solver, a Forward Euler (FE) scheme [98] is coupled to a Newmark method [99] for the fluid and structural problems respectively. Implicit Newmark integration schemes are often used in structural codes, while the explicit Forward Euler scheme is easier to manage with filament emissions as it allows the induced velocity to be computed only once per time step. One can see through fig. 2.6 that a single time step is used both for fluid and structural time integration as well as coupling time step for data transfer. This can become a limitation when two physics have different characteristic times, thus limiting the coupling time step to the smallest one. In such scenario, one of the two physics is "over-resolved", thus increasing computational costs.

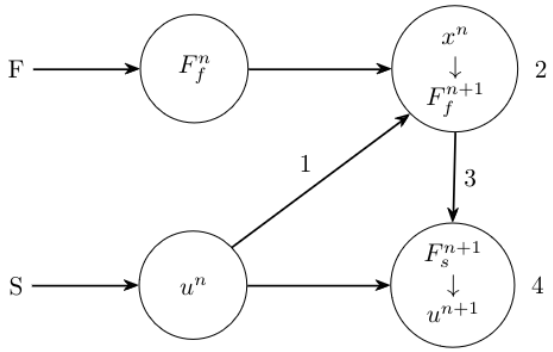


Figure 2.6: Conventional Serial Staggered scheme [2]

2.2.2.1 Interface data exchange or "predictions"

In fig. 2.6, the "lag" described in section 2.2.1 for this partitioned technique is observed to result from the aerodynamic solver computing aerodynamic forces at time step $n+1$ while using structural displacement data from the previous step n:

$$x^{n+1} = u^n.$$

This offset introduces an imbalance between the sub-problems, which can accumulate over the course of the simulation and potentially lead to stability issues. One way to address this is by applying first or second-order displacement extrapolations to enforce continuity of displacements across the physics, as shown below:

$$x^{n+1} = u^n + \Delta t \dot{u}^n, \tag{2.18}$$

$$x^{n+1} = u^n + \frac{3}{2}\Delta t \dot{u}^n - \frac{1}{2}\Delta t \dot{u}_S^{n-1}. \tag{2.19}$$

By using these extrapolation techniques, the boundary displacements are predicted not only based on the displacements computed by the structural solvers but also on the velocities from the previous time steps. For the first-order extrapolation, velocity data from the previous time step is included, while the second-order method also incorporates structural velocities from two earlier time steps.
A stability analysis conducted in [80] demonstrated that these prediction methods can enhance stability and accuracy. These techniques are relatively straightforward to implement in coupling schemes and can be adapted to all the algorithms discussed earlier [87].

2.2.2.2 Time discretisation

The CSS scheme is illustrated in fig. 2.6, where the fluid sub-problem is initially resolved as the primary physics of the two oscillators. At time step $n+1$, its state explicitly depends on the previous states of both the fluid and structural oscillators at time step n. Applying a Forward Euler (FE) time integration scheme to the equation corresponding to the fluid-coupled oscillator in eq. (2.16) yields:

$$\delta \dot{W}^{n+1} = CQ^n + D\delta W^n, \tag{2.20}$$

$$\frac{\delta W^{n+1} - \delta W^n}{\Delta t} = CQ^n + D\delta W^n.$$

Re-arranging the equation to isolate the states at time $n+1$ from the previous states at time step n, the time-discretized fluid sub-problem can be expressed as follows:

$$\begin{bmatrix} X_f^{n+1} \\ \dot{X}_f{}^{n+1} \end{bmatrix} = \Delta t C \begin{bmatrix} X_s^n \\ \dot{X}_s{}^n \end{bmatrix} + (I + \Delta t D) \begin{bmatrix} X_f^n \\ \dot{X}_f{}^n \end{bmatrix}. \tag{2.21}$$

Since the structural sub-problem is solved after the fluid sub-problem, its state at step $n+1$ is implicitly related to both the fluid problem and the structural problem:

$$\dot{Q}^{n+1} = AQ^{n+1} + B\delta W^{n+1}, \tag{2.22}$$

$$\begin{bmatrix} \dot{X}_s{}^{n+1} \\ \ddot{X}_s{}^{n+1} \end{bmatrix} = A \begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \end{bmatrix} + B \begin{bmatrix} X_f^{n+1} \\ \dot{X}_f{}^{n+1} \end{bmatrix}. \tag{2.23}$$

This structural state is resolved using Newmark's scheme which can be re-arranged as follows:

$$\dot{X}_s{}^{n+1} = \dot{X}_s{}^{n} + (1-\gamma)\Delta t \ddot{X}_s{}^{n} + \gamma \Delta t \ddot{X}_s{}^{n+1},$$

$$X_s^{n+1} = X_s^n + \Delta t \dot{X}_s{}^{n} + \Delta t^2 \left[\left(\frac{1}{2} - \beta \right) \ddot{X}_s{}^{n} + \beta \ddot{X}_s{}^{n+1} \right],$$

$$\begin{bmatrix} \dot{X}_s{}^{n+1} \\ \ddot{X}_s{}^{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\gamma}{\beta\Delta t} & 0 \\ \frac{1}{\beta\Delta t^2} & 0 \end{bmatrix}}_{\text{N1}} \left\{ \begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \end{bmatrix} - \begin{bmatrix} X_s^{n} \\ \dot{X}_s{}^{n} \end{bmatrix} \right\} + \underbrace{\begin{bmatrix} (1-\frac{\gamma}{\beta}) & \Delta t(1-\frac{\gamma}{2\beta}) \\ \frac{-1}{\beta\Delta t} & (1-\frac{1}{2\beta}) \end{bmatrix}}_{\text{N2}} \begin{bmatrix} \dot{X}_s{}^{n} \\ \ddot{X}_s{}^{n} \end{bmatrix}. \tag{2.24}$$

Replacing the left hand side of matrix system (4.14) by (2.23), yields the following relation:

$$A \begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \end{bmatrix} + B \begin{bmatrix} X_f^{n+1} \\ \dot{X}_f{}^{n+1} \end{bmatrix} = N1 \left\{ \begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \end{bmatrix} - \begin{bmatrix} X_s^{n} \\ \dot{X}_s{}^{n} \end{bmatrix} \right\} + N2 \left(A \begin{bmatrix} X_s^{n} \\ \dot{X}_s{}^{n} \end{bmatrix} + B \begin{bmatrix} X_f^{n} \\ \dot{X}_f{}^{n} \end{bmatrix} \right). \tag{2.25}$$

By combining system's (2.25) and (2.21), the global coupled problem can be rewritten into a single matrix problem. The spectral properties of transfer matrix Z in (2.26) will help assess the system's numerical stability which will be the focus of section 2.3.1.

$$\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix} \begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \\ X_f^{n+1} \\ \dot{X}_f{}^{n+1} \end{bmatrix} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} \begin{bmatrix} X_s^{n} \\ \dot{X}_s{}^{n} \\ X_f^{n} \\ \dot{X}_f{}^{n} \end{bmatrix},$$

$$\begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \\ X_f^{n+1} \\ \dot{X}_f{}^{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}^{-1} \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix}}_{\text{Z}} \begin{bmatrix} X_s^{n} \\ \dot{X}_s{}^{n} \\ X_f^{n} \\ \dot{X}_f{}^{n} \end{bmatrix}. \tag{2.26}$$

2.2.3 Subcycling coupling scheme

The next algorithm focuses on a partitioned scheme that prevents the "over-resolution" problem mentioned in section 2.2.2. The subcycling scheme's architecture is depicted in fig. 2.7 in which the structural problem represents the physics with lowest characteristic time in comparison to the fluid. Subcycling allows for an extra time step to be added to the coupling scheme: Δt_s being the structure's time step and Δt_f counting as both the fluid integration as well as coupling time steps. The main stages of this algorithm are very similar to that of CSS: extra resolution steps are added to the structure where intermediate stages of structural displacements are computed $u^{n+1,1} \cdots u^{n+1,k}$ (stages 3 through 5) while the aerodynamic data is kept constant.

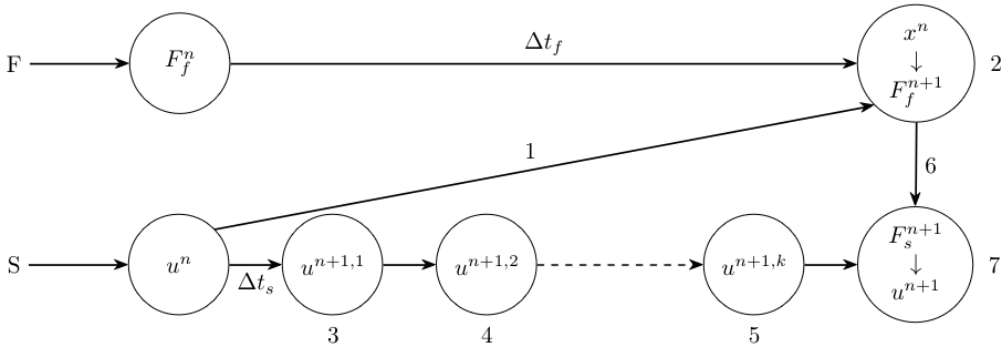


Figure 2.7: Subcycling scheme [2]

2.2.3.1 Time discretisation

In the following parts, the time discretisation of the subcycling algorithm will be discussed. This includes both the "classic" subcycling scheme, as illustrated in fig. 2.7 where the aerodynamic state remains constant during the structural sub-iterations, and a revised version that incorporates predictions for the aerodynamic state, similar to those described in section 2.2.2.1.

Constant regression subcycles

The fluid's problem discretisation is straightforward as it remains the same as in the CSS case given by relation (2.21). However, the structural part of the problem needs to account for the k sub-steps of the subcycles, resulting in the following discretisation:

$$\dot{Q}^{n+1,k+1} = AQ^{n+1,k+1} + B\underbrace{\delta W^{n+1,k+1}}_{\delta W^{n+1}} \quad , \tag{2.27}$$

where $\delta W^{n+1,k+1} = \delta W^{n+1}$ due to constant forecasting.
Newmark's scheme can also be rewritten following the sub-steps:

$$\begin{bmatrix} \dot{X}_s^{\,n+1,k+1} \\ \ddot{X}_s^{\,n+1,k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\gamma}{\beta\Delta t_s} & 0 \\ \frac{1}{\beta\Delta t_s^2} & 0 \end{bmatrix}}_{\text{N1}} \left\{ \begin{bmatrix} X_s^{n+1,k+1} \\ \dot{X}_s^{\,n+1,k+1} \end{bmatrix} - \begin{bmatrix} X_s^{n+1,k} \\ \dot{X}_s^{\,n+1,k} \end{bmatrix} \right\} + \underbrace{\begin{bmatrix} (1-\frac{\gamma}{\beta}) & \Delta t_s(1-\frac{\gamma}{2\beta}) \\ \frac{-1}{\beta\Delta t_s} & (1-\frac{1}{2\beta}) \end{bmatrix}}_{\text{N2}} \begin{bmatrix} \dot{X}_s^{\,n+1,k} \\ \ddot{X}_s^{\,n+1,k} \end{bmatrix},$$

where $N1$ and $N2$ are different to the previous matrices obtained for CSS in system (4.14), as structural time steps are now accounted for.
Combining equation (2.27) with the above Newmark discretisation, the structural subcycled discretisation yields:

$$\underbrace{(A-N1)}_{A_1} Q^{n+1,k+1} + \underbrace{(I-N2)B}_{B_1} \delta W^{n+1,k+1} = \underbrace{(N2A-N1)}_{A_0} Q^{n+1,k}. \tag{2.28}$$

A concrete example of the time discretization of a subcycled coupling using constant regression is provided below, with 3 subcycles applied to the structure:

1. Fluid time integration: $\delta W^{n+1} = C_0 Q^n + D_0 \delta W^n$
2. Structural time integration
   (a) sub-it 1 : $A_1 Q^{n+1,1} + B_1 \delta W^{n+1} = A_0 Q^n$
   (b) sub-it 2 : $A_1 Q^{n+1,2} + B_1 \delta W^{n+1} = A_0 Q^{n+1,1}$
   (c) sub-it 3 : $A_1 Q^{n+1,3} + B_1 \delta W^{n+1} = A_0 Q^{n+1,2}$
   (d) Coupling time step : $A_1 Q^{n+1} + B_1 \delta W^{n+1} = A_0 Q^{n+1,3}$

The above steps can be represented by an equivalent matrix system as shown in eq. (2.29). Here, the matrix product of $Z1$'s inverse and $Z0$ produces the equivalent of the Z matrix from eq. (2.26), which enables the stability assessment of the coupled numerical scheme.

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & A_1 & B_1 \\ 0 & 0 & A_1 & -A_0 & B_1 \\ 0 & A_1 & -A_0 & 0 & B_1 \\ A_1 & -A_0 & 0 & 0 & B_1 \end{bmatrix}}_{\text{Z1}} \begin{bmatrix} Q^{n+1} \\ Q^{n+1,3} \\ Q^{n+1,2} \\ Q^{n+1,1} \\ \delta W^{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} C_0 & 0 & 0 & 0 & D_0 \\ A_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\text{Z0}} \begin{bmatrix} Q^n \\ Q^{n+1,3} \\ Q^{n+1,2} \\ Q^{n+1,1} \\ \delta W^n \end{bmatrix}. \tag{2.29}$$

Linear regression subcycles

Considering subcycling steps where a linear extrapolation of aerodynamic forces is used, the k sub-steps in the structural subcycles can be expressed as follows:

$$\dot{Q}^{n+1,k+1} = AQ^{n+1,k+1} + B \underbrace{\delta W^{n+1,k+1}}_{X_f^{n+1}+\Delta t_s(k+1)\dot{X}_f^{n+1}} . \tag{2.30}$$

Taking Newmark's scheme obtained in the previous section and combining it with the above equation, one gets:

$$AQ^{n+1,k+1} + B^{**}\delta W^{n+1} = N1\left\{Q^{n+1,k+1} - Q^{n+1,k}\right\} + N2\left[AQ^{n+1,k} + B^{*}\delta W^{n+1}\right],$$

$$\underbrace{(A - N_1)}_{A_1} Q^{n+1,k+1} + \underbrace{(B^{**,k+1} - N_2 B^{*,k})}_{B_1^*} \delta W^{n+1} = \underbrace{(N_2 A - N_1)}_{A_0} Q^{n+1,k} \quad ,$$

where $B^{**,k+1}$ and $B^{*,k}$ are expressed as follows:

$$B^{*,k} = \begin{bmatrix} 0 & 0 \\ M & kM\Delta t_s \end{bmatrix}, \quad B^{**,k+1} = \begin{bmatrix} 0 & 0 \\ M & (k+1)M\Delta t_s \end{bmatrix}.$$

Matrix system (2.29) would maintain the same structure when linear regression subcycles are used. However, in this case, the matrices B_1 need to be replaced with the newly defined B_1^*.

2.2.4 Conventional parallel staggered coupling scheme

The following diagram represents the "Conventional Parallel Staggered" algorithm. This method can be implemented using both conventional serial staggered and subcycling algorithms. As shown, at each coupling time step, information is exchanged between the fields, requiring two predictions: one for the structural displacements to be transmitted to the fluid and another for the fluid forces acting on the structure. This algorithm offers faster computation times, as it is optimized for parallel computing architectures. However, despite the speed advantage, it is important to note that this method is generally less stable and less precise compared to other approaches, due to the necessity of making two predictions [87]. The reduction in stability has been demonstrated in a 1D case study, as discussed in [80].

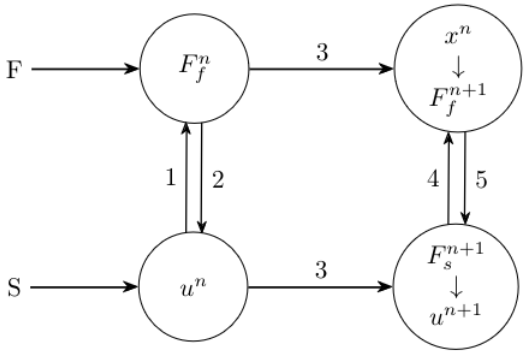


Figure 2.8: Parallel scheme

2.2.5 Asynchronous coupling

In order to counter the lack of precision when predicting the solution of structural displacements or fluid forces in either one of the above algorithms, Piperno *et al.* [80] suggested to implement a new type of architecture which calls for desynchronizing data transmission between codes. The idea behind this method is to add half a time step for which in the first half, both codes solve their respective problems in a parallel manner, before exchanging data, allowing the algorithm to get a better solution at each full iteration. Farhat *et al.* proved in [79] that desynchronizing the coupling time steps between two codes would allow for a better precision in time coupling schemes of up to second order.

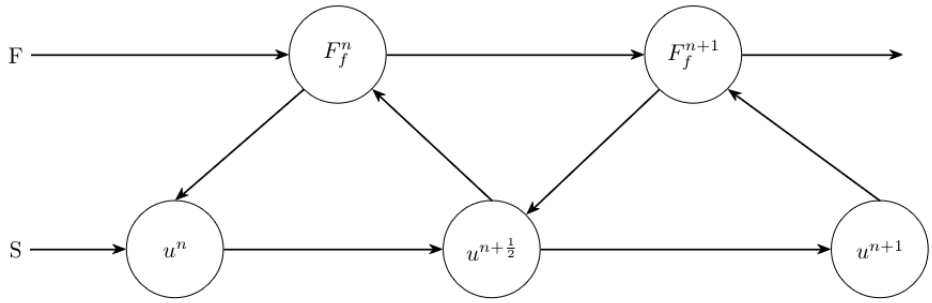


Figure 2.9: Asynchronous scheme

2.2.6 Spatial multi-domain coupling strategies

The first part of this chapter focused on various coupling schemes, with a particular emphasis on time discretisation strategies. As outlined in the previous chapter, wind turbines are complex multi-body systems involving multiple interacting physical domains. Additionally, individual components can be subdivided into smaller sub-parts to account for specific spatial discretisations. In the context of fluid-structure interaction (FSI) and the broader modelling of the entire wind turbine system, these sub-parts must be coupled effectively to ensure accurate simulation. Therefore, spatial coupling techniques are essential to integrate the different physical domains and subcomponents. This section will provide a general overview of spatial coupling techniques, though it is not the primary focus of this work.

Just as in time integration coupling schemes, where different physical domains may have their own time discretisations, each subdomain can be modelled using different meshes, each having their own spatial discretisations. For instance, a fine mesh may be required in certain subdomains to ensure sufficient accuracy, while other regions can use a coarser mesh to reduce computational costs. Maintaining proper connection conditions at the boundaries between these subdomains is crucial to preserving the overall stability and accuracy of the system.

In cases where non-matching grids arise, various methods can be employed to ensure accurate field transfer between subdomains. These methods range from simpler direct approaches, such as interpolation techniques, to more indirect methods involving Lagrange multipliers [100], with the choice depending on the complexity and specific requirements of the problem.

One of the most basic approaches is the "nearest neighbour interpolation" [101], where each point in one domain (e.g., the fluid) is assigned the value (such as displacement or velocity) of the closest point in the neighbouring domain (e.g., the structure). While this method is one of the easiest to implement, it is often unsuitable for high-accuracy applications in complex cases, as satisfactory results are typically only achieved when the meshes are almost matching [100].

Data transfer between non-matching meshes can also be achieved through projection methods, where a point or element from one mesh is orthogonally projected onto the adjacent mesh, and the value at the projection point or the area of intersection is used to assign the value at the original grid point [100, 77, 102].

Spline methods, including radial basis functions (RBFs) [103], provide a more accurate and versatile approach for transferring data between non-matching meshes.

In contrast to these direct approaches, Lagrange multiplier methods [104] use a more indirect way to handle the coupling between non-matching or non-conforming grids. Lagrange multipliers are often used to enforce displacement continuity across structural subdomains, particularly in scenarios involving non-conforming meshes.

2.2.7 Outcomes

The second part of this chapter focused on presenting various coupling strategies for fluid-structure interaction simulations, with particular emphasis on time coupling techniques. Different partitioned methods were explored, with time discretisation achieved using linear multistep methods for both Conventional Serial Staggered

(CSS) and subcycling coupling techniques. These methods were applied to the linear model problem introduced in the first section. The upcoming section will delve into the application of these discretisation techniques to the model problem, with the aim of evaluating their numerical properties, specifically in terms of stability and accuracy. This analysis will provide valuable insights into the potential behaviour of these methods when applied to larger-scale problems, offering a simplified perspective.

2.3 Properties of coupling schemes

2.3.1 Stability CSS vs Subcycling

In section 2.2.1, a concern raised regarding partitioned coupling is the issue of numerical stability. This instability can arise from the data exchange process, which can introduce a lag between the states of the two physical systems in certain schemes. Over time, the error caused by this lag can accumulate, potentially leading to stability problems in the simulation. The following sections will provide a comparative stability analysis of the CSS and subcycling schemes applied to the linear two-oscillator model problem.

2.3.1.1 Approach for numerical stability analysis

In previous sections where the physical problem was described, when stability was discussed, the focus was on physical stability. In this context, stability refers to the behaviour of a system over time, particularly whether small perturbations decay or grow as time progresses. When considering a general system described by the equation $\dot{x} = f(x)$, where x represents the state of the system, analysing physical stability is achieved by perturbing the system around an equilibrium point x_0, so that $x = x_0 + \epsilon x'$, where ϵ is a small perturbation and x' represents the deviation from equilibrium. Physical stability examines whether the norm $||x(t)||$ of the system's state remains bounded or decays to zero as time t tends toward infinity. If small perturbations decay over time, the system is here said to be stable in the physical sense.
In the next parts, the focus is on numerical stability, which concerns the behaviour of discrete approximations of the system. When now discretising the system and representing the evolution of its state at time step k as $\bar{x}_{k+1} = A\bar{x}_k$, where A is the matrix governing the system's evolution in discrete time, numerical stability requires that any small errors introduced at each time step remain bounded over time. Stability can be evaluated by analysing the spectral properties of the matrix A, specifically by ensuring that the magnitudes of its eigenvalues are less than or equal to 1.
The approach used to analyse the numerical stability of different schemes is as follows. First, the stability of each individual uncoupled problem with its associated time integrator is assessed. This can be done by omitting the coupling terms, thus removing the influence of the other oscillator in the time discretisation described in section 2.2.2.2. For the fluid resolution, the fluid discretisation from eq. (2.21),

without the coupled structural oscillator, would then be written only in terms of the fluid states as:

$$\begin{bmatrix} X_f^{n+1} \\ \dot{X}_f{}^{n+1} \end{bmatrix} = \underbrace{I + \Delta t D}_{S} \begin{bmatrix} X_f^n \\ \dot{X}_f{}^n \end{bmatrix}. \tag{2.31}$$

Similarly, the structural oscillator's time discretisation, without the coupling to the fluid oscillator as in eq. (2.25), would give:

$$\begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \end{bmatrix} = \underbrace{(A - N_1)^{-1}(N_2 A - N_1)}_{S} \begin{bmatrix} X_s^n \\ \dot{X}_s{}^n \end{bmatrix}. \tag{2.32}$$

To analyse the numerical stability of each uncoupled system, the spectral properties of the matrix S are examined, providing a condition on the time step Δt for which the system remains stable.
In the coupled cases, the same analysis is applied both for the CSS scheme and the subcycling scheme. For the CSS scheme, the full matrix system including the coupling terms from eq. (2.26) is analysed, while for the subcycling scheme, the matrix system in eq. (2.29) is analysed for each given number of sub-iterations.
The full matrix system for the CSS scheme is provided again below:

$$\begin{bmatrix} X_s^{n+1} \\ \dot{X}_s{}^{n+1} \\ X_f^{n+1} \\ \dot{X}_f{}^{n+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_1 & B_1 \\ C_1 & D_1 \end{bmatrix}^{-1} \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix}}_{\mathrm{Z}} \begin{bmatrix} X_s^n \\ \dot{X}_s{}^n \\ X_f^n \\ \dot{X}_f{}^n \end{bmatrix}.$$

By analysing the spectral properties of matrix Z, the numerical stability of the coupled system can be assessed. This analysis yields a critical coupling time step Δt_c, below which the system is numerically stable and above which it becomes unstable.
A concrete example is shown in fig. 2.10, fig. 2.11, fig. 2.12, and fig. 2.13, where the time evolution of the structural and fluid problems is presented and compared to a monolithic solution. The critical time step for the given problem using a CSS scheme is found to be at $\Delta t_c = 0.03\,\mathrm{s}$. As seen in fig. 2.10, at this time step resolution, both the structural and fluid problems exhibit stable behaviour and converge over time. On the other hand, when using a slightly larger time step of $\Delta t = 0.04\,\mathrm{s}$, fig. 2.11 shows that both systems become unstable, with a faster divergence rate in the structure compared to the fluid.
At this same time step, using a subcycling scheme with a single sub-iteration step renders the system numerically stable, unlike the CSS scheme. The subcycling scheme has its own critical time step, here found to be $\Delta t_c = 0.07\,\mathrm{s}$, beyond which the system becomes unstable, as shown in fig. 2.13.

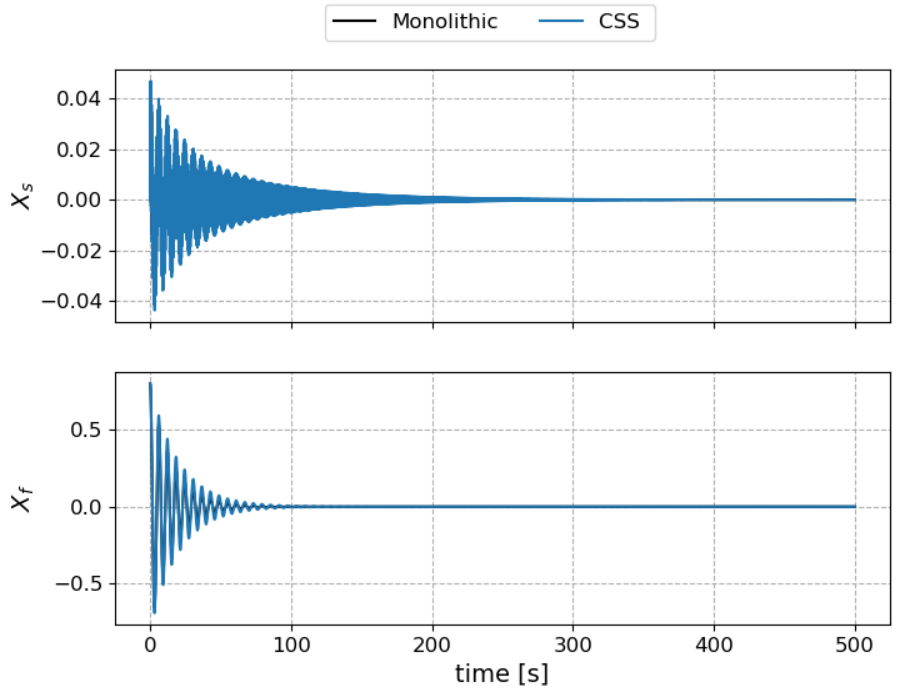


Figure 2.10: Stable case of coupled problem using CSS coupling scheme at critical time step $\Delta t_c = 0.03\,\mathrm{s}$

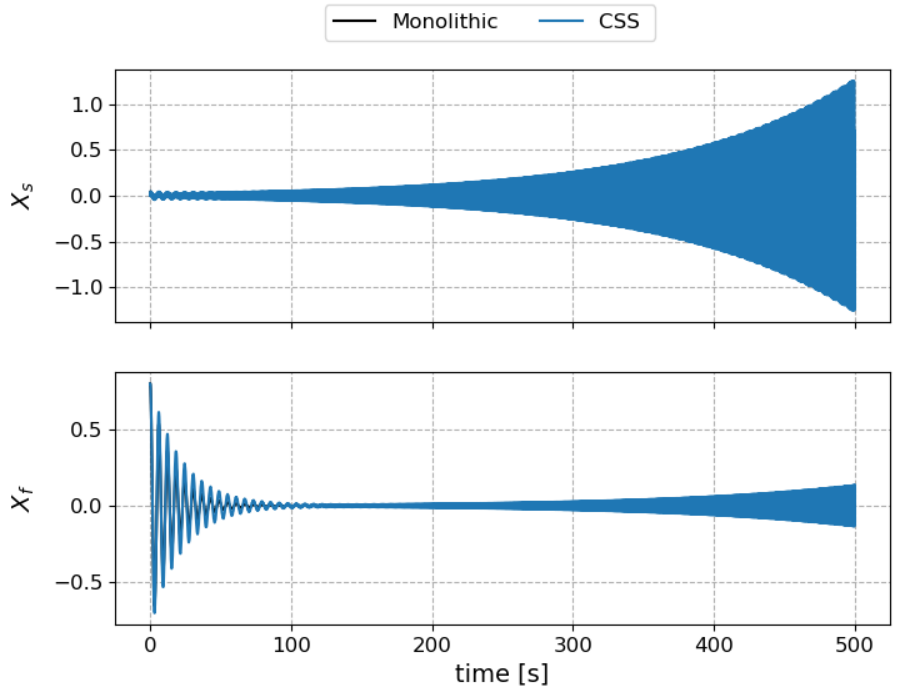


Figure 2.11: Unstable case of coupled problem using CSS coupling scheme above critical time step $\Delta t = 0.04\,\mathrm{s}$

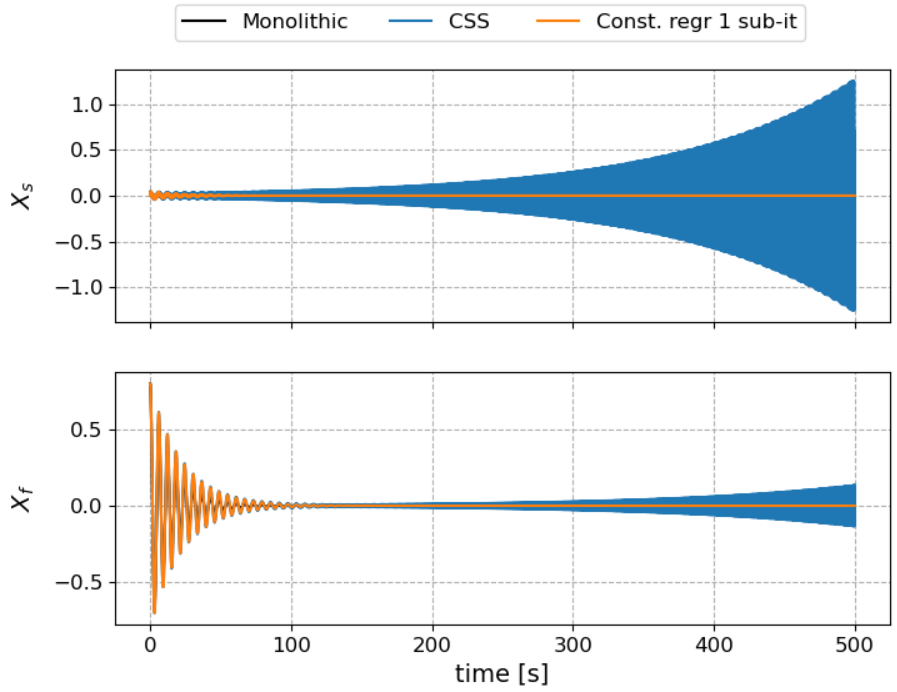


Figure 2.12: Unstable case of coupled problem using CSS coupling scheme and stable case using subcycling above critical time step, at $\Delta t = 0.04\,\mathrm{s}$

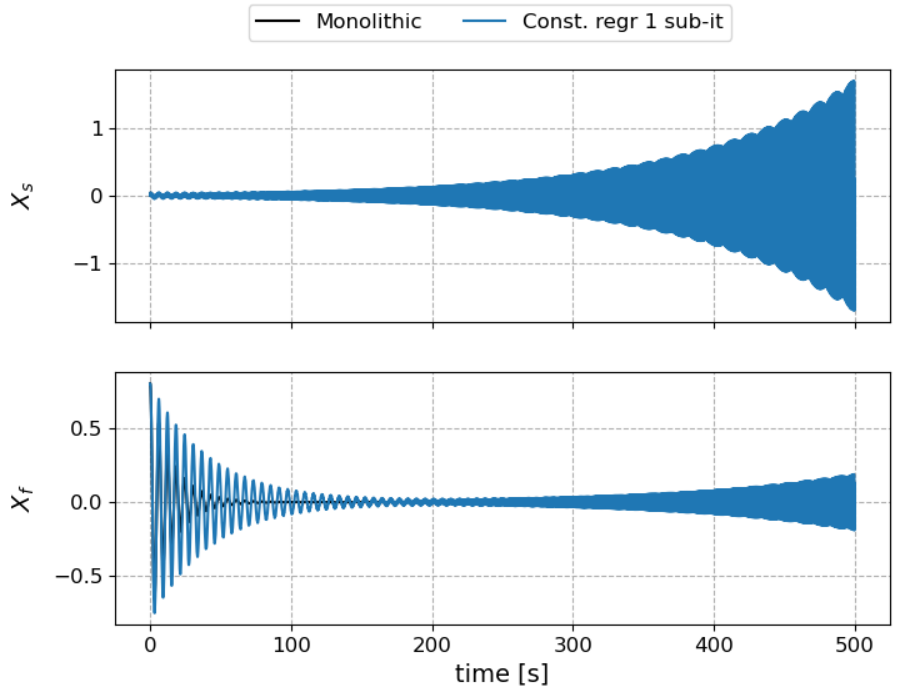


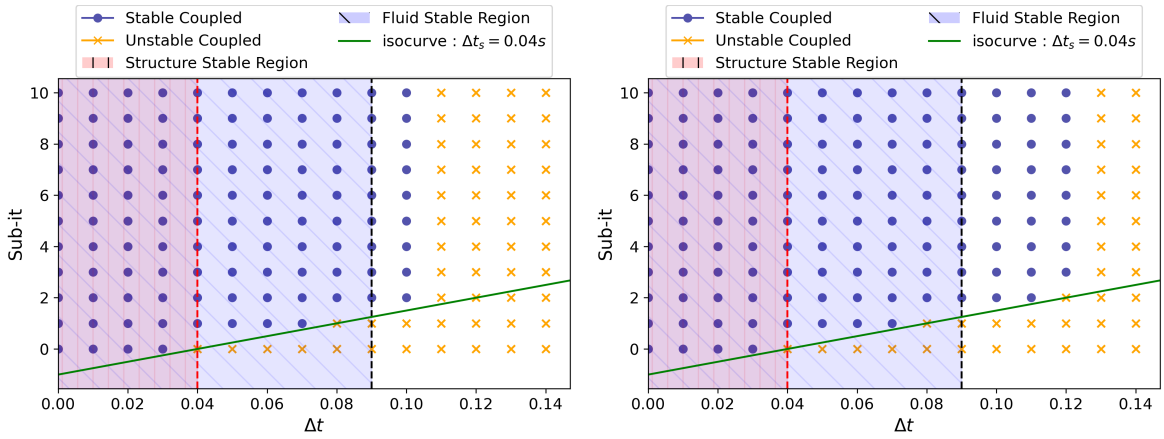
Figure 2.13: Unstable case of coupled problem using subcycling above critical time step, at $\Delta t = 0.08\,\mathrm{s}$

2.3.1.2 Damping coupling effects

The first stability analysis focuses on the model problem where damping coupling effects are present, using parameters from the initial case presented in section 2.1.2.3. This analysis examines the numerical stability of the fluid and structural oscillators both in a decoupled state and when coupled through a CSS or subcycling scheme. The study is conducted for varying degrees of coupling between the oscillators and includes subcycling cases that employ both constant and linear forecasting techniques for predicting aerodynamic forces to be transmitted to the structural solver.

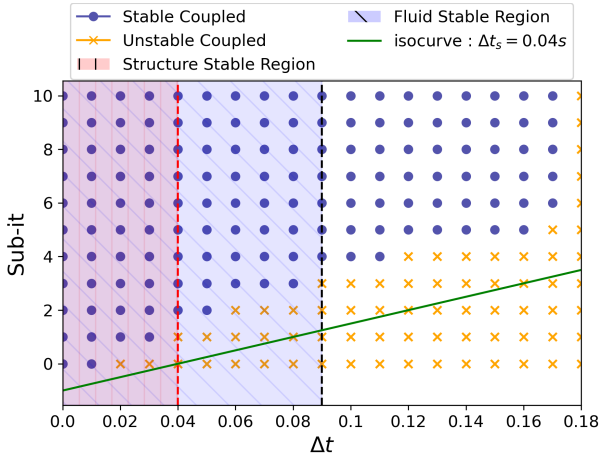
Constant regression sub-steps

In this first case, a stability analysis is presented in fig. 2.14, where constant regression is used for the subcycling steps. Figure 3.17a through fig. 3.17c represent different degrees of coupling between the oscillators. In each graph, the red and blue zones indicate the stability regions of the uncoupled structural and fluid oscillators, respectively. Crosses denote coupled unstable regions, while dots denote stable regions for a given number of subcycles. A coupling with zero subcycles corresponds to a CSS coupling. The dotted green line represents the isocurve defining the structural oscillator's stability zone: for instance, with a coupling time step of $\Delta t = 0.09s$ and two subcycling steps, the structural time step is reduced to $\Delta t_s = 0.03s$, thus re-entering the structural oscillator's stability zone (beneath the isocurve).
It can be observed that when using a CSS scheme in weakly or moderately coupled cases, the stability limit of the coupled oscillators is constrained by the critical time step of the most restrictive uncoupled oscillator, which is the structure in this scenario. In strongly coupled cases, the CSS scheme's critical time step is even lower than that of the structural oscillator. By introducing sub-steps, the stability region of the coupling is expanded in all cases, with critical coupling time steps increasing as the degree of coupling increases. In weak and moderately coupled cases, the slope of the subcycling stability limit follows that of the isocurve. This indicates that in regions above the fluid's critical time step, the stability of the structure stabilizes the coupling of both oscillators. Conversely, in fig. 3.17c, the slope of the subcycling stability limit deviates from the isocurve, indicating that in regions above $\Delta t = 0.09s$, the fluid's instability destabilizes the coupling between the two oscillators.



(a) Weakly coupled

(b) Medium coupled

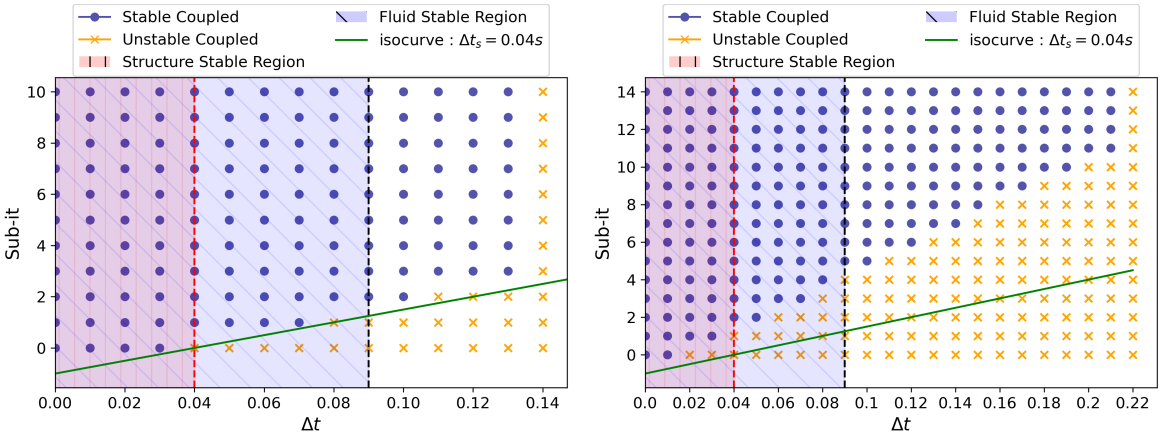


(c) Strongly coupled

Figure 2.14: Stability analysis of different degrees of coupling using the constant regression subcycling scheme.

Linear regression sub-steps

In this section, stability graphs for a linear regression subcycling scheme are presented. Results for the weakly coupled case are omitted as they mirror those in fig. 3.17a using constant regression. When comparing fig. 2.15a with fig. 3.17b, it can be seen that linear regression subcycles extend the stability region further compared to constant regression, allowing for a slightly larger critical coupling time step of $\Delta t = 0.13s$ versus $\Delta t = 0.12s$ previously observed. This effect is more pronounced in the strongly coupled cases: fig. 3.17c indicates a critical coupling time step of $\Delta t = 0.17s$, whereas the linear regression technique achieves $\Delta t = 0.22s$ (shown in fig. 2.15b). Here, the slope of the subcycling stability limit diverges more noticeably from the isocurve compared to the constant regression case, indicating that the fluid oscillator's instability and its approximation using linear forecasting destabilize the coupling to a greater extent.



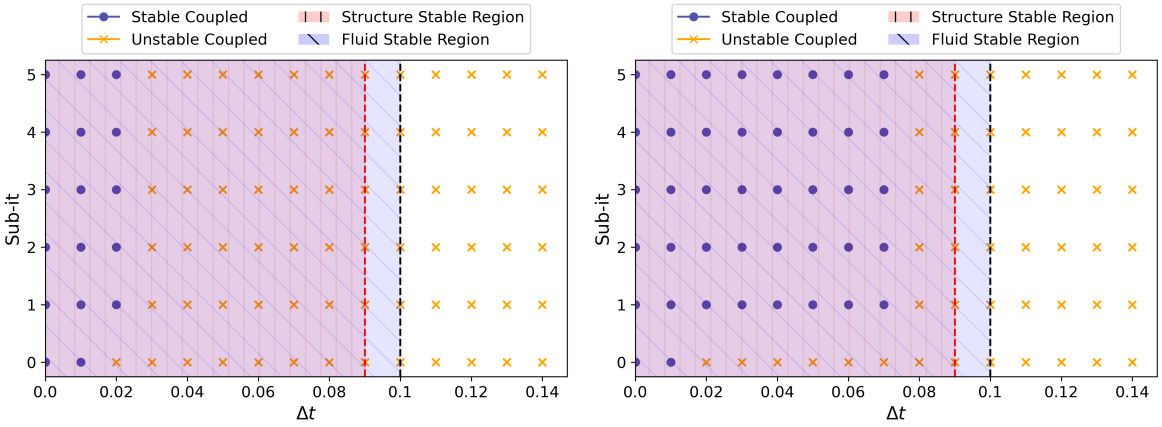
(a) Medium coupled

(b) Strongly coupled

Figure 2.15: Stability analysis of different degrees of coupling using the linear regression subcycling scheme.

2.3.1.3 Frequency and damping coupling effects

The second case focuses on a stability analysis of the model problem, where both frequency and damping coupling effects are present, as introduced in section 2.1.2.3. In figure 2.16, fig. 2.16a and fig. 2.16b display the results for subcycling schemes using constant and linear regressions, respectively. Notably, in this case, the stability limits for both uncoupled oscillators are much closer than in the previous scenario. When applying the CSS scheme (subcycling with zero sub-iterations) under conditions with added frequency coupling effects, the critical time step for the coupling is significantly smaller than the critical time steps of either of the two uncoupled oscillators, with a critical coupling time step of $\Delta t_c = 0.01s$ compared to $\Delta t_s = 0.09s$ and $\Delta t_f = 0.1s$ for the structural and fluid critical time steps, respectively. Additionally the introduction of frequency coupling effects influences the stability when using a subcycling scheme. In the previous case, the subcycling scheme extended the stability limit of the coupling beyond that of the less restrictive oscillator. However, in this scenario, using subcycling with constant regression only slightly extends the stability region of the coupling and does not exceed the critical time steps of either uncoupled oscillator. The same applies to the case with subcycling using linear regression, where a critical coupling time step of $\Delta t_c = 0.07s$ is achieved, compared to $\Delta t_c = 0.02s$ with constant regression. Nevertheless, the stability region of the coupling remains reduced compared to the stability regions of the uncoupled systems.



(a) Constant regression

(b) Linear regression

Figure 2.16: Stability analysis of different degrees of coupling using the constant and linear regression subcycling schemes.

2.3.2 Accuracy

The order comparison between a coupled system using a CSS and various subcycling schemes is illustrated in fig. 2.17. This analysis was conducted on the coupling case from section 2.3.1.2 where damping coupling effects are prominent. Each of the plots below shows the $L2$ norm of the error relative to a reference solution computed by monolithic solving of eq. (2.16) plotted against the coupling time steps. The CSS scheme exhibits first-order behaviour, attributable to the Forward Euler scheme used for the fluid's time integration. For each subcycling scheme, using different types of regressions and varying numbers of sub-steps results in first-order coupling. Both constant and linear regression subcycling schemes show a slightly smaller relative error compared to the CSS case, a difference that becomes more pronounced with additional sub-steps and consequent reduction in the structural integration time step.

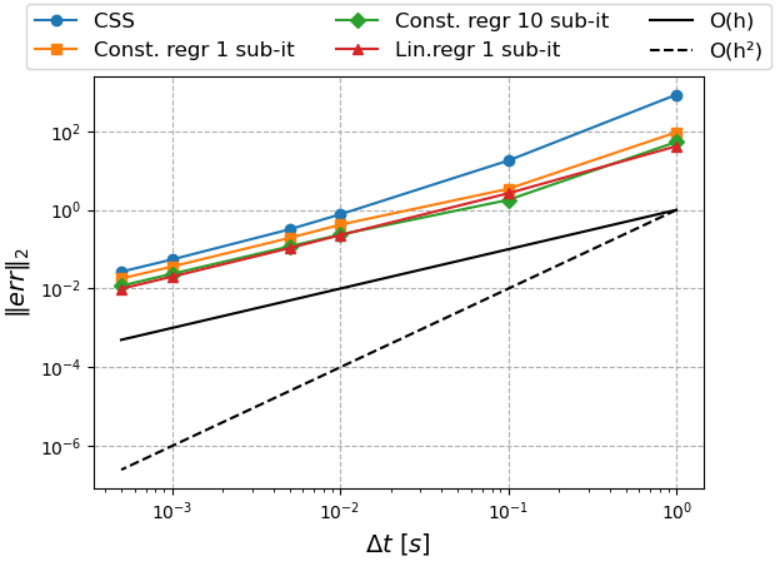


Figure 2.17: Order comparison of CSS and subcycling schemes.

2.3.3 Outcomes

In the previous section, the time discretisation technique using a linear multistep method was introduced for both CSS and subcycling schemes, applied to the coupled oscillator problem. This provided a foundation for the stability and precision analysis presented in this section. In cases where damping coupling effects are prominent, it was revealed that the stability region is extended when a subcycling scheme is used, compared to CSS coupling, while the same order of accuracy is maintained. However, a reduction in the stability region is observed when frequency coupling effects are also present. The expanded stability zone provided by the subcycling scheme allows for a larger coupling time step between the structural and aerodynamic solvers. This aspect is particularly relevant in chapter 4, where subcycling is applied between a structural and aerodynamic solver in a large-scale wind turbine problem, in which the characteristic timescales of each sub-problem differ significantly. In the upcoming section, the implementation details of the presented schemes within the DeepLines Wind^{TM} framework, introduced in chapter 1, will be discussed.

2.4 Subcycling architecture for DeepLines Wind

Having presented the numerical properties of various partitioned schemes, the focus now shifts to demonstrating how certain coupling schemes can be integrated into the aeroelastic tool, DeepLines Wind™. Additionally, this section will outline the adaptations required for implementing a subcycling algorithm within this framework.
The first section provides an overview of the implementation architecture necessary to enable subcycling in this context. Next, a revisited and adapted subcycling scheme tailored to fit within this specific framework, is presented. Finally, the third section explores aerodynamic force forecasting methods, including polynomial regression and ARIMA models, expanding on the constant and linear forecast methods discussed in the previous model problem.

2.4.1 Implementation details

To enable subcycled coupling between DeepLines and any aerodynamic solver, it was necessary to revisit the original coupling implementation used in CSS aeroelastic simulations. The original approach involved connecting DeepLines to the appropriate APIs (Application Programming Interface) to call the required dynamic link library (DLL) associated with a specific aerodynamic solver. This setup allowed for the transfer of information between the structural code and the aerodynamic solver at each time step using dedicated API functions.
Figure 2.18 illustrates the updated coupling architecture, which introduces a new component, the coupler, and an additional aerodynamic library, the meta model. The coupler module enables the use of two aerodynamic libraries:

- "High-fidelity" resolution: This library corresponds to an actual fluid solver, such as CASTOR that uses a free vortex wake method or the BEM solver AeroDeep, to compute precise aerodynamic forces.
- "Low-fidelity" resolution: The meta model provides a simplified, less computationally intensive approach. Instead of solving the full aerodynamic problem at each time step, the meta model generates approximate aerodynamic force data.

This new setup defines two distinct coupling time steps: a structural solver time step Δt_s, which defines the times at which DeepLines receives aerodynamic force data via the meta model, providing a continuous update without requiring the full aerodynamic problem to be resolved. A fluid solver resolution time step Δt_f, which represents the lower frequency at which the more computationally intensive aerodynamic solver runs, thus reducing the need for costly fluid force calculations at each time step.

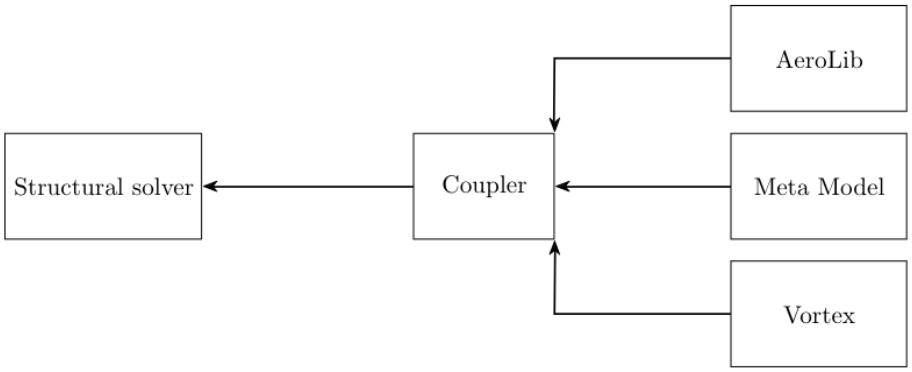


Figure 2.18: Meta model implementation architecture

2.4.2 Subcycling scheme with meta model

Figure 2.19 highlights the use of the integrated meta model within the revised subcycling algorithm. At each coupling time step, corresponding to the aerodynamic solver's resolution rate Δt_f, the fluid solver communicates aerodynamic forces directly to the structural code. During the intermediate stages, which align with the structural solver's characteristic time step Δt_s, the meta model interfaces with DeepLines. It estimates the fluid state during these intermediate steps by extrapolating aerodynamic forces based on previous computations. Various regression techniques were considered and implemented in this process, including constant, linear, and polynomial extrapolations. More advanced methods, such as ARIMA models [105, 106], can also be integrated to handle more complex aerodynamic force scenarios. The use of either of the forecasting methods will be the focus of the next section.

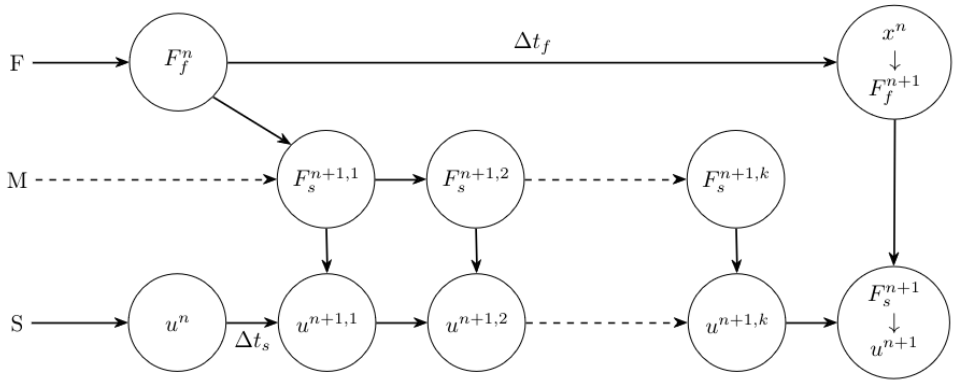


Figure 2.19: Subcycling scheme [2]

2.4.3 Forecasting techniques

The meta-model library provides access to the aerodynamic solver's computed forces, allowing them to be processed as a time series for forecasting future values to feed back into the structural solver. Several forecasting techniques can be considered for this purpose, which will be outlined in the following sections. First, simpler methods such as constant, linear, and polynomial regression will be presented for their application as meta-models. Following this, a more advanced technique, the ARIMA (AutoRegressive Integrated Moving Average) model, will be discussed in its role as a meta-model. In the final section, the advantages and disadvantages of each method will be examined, along with an analysis of the types of simulations where each approach is most suitable.

2.4.3.1 Constant, linear and polynomial regressions

This section is dedicated to presenting how aerodynamic forces are forecasted by the meta model from fig. 2.18 through the use of methods that were presented in section 2.2.3.1 as well as polynomial and ARIMA methods in the DeepLines Wind^{TM} framework.

Constant regression meta model

In the constant regression approach, the aerodynamic forces are assumed to remain constant, using only the aerodynamic force at the last computed time step t^n. This simple method assumes no change in forces between the last computed time step and any future intermediate time steps. Thus, the forecasted aerodynamic force for any intermediate time $t^{n,k}$ is given directly by the last known value of the force:

$$F_s^{n,k} = F_f^n.$$

Linear regression meta model

For linear extrapolation, the aerodynamic forces are estimated by assuming a linear trend between the forces at the last two computed time steps: F_f^n at t^n and F_f^{n-1} at t^{n-1}. The prediction of the forces at the intermediate time steps is achieved by fitting the two last computed aerodynamic forces to the following equation:

$$F(t) = at + b,$$

where the slope a and intercept point b can be obtained through:

$$a = \frac{F_f^n - F_f^{n-1}}{t^n - t^{n-1}},$$

$$b = F_f^n - at^n.$$

Any intermediate stage of forecasted aerodynamic forces can thus be obtained through:

$$F_s^{n,k} = at^{n,k} + b.$$

Polynomial regression meta model

To forecast the aerodynamic forces required by the structural solver during intermediate stages through polynomial regression, a quadratic polynomial is constructed to approximate the forces based on the last three computed time steps. The quadratic polynomial is defined as:

$$F(t) = at^2 + bt + c,$$

where a, b and c are coefficients determined by fitting the polynomial through three known points: F_f^{n-2}, F_f^{n-1} and F_f^n. These points represent the aerodynamic forces computed by the aerodynamic solver at the corresponding time steps t^{n-2}, t^{n-1} and t^n.
By solving the system of equations generated for each of these points, coefficients a, b and c can be derivede through:

$$a = \frac{t^{n-2}(F_f^n - F_f^{n-1}) + t^{n-1}(F_f^{n-2} - F_f^n) + t^n(F_f^{n-1} - F_f^{n-2})}{(t^{n-2} - t^{n-1})(t^{n-2} - t^n)(t^{n-1} - t^n)},$$

$$b = \frac{(F_f^{n-1} - F_f^{n-2})}{(t^{n-1} - t^{n-2})} - a(t^{n-2} + t^n),$$

$$c = F_f^{n-2} - a(t^{n-2})^2 - bt^{n-2}.$$

With these coefficients, the aerodynamic forces at any intermediate time $t^{n,k}$, denoted by $F_s^{n+1,k}$, can be forecasted for transfer to the structural solver. The expression for the forecasted forces is:

$$F_s^{n,k} = a(t^{n,k})^2 + bt^{n,k} + c.$$

2.4.3.2 ARIMA model

One of the forecasting techniques considered in this work is the ARIMA model (Autoregressive Integrated Moving Average), which is widely used for time series forecasting based on historical data. The general steps behind this method will be presented, but for more detail the reader is referred to ([107], [106], [108], [105]). The ARIMA model combines three key components:

- AR (AutoRegressive): This component establishes a relationship between the current value of the time series and its past values. Specifically, it assumes that the current value can be modelled as a linear combination of previous values in the series, allowing for forecasting based on past observations.

- I or Integrated component: This handles any non-stationarity in the time series by differencing the data to remove trends or seasonal patterns. This step ensures that the model focuses on the underlying stochastic process rather than on trends or cyclic behaviours.
- MA or Moving Average: This complements the autoregressive component by not only considering past observations but also accounting for the errors from previous forecasts. This allows the model to correct itself by incorporating deviations between predicted and actual values, refining future predictions.

An ARIMA model is typically defined as an ARIMA(p,d,q) method where p is the order of the autoregressive part, meaning the number of past observations used to predict the next value, d is the degree of differencing required to make the time series stationary and q is the order of the moving average part, representing the number of past forecast errors incorporated into the model to improve prediction accuracy. This can be translated into the following expression:

$$\begin{aligned}\Delta^d F^n = \phi_1 F^{n-1} + \phi_2 F^{n-2} + \cdots + \phi_p F^{n-p} \\ + \theta_1 \epsilon^{n-1} + \theta_2 \epsilon^{n-2} + \cdots + \theta_q \epsilon^{n-q} + \epsilon^n,\end{aligned} \tag{2.33}$$

where $\Delta^d F^n$ is the number of times (d) that the time series needs differencing to become stationnary (with $\Delta F^n = F^n - F^{n-1}$), $\phi_1 \cdots \phi_p$ are the autoregressive coefficients, $\theta_1 \cdots \theta_q$ the moving average coefficients and ϵ_n is an error term at time n.
In order to fit the ARIMA model to the aerodynamic forces time series, the parameters of the method need to be determined. First, the number of autoregressive, differencing and moving average terms p, d and q need to be chosen. For that matter, autocorrelation functions (ACF), partial autocorrelation functions (PACF) and Augmented Dickey-Fuller (ADF) tests help define the parameters.
The ADF test checks whether a time series is stationnary and is going to define the number of differencing steps needed to make the time series stationnary. The ACF shows how correlated the current value of the time series is with the previous values and helps identify the $MA(q)$ terms, while the PACF shows the correlation between the time series and the past values without taking into account the influence of the intermediate values which helps identify the $AR(q)$ terms.
Once, the p, d and q terms are determined, eq. (2.33) is fit to the time series data using statistical techniques such as the Maximum Likelihood Estimation (MLE) or a least square method. This helps determine the values of the autoregressive and moving average coefficients. The latter are then used in the ARIMA model to predict the next full time step in the time series F^{n+1}, from which sub-steps $F^{n,k}$ are then determined through linear or polynomial interpolation techniques between the predicted.

2.4.3.3 Discussion on forecasting methods

The aerodynamic forces acting on wind turbine blades are dynamic and highly dependent on inflow conditions. They can vary significantly with changes in wind speed, direction, turbulence intensity, and wind gusts. As a result, different forecasting methods may be more suitable for predicting aerodynamic force behaviours in specific scenarios.

In steady inflow conditions, where wind speed and direction remain nearly constant over time, aerodynamic forces experience minimal fluctuations. Conversely, wind gusts, which are short-term increases in wind speed, can cause sharp variations in aerodynamic forces. Similarly, highly turbulent inflow conditions, characterised by rapid changes in wind speed and direction, result in complex, stochastic variations in aerodynamic forces.

Constant, linear, and polynomial regression techniques are expected to be most effective in scenarios where aerodynamic forces change gradually or follow predictable trends. These methods are anticipated to work well under steady inflow conditions, where fluctuations in aerodynamic forces between fluid time steps are minimal, and even less so between finer structural time steps (see fig. 2.18). Additionally, these techniques are straightforward to implement and computationally efficient compared to solving the full aerodynamic problem.

In more dynamic inflow conditions, where aerodynamic forces may exhibit complex variations, models like ARIMA could potentially capture both deterministic trends and random fluctuations. By leveraging autoregressive components and moving average terms, ARIMA models might adapt better to variable conditions. However, they tend to be computationally more demanding than simpler regression-based approaches.

Chapter 3

Free vortex wake method implementation process

Wind turbine design and performance analysis often rely on simple, computationally efficient modelling methods such as the Blade Element Momentum (BEM) method [37]. During the design phase, a large number of Design Load Cases (DLCs) must be validated for which using computationally intensive modelling tools is not feasible within the industry's time constraints. As a result, faster, more efficient methods like BEM are favoured to ensure that all necessary DLCs can be evaluated within a reasonable time frame. However, BEM's basic assumptions, such as uniform, steady inflow and load evaluation restricted to the rotor plane, make it inadequate for the industry's latest developments where rotor diameters now exceed $200m$ [109, 110]. Large floating offshore wind turbines, for instance, experience unsteady inflow conditions induced by the motion of their floating foundations, which must be accounted for to accurately evaluate aerodynamic loads. The Free Vortex Wake (FVW) method [58, 111, 112, 113, 114, 115, 116] addresses these challenges by accounting for phenomena such as non-uniform inflow, unsteady blade-element behaviour, and blade-wake interactions.

The FVW method provides a more physically accurate representation of the wake by modelling it as a system of discrete vortices shed from the blades. This makes FVW a promising tool for wind turbine design, but its advantages are tempered by the high computational cost. One of the key strengths of vortex methods is their ability to resolve the temporal and spatial evolution of the wake by shedding discrete vortical structures, such as vortex particles, filaments or other vorticity carrying structures. However, tracking these structures to evaluate aerodynamic loads adds significant computational cost, especially when integrated into aeroelastic simulations. In cases where aerodynamic and structural solvers operate on different time steps and use partitioned coupling, the structural solver can limit the fluid's integration time step, increasing the rate of vortical structure emissions, as discussed in section 2.2.2. Despite these challenges, methods such as subcycling, wake reduction or merging techniques, and GPU acceleration offer potential solutions to mitigate computational costs [117, 118], bringing vortex methods closer to being viable tools for wind turbine design.

This chapter explores the fundamental principles of Free Vortex Wake methods,

from their theoretical foundations to their application in modern wind turbine analysis, with a focus on lifting-line-based approaches and vortex filament discretised wakes. The chapter is structured as follows: the first section presents the theoretical background, followed by two sections dedicated to GPU-accelerated FVW solvers: CASTOR, a C++ code that was already available for this work, and Pitchou, a Python code developed during this thesis. Pitchou aims to provide readers with insights into the implementation of such methods. The final section presents a sensitivity analysis of the wake, highlighting the significant impact of wake discretisation on aerodynamic force evaluation when using free vortex wake methods.

3.1 Theory of vortex methods

3.1.1 Navier Stokes equations: ($\rho-u$) and ($u-\omega$) formulations

The Navier-Stokes (NS) equations for an incompressible flow are first written in their pressure-velocity form, where the velocity vector and pressure scalar fields are respectively $\vec{u} = \vec{u}(\vec{x}, t)$ and $P = P(\vec{x}, t)$, where $\vec{x}$ is the position vector and t the time. The conservation of mass and momentum equations, for a fluid not subjected to any outside force, are given respectively by:

$$\vec{\nabla} \cdot \vec{u} = 0, \tag{3.1}$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \vec{\nabla}\vec{u} = -\frac{1}{\rho}\vec{\nabla}P + \nu\vec{\nabla}^2\vec{u}, \tag{3.2}$$

where ρ is the fluid density and ν its kinematic viscosity. In order to rewrite these equations in a velocity-vorticity formulation, the following expression relating the vorticity $\vec{\omega}$ and the velocity fields is given:

$$\vec{\omega} = \vec{\nabla} \times \vec{u}. \tag{3.3}$$

By taking the curl of eq. (3.2), the velocity-vorticity form of the Navier-Stokes equations are given:

$$\vec{\nabla} \cdot \vec{u} = 0, \tag{3.4}$$

$$\frac{\partial \vec{\omega}}{\partial t} + \left(\vec{u} \cdot \vec{\nabla}\right)\vec{\omega} = \left(\vec{\omega} \cdot \vec{\nabla}\right)\vec{u} + \nu\Delta\vec{\omega}. \tag{3.5}$$

By performing operator splitting in eq. (3.5), the NS equation can be broken down into a convection and a diffusion equation, respectively given by:

$$\frac{\partial \vec{\omega}}{\partial t} + \left(\vec{u} \cdot \vec{\nabla}\right)\vec{\omega} = \left(\vec{\omega} \cdot \vec{\nabla}\right)\vec{u}, \tag{3.6}$$

$$\frac{\partial \vec{\omega}}{\partial t} = \nu\Delta\vec{\omega}. \tag{3.7}$$

In this work, an inviscid flow is assumed, meaning only eq. (3.6) is used. Initially presented in its Eulerian form, it can be reformulated in a Lagrangian framework using the particle derivative $\frac{D}{Dt} = \frac{\partial}{\partial t} + (\vec{u} \cdot \vec{\nabla})$, as follows:

$$\frac{D\vec{\omega}}{Dt} = \left(\vec{\omega} \cdot \vec{\nabla}\right) \vec{u}. \tag{3.8}$$

3.1.2 Biot-Savart law

The convection equation discussed in the previous section requires evaluating the velocity field. This is here obtained following the work of [119] also dscribed in [120, 121, 122, 123, 124]. According to Helmholtz's theorem [125], the velocity field $\vec{u}$ can be decomposed into two components: the potential velocity $\vec{u}_\Phi$ and the vorticity-induced velocity $\vec{u}_\Psi$, as follows:

$$\vec{u} = \vec{u}_\Phi + \vec{u}_\Psi = \vec{\nabla}\Phi + \vec{\nabla} \times \vec{\Psi}, \tag{3.9}$$

where Φ is a scalar potential field and $\vec{\Psi}$ is a divergence-free vector potential field such that $\vec{\nabla} \cdot \vec{\Psi} = 0$.
By taking the curl of the velocity field defined by eq. (3.9), one gets:

$$\begin{aligned} \vec{\nabla} \times \vec{u} &= \vec{\nabla} \times \vec{u}_\Phi + \vec{\nabla} \times \vec{u}_{\vec{\Psi}} \\ &= \underbrace{\vec{\nabla} \times (\vec{\nabla} \cdot \Phi)}_{=\vec{0}} + \vec{\nabla} \times (\vec{\nabla} \times \vec{\Psi}), \end{aligned} \tag{3.10}$$

where the vorticity appears as the solenoidal component of the velocity field.
By extending eq. (3.10), such that:

$$\begin{aligned} \vec{\omega} &= \vec{\nabla} \times (\vec{\nabla} \times \vec{\Psi}) \\ &= \vec{\nabla} \underbrace{(\vec{\nabla} \cdot \vec{\Psi})}_{=0} - \vec{\nabla}^2 \vec{\Psi}, \end{aligned} \tag{3.11}$$

the divergence-free vector potential leeds to the following Poisson equation [126]:

$$\Delta \vec{\Psi} = -\vec{\omega}. \tag{3.12}$$

If G defines Green's function of the Laplacian operator, given in 3D by:

$$G(\vec{x}) = \frac{1}{4\pi \|\vec{x}\|}, \tag{3.13}$$

the solution to eq. (3.12) is a convolution product ($\circledast$) of G with $\vec{\omega}$, such that:

$$\vec{\Psi}(\vec{x}, t) = G(\vec{x}) \circledast \vec{\omega}(\vec{x}, t) = \iiint_{\vec{y} \in \mathbb{R}^3} G(\vec{x} - \vec{y}) \vec{\omega}(\vec{y}, t) \, \mathrm{d}v(\vec{y}), \tag{3.14}$$

where $dv(\vec{y})$ is the volume of integration. By taking the curl of the above equation, one gets the expression for the vorticity induced velocity, also known as the Biot-Savart law:

$$\vec{u}^{\Psi}(\vec{x},t) = \vec{\nabla} \times \vec{\Psi}(\vec{x},t) = \iiint_{\vec{y}\in\mathbb{R}^3} \vec{K}(\vec{x}-\vec{y}) \times \vec{\omega}(\vec{y},t)\, \mathrm{d}v(\vec{y}), \tag{3.15}$$

where $\vec{K}(\vec{x})$ is the Biot-Savart kernel defined by:

$$\vec{K}(\vec{x}) = \vec{\nabla} G(\vec{x}) = -\frac{1}{4\pi}\frac{\vec{x}}{\|\vec{x}\|^3}.$$

The kernel has a singular behaviour when $\|\vec{x}\| \to 0$, which is avoided through the use of regularised kernels [122, 113]. Some of the available kernels in CASTOR and their effects on aerodynamic forces computations will be presented in sections 3.1.3.3 and 3.4.1.

3.1.3 Filament based wake discretisation

In the wake, vorticity can be discretised using vortex particles or vortex filaments. When vortex filaments are used, the stretching term in eq. (3.6) becomes numerically difficult to handle. To avoid this complexity, vortex filament discretisations typically solve a transport equation for circulation, which eliminates the need to directly account for the stretching term.
The general derivation of this approach will be given next following the work of [113] and more detailed presentations can be found in works such as [127, 128]. In [113], the rate of change of a material line element ($\mathrm{d}\vec{r}$) is given by:

$$\frac{D(d\vec{r})}{Dt} = \mathrm{d}\vec{r} \cdot \vec{\nabla}\vec{u}. \tag{3.16}$$

By comparing this with the Lagrangian form of the Navier-Stokes equation in its vorticity-velocity form (eq. (3.8)), it can be observed that, in a circulation-preserving flow, vortex lines can be treated as material lines that evolve with the flow, meaning fluid elements on a vortex line remain on that line as it evolves [113], which is defined as Helmholtz second theorem. From this follows a property for vortex lines and tubes, stating that they must either form closed curves or end at the boundaries of the fluid [113].
Kelvin introduced a method to describe the motion of vorticity-carrying fluid elements by focusing on the circulation around a closed curve. By applying the definition of circulation to a line around the cross-section of a vortex tube, one gets:

$$\Gamma = \int_L \vec{u} \cdot \mathrm{d}\vec{r}, \tag{3.17}$$

directly leading to:

$$\begin{aligned}\frac{D\Gamma}{Dt} &= \frac{D}{Dt}\int_L \vec{u} \cdot \mathrm{d}\vec{r} \\ &= \int_L \frac{D\vec{u}}{Dt} \cdot \mathrm{d}\vec{r} + \int_L \frac{D\mathrm{d}\vec{r}}{Dt} \cdot \vec{u}.\end{aligned}$$

By taking the expression for rate of change of a material line element in eq. (3.16) and since material lines are tracked, the second term on the right hand side of the above equation can be rewritten as:

$$\int_L \frac{D\mathrm{d}\vec{r}}{Dt} \cdot \vec{u} = \int_L \mathrm{d}\vec{r} \cdot \vec{\nabla}\vec{u} \cdot \vec{u} = \int_L \vec{\nabla}\left(\frac{1}{2}\vec{u} \cdot \vec{u}\right) \cdot \mathrm{d}\vec{r}. \tag{3.18}$$

Since the line integral of the gradient of a scalar field over a closed loop is always zero, one gets:

$$\frac{D\Gamma}{Dt} = \int_L \frac{D\vec{u}}{Dt} \cdot \mathrm{d}\vec{r}. \tag{3.19}$$

By injecting the velocity-pressure formulation of the Navier-Stokes equation in eq. (3.2) into the previous equation, one gets:

$$\frac{D\Gamma}{Dt} = -\int_L \vec{\nabla}P \cdot \mathrm{d}\vec{r} + \nu \int_L \Delta\vec{u} \cdot \mathrm{d}\vec{r} = 0. \tag{3.20}$$

Since an inviscid flow is considered, the second term in the above equation becomes zero. The first term also vanishes because it represents the line integral of the pressure gradient over a closed loop, which equals zero. Equation (3.20) is Kelvin's theorem, describing the evolution of circulation of fluid elements, with no stretching term present in the equation. This equation also requires evaluating the velocity field, for which the associated Biot-Savart equation in terms of circulation is given in [128].

3.1.3.1 Lifting line, circulation, effective velocity and wake node positions

The blades of a wind turbine are a source of circulation. The total circulation of the system, can be split into two components: the circulation of the bound filaments located on the lifting line of the blade and the circulation of the wake filaments. The circulation of the bound filaments can be derived by equating the lift expression from blade element theory with Kutta-Joukowski's theorem:

$$L = \frac{1}{2}\rho|\vec{u}_{eff}|^2 c C_L \quad , \quad L = \rho|\vec{u}_{eff}|\Gamma_{bound},$$

where $\vec{u}_{eff}$ is the sectional effective velocity of the flow, c is the chord length of the airfoil and C_L is the lift coefficient from the airfoil's lift polar curve. From this, the bound circulation can be expressed as:

$$\Gamma_{bound} = \frac{1}{2}|\vec{u}_{eff}|cC_l. \tag{3.21}$$

The strength of the wake filaments is derived using Kelvin's theorem, which states that the total circulation in an inviscid system remains constant over time [59], as discussed in the previous section. According to Kelvin's theorem, any change in the bound circulation along the blade is balanced by the generation of circulation in the

wake, ensuring the preservation of total circulation in the system. The wake is discretised into shed and trailing filaments, and Kelvin's theorem provides expressions for the shed and trailing circulations as functions of the bound circulation:

$$\Gamma_{trail}(r,t) = \Gamma_{bound}(r,t) - \Gamma_{bound}(r - \Delta r, t),$$

$$\Gamma_{shed}(r,t) = \Gamma_{bound}(r,t) - \Gamma_{bound}(r, t - \Delta t),$$

where r represents the radial position along the blade and t is the time at which the circulation is evaluated.
Once the circulation of the system is determined, the effective velocity ($\vec{u}_{eff}$) can be calculated. The effective velocity represents the velocity experienced by each blade element, which is necessary for determining the aerodynamic forces acting on the blade and for updating the positions of the wake nodes. The effective velocity is a combination of three components: the wind velocity, the geometrical velocity due to the blade's rotation, and the induced velocity generated by the wake and can be expressed as follows:

$$\vec{u}_{eff} = \vec{u}_{wind} + \vec{u}_{geom} + \vec{u}_{induced}. \tag{3.22}$$

In both solvers used in this work, the wake nodes are updated using an explicit Euler time-marching approach. This method uses the effective velocity at the current time step to update the position of each Lagrangian marker. The position update for each marker is given by:

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{u}_{eff}(t)\Delta t. \tag{3.23}$$

This explicit Euler formulation updates the position $\vec{x}(t + \Delta t)$ of the Lagrangian markers based on their current position $\vec{x}(t)$ and the effective velocity at time t, moving the markers downstream in the wake.
All of these computations, including induced velocity, bound circulation, aerodynamic forces, and the advection of wake nodes, are performed iteratively at each time step. This iterative procedure will be described in further detail in a later section, presented in algorithm form to provide a clear overview of the computational process.

3.1.3.2 Time marching method: wake update

Figure 3.1 illustrates the discretisation of the wake using the filament-based approach. At the beginning of the wake, trail and shed filaments are generated from the trailing edge nodes of the turbine blades. The trailing filaments represent a form of spatial discretisation, with their strength determined by the radial variation in circulation between two adjacent bound vortices along the blade. Meanwhile, the shed filaments capture the time evolution of the wake, as their circulation is governed by the azimuthal change in bound vortices, which varies over time as the blades rotate. Each of the filaments is interconnected through Lagrangian markers, which are identified in 3.1 as wake nodes. Together, these markers form a vortex sheet that, at each time step, is convected downstream and deforms based on the mutual

interactions of all vortex elements in the wake. The motion and deformation of the wake are determined by the induced velocities, calculated from the influence of each wake filament on each wake node. To accurately update the wake in a time-marching manner, two key components must thus be computed at each time step: the bound circulation, which is essential for determining the strength of the new trailing and shed vortices that will be emitted as it defines the characteristics of the near wake and the induced velocities at the wake nodes which is crucial for updating the wake node positions, allowing the wake to be convected and deformed.

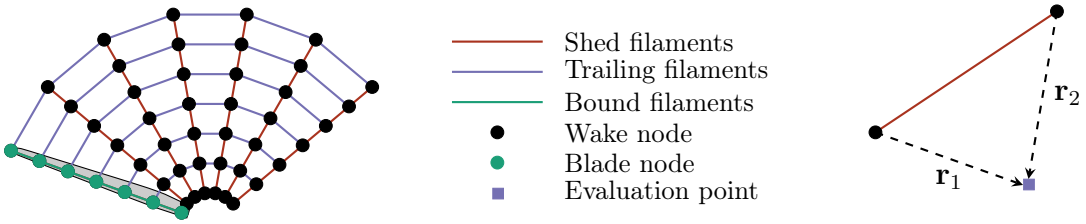


Figure 3.1: Schematic representation of the wake and notations for induction calculations with $\mathbf{r}_1$, $\mathbf{r}_2$ the vectors pointing from the filament end nodes to the evaluation point. [1]

3.1.3.3 Induced velocity and desingularisation methods

As stated previously, the FVW method is based on studying the influence of each vortex element on others through the evaluation of induced velocities. In section 3.1.2, the Biot-Savart law was introduced through eq. (3.15) for the induced velocity computation and can be rewritten as follows, according to the filament discretization method presented in the previous part:

$$\vec{u}_{induced}(\vec{x}_p) = \frac{\Gamma}{4\pi} \frac{(|\vec{r_1}| + |\vec{r_2}|)(\vec{r_1} \times \vec{r_2})}{|\vec{r_1}||\vec{r_2}|(|\vec{r_1}||\vec{r_2}| + \vec{r_1} \cdot \vec{r_2})}, \tag{3.24}$$

where Γ is the total vortex strengths and where $\vec{r_1}$ and $\vec{r_2}$ represent the position vectors pointing from the end nodes of a filament to the evaluation point, as shown in fig. 3.1.

The problem discussed in the previous section regarding the computation of induced velocity arises when the evaluation point approaches too closely to the vortex filament. This issue becomes particularly pronounced when a fine time step is used, leading to a high emission rate of filaments and increasing the likelihood of the evaluation point being near a filament. When this occurs, the terms $|\vec{r_1}|$ and $|\vec{r_2}|$ in eq. (3.24) tend toward zero. This causes a singularity in the induced velocity calculation, where the velocity becomes theoretically infinite leading to an unphysical behavior. To resolve this and ensure stable and realistic wake simulations, desingularisation methods are applied to smooth out the singularity and provide finite, accurate velocity values even in close proximity to the filaments.

Several desingularisation techniques have been presented in the literature, such as the Lamb-Oseen and Rankine models [87]. However, in this work, the focus is on two desingularization methods available in CASTOR, namely the Offset method and the Vatistas method [129].

The Offset method desingularises the Biot-Savart law by adding a corrective term to the denominator, which prevents the induced velocity from becoming infinite:

$$\vec{u}_{induced}(\vec{x}_p) = \frac{\Gamma}{4\pi} \frac{(|\vec{r_1}| + |\vec{r_2}|)(\vec{r_1} \times \vec{r_2})}{|\vec{r_1}||\vec{r_2}|(|\vec{r_1}||\vec{r_2}| + \vec{r_1} \cdot \vec{r_2}) + (r_c l_0)^2}. \tag{3.25}$$

In this equation, r_c represents the core radius of the filament, and l_0 is the filament length. This method is one of the simplest approaches to desingularizing the Biot-Savart law, effectively smoothing the velocity profile near the filament by introducing a cutoff radius.
The Vatistas second-order model introduces a more sophisticated correction based on a normalized factor $K(\rho)$ where ρ is the radial distance to the vortex line, defined as $\rho = |\vec{r_1} \times \vec{r_2}|/l_0$. The induced velocity is then expressed as:

$$\vec{u}_{induced}(\vec{x}_p) = K(\rho)\frac{\Gamma}{4\pi} \frac{(|\vec{r_1}| + |\vec{r_2}|)(\vec{r_1} \times \vec{r_2})}{|\vec{r_1}||\vec{r_2}|(|\vec{r_1}||\vec{r_2}| + \vec{r_1} \cdot \vec{r_2})}, \tag{3.26}$$

where $K(\rho)$ is defined as:

$$K(\rho) = \frac{\frac{\rho^2}{r_c^2}}{\sqrt{1 + \frac{\rho^4}{r_c^4}}}.$$

This method provides a more gradual desingularisation effect, adjusting the induced velocity based on the radial distance from the vortex line.
In section 3.4.1, the effects of these different desingularisation methods on the wake's discretisation will be studied and discussed. The next section will present the methods used to compute the total circulation in the system, which is a key element for calculating the induced velocities.

3.1.4 Outcomes

This first part of the chapter aimed to present the theoretical foundation of Free Vortex Wake methods using lifting line theory with filament wake discretisations. The key components necessary for aerodynamic force computations were discussed, with particular emphasis on the interdependent roles of induced velocity and bound circulation. Together, these elements enable the computation of aerodynamic forces acting on the blades while allowing the update of the wake filaments' positions and strengths. In the next part, the focus will be set on the iterative process and the algorithms used in the FVW solvers, which integrate the computations of induced velocity, bound circulation, and wake evolution. As wake handling is the most computationally intensive aspect of FVW methods, acceleration techniques will also be introduced to optimise performance.

3.2 CASTOR: a GPU accelerated FVW code

The Free Vortex Wake (FVW) lifting-line flow solver, CASTOR, is a C++ code designed for modelling various wind turbine geometries, including horizontal and vertical axis rotors, as well as multi-rotor configurations [71, 130]. The implementation is based on the work of [41], and the core algorithm will be presented in the following section. The subsequent section presents additional functionalities of the code that further reduce computational costs, complementing the GPU acceleration features already in place.

3.2.1 General algorithm

The main algorithm and its steps are detailed in algorithm 1. As previously mentioned, CASTOR is a GPU-accelerated FVW solver, although not all stages of the algorithm are ported to the GPU. The specific steps that are GPU-accelerated will be indicated accordingly. The process begins with the emission of a new set of shed and trailing filaments from the blades, where they are initialised with zero circulation. The trailing filaments are attached at one end to the blade nodes and at the other to previously advected wake markers. Following this initialisation, the blade and wake's filament information is copied to the GPU. Next, the inductions from wake filaments on both wake and blade nodes are computed. This step is computationally intensive as it involves solving an n-body problem, where the interaction between all vortex filaments must be calculated. To improve efficiency, this process is performed on GPU. First, the wake and blade node data are transferred to the GPU, where the Biot-Savart law (see eq. (3.25)) is applied to compute the induced velocities. Once these calculations are completed, the induced velocity data are copied back to the CPU, where the remaining steps of the algorithm, which are less computationally demanding, are carried out. Once the induced velocities are known, the bound circulation can be computed according to Kutta-Joukowski's theorem (eq. (3.21). To avoid divergence in the numerical algorithm, a convergence process is undertaken. This iterative loop involves the following steps: first, the induced velocities from the new filaments on the lifting line are computed using the updated bound circulation as well as trail and shed circulations obtained in the previous convergence sub-step. With the newly computed induced velocities, an updated bound circulation can be derived, and a relaxation procedure is applied:

$$\tilde{\Gamma}_b^{new} = \tilde{\Gamma}_b^{old} + \delta \left(\Gamma_b - \tilde{\Gamma}_b^{old} \right),$$

where $\tilde{\Gamma}_b^{old}$ is the bound circulation from the previous convergence step, $\tilde{\Gamma}_b^{new}$ is the bound circulation from the current sub-step and δ is a user defined relaxation parameter. Once the bound circulation is updated, the previously initialised shed and trailing filaments, which originally had zero circulation in the full iteration, can be revised accordingly. This updated information is then used to re-compute the induced velocities in the next sub-iteration.
After exiting the convergence loop, the positions of the Lagrangian markers are updated by advecting the wake through the induced velocities by applying eq. (3.23).

The last step of the algorithm involves applying accomodation techniques to the wake, which is meant to accelerate the computation, this will be discussed in the next part.

Algorithm 1: Vortex method algorithm overview

Data: Turbine geometry and kinematic: blades, tower, struts element data
Result: Turbine loads and wake geometry

Add a new row of shed and trail filaments in the wake
Copy filaments' information on GPU, clear GPU memory
Compute filament inductions on wake and blade nodes
    Copy wake and blade nodes information on GPU
    Apply Biot-Savart and perform the reduction at every wake node on GPU
    Copy the induced velocities back to CPU
while $|\tilde{\Gamma}_b^{new} - \tilde{\Gamma}_b^{old}| > \epsilon$ **do**
    Compute new induced velocities on the lifting line on CPU using updated $\tilde{\Gamma}_b^{new}$ and newly created Γ_s and Γ_t
    Compute Γ_b and apply relaxation to obtain $\tilde{\Gamma}_b^{new}$
    Update newly created Γ_s and Γ_t
end
Advect all wake nodes and apply any stretching corrections on CPU
Apply wake accommodation techniques on CPU

3.2.2 Wake accommodation methods for faster computing

It was previously discussed that coupling a FVW code with a servo-elastic solver can impose very restrictive time step requirements on the fluid code. These reduced time steps increase the number of wake filaments, which in turn makes solving the n-body problem more computationally intensive, as the algorithmic complexity scales with the square of the number of filaments.
In addition to the use of the GPU to accelerate the computation of induced velocities, CASTOR employs other wake management techniques to reduce computational intensity. Among these are tip vortex models and wake coarsening approaches, which help alleviate the computational cost. These methods are discussed in detail in [1] and their graphical representations including trail and shed filament merging methods as well as tip vortex models are presented in fig. 3.2.
The tip vortex model involves merging trailing filaments that have reached a specific "age", defined by the number of iterations since their release from the blade. This process simplifies the wake structure by reducing it to two remaining trailing filaments, with their radial positions determined by the local circulation at a user-defined filament age. Similarly, shed filaments are merged into a single filament after a certain number of iterations. While this method effectively reduces computational complexity, it is recommended for use primarily in regions of the wake further away from the blade as applying such a radical simplification to the near wake could lead to significant accuracy losses [1]. For regions closer to the blade, a more gradual wake coarsening method is advisable. This approach progressively merges two adja-

cent rows of trailing and shed filaments, allowing for a controlled reduction in the number of filaments while maintaining higher accuracy in the near-wake region. Both techniques significantly reduce the total number of wake filaments, making the overall computation time more affordable and closer to that achieved with the BEM method.

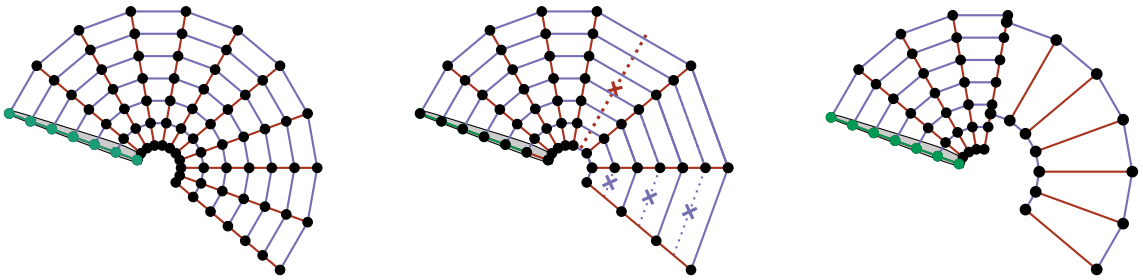


Figure 3.2: Schematic representation of the full wake model (left), the wake coarsening process with one shed- and one trail-merging steps (middle) and the tip-vortex model, with shed vorticity (right). Bound, shed and trailing filaments are green, red and blue, respectively [1].

3.3 Pitchou: FVW model in python

This section focuses on the FVW solver Pitchou, which was implemented throughout this work. Pitchou is developed in Python and, like CASTOR, is accelerated through GPU computations using the pyCUDA framework [131]. The development of Pitchou had two main objectives. First, it aimed to provide a deeper understanding of the Free Vortex Wake method and its implementation for wind turbine applications. More importantly, it serves as a user-friendly tool, enabling easy testing for simpler fluid-structure interaction (FSI) problems without the complexity and computational demands of using a more advanced aeroelastic solver like DeepLines Wind™. The following subsections will offer an overview of Pitchou's architecture, outlining the minimal components necessary for a functional vortex method code. Additionally, a detailed explanation of the validation process will be provided, covering test cases where Pitchou was validated against CASTOR, as well as theoretical and experimental data, for scenarios ranging from a static elliptical wing to a rotating wind turbine.

3.3.1 Code implementation

Pitchou follows the same algorithmic steps as those presented for the CASTOR solver. The aim was to develop a user-friendly Python-based solver that retains the core functionalities and behaviour of the original C++ implementation while maintaining minimal complexity. The overall architecture of Pitchou is illustrated in fig. 3.3. The code consists of four primary classes, which are responsible for creating objects related to the wind turbine, blades, airfoils, and wake. Additionally, there are three modules: the solver module, which executes all computations following the

steps outlined in Algorithm 1; the inductions module, which handles the computation of induced velocities; and the kernel module, which contains the GPU-accelerated portion of the code, implemented using PyCUDA. The different functionnalities and the way the different components work and interact will be discribed next:

- the **Airfoil** class manages aerodynamic data for airfoil sections of a blade. It reads polar data from a user defined file, creates interpolation functions for lift and drag coefficients based on the angle of attack (AOA), and provides methods ("getLift" and "getDrag") to compute lift and drag coefficients for any given AOA using linear interpolation techniques.

- the **Blade** class models all relevant information for the turbine blades, including nodes and centers along the blade span, their corresponding chord lengths, and orientation matrices that define the blade's pitch and twist. It also associates each blade section with an airfoil and tracks translation velocities for both nodes and centers, which are used to compute effective velocities. Additionally, the class manages bound circulations at blade nodes, from which trail and shed circulations are derived, and handles the wake nodes along with their associated inductions in a filament-based wake discretisation. The class provides methods for initializing wake nodes ("initializeWake"), updating circulation values for shed and trail filaments ("updateFilamentCirculations", "updateSheds", "updateTrails"), and splicing the near wake to propagate these values ("spliceNearWake"). It also provides functionality to compute and update the bound circulation ("estimateGammaBound"), advect wake filaments ("advectFilaments"), and extract filament node and circulation data for eventual vortex particle creation ("getLastFilamentsInfo").

- the **Wind Turbine** class handles the geometry of a multi-blade rotor. It manages a list of Blade objects, updating their positions, orientation, and translation velocities based on the current azimuthal position. Key methods include "initializeTurbine", which sets up the blade nodes, airfoils, and wake, and "updateTurbine", which recalculates the blades' geometries and velocities over time. An additional "evaluateForces" method computes aerodynamic forces on the blades based on their angles of attack and effective velocities.

- the **Wake** class is meant for managing wake particles (if a particle wake discretisation is chosen by the user). It stores the particles' positions, vorticity, and radii and includes methods to create particles from filaments ("addParticlesFromFilaments" if a hybrid filament/particle discretisation is chosen) and advect them based on wind and induced velocities ("advectParticles"). The class also uses JIT compilation [132, 133] for performance optimisation during particle creation and movement.

The Solver module defines the "update" function, which implements the time-marching algorithm outlined in 1. It relies on the "Inductions" module to compute induced velocities, with key functions like "wakeFilamentsInductionsOnBladeOrWake" for filament inductions on blade nodes and wake nodes, and "bladeInductionOnWakeFilaments" for calculating blade inductions on wake filaments. For

simulations using particle discretisation or a hybrid particle/filament approach, additional functions are available. All induction computations are accelerated by the Kernels module, leveraging GPU computing via PyCUDA. The CUDA kernels distribute the work across threads and blocks for efficient parallel processing. For instance, when computing blade induction on wake filaments, each thread index corresponds to a blade node, iterating over all wake filaments. This parallelisation allows simultaneous processing of all blade nodes, significantly speeding up the calculations compared to CPU methods. Once the induced velocities are computed, the "Solver" module updates bound circulation, shed and trailing filament strengths, and performs wake advection, all through functions defined in the "Blade" class.

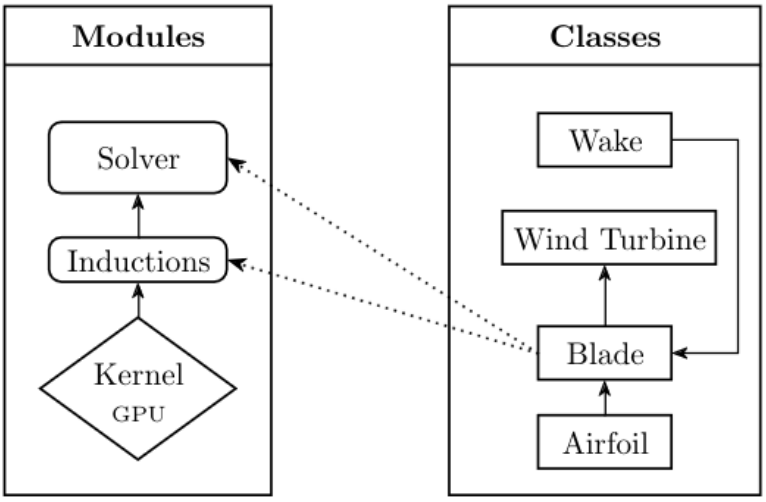


Figure 3.3: Architecture scheme for the FVW Pitchou solver.

3.3.2 Numerical validation

A series of test cases were implemented to validate the vortex method solver, Pitchou, including an elliptical wing and a three-bladed wind turbine based on the New Mexico model [134]. The following sections present results for both a static elliptical wing and one undergoing pitch changes, with comparisons made to analytical solutions, following the work of [41]. For the dynamic wind turbine case, the results are compared to those obtained with CASTOR and experimental data.

3.3.2.1 Static elliptical wing

The first test case involves an untwisted static elliptical wing, which is a practical example for validating the aerodynamic solver due to its constant induced angle of attack, leading to a uniform lift distribution along the wingspan. This case is compared to the analytical solution for lift given by [41]:

$$C_L = \frac{2\pi}{1 + \frac{2}{AR}}\alpha, \tag{3.27}$$

where α is the wing's angle of attack and AR represents the aspect ratio which can be expressed in terms of the root chord (c) and wingspan (b) as $AR = \frac{4b}{\pi c}$. This case aims to demonstrate the constant lift coefficient along the wingspan while evaluating the effects of varying cutoff radii using the Van-Garrel desingularisation model presented in section 3.1.3.3. For that matter, a simulation similar to the one in [41] was conducted. A 10-second simulation with a timestep of $\Delta t = 0.1s$ was run on an elliptical wing with 40 spanwise nodes, an aspect ratio of $AR = 6$, an inflow velocity of $U_\infty = 1$ m/s, and an angle of attack of $\alpha = 5°$. The results for cutoff radii ranging from $\delta^2 = 1$ to $\delta^2 = 0.001$ are shown in figure 3.4. The results indicate that for all cutoff radii, the computed lift coefficients indeed remain constant along the wingspan, converging to a value of $C_L \approx 0.41$ as $\delta^2 \to 10^{-3}$. This value closely matches the theoretical lift coefficient given by the analytical expression (3.27), labeled as the "reference value" in the graph, which falls between the cases for $\delta^2 = 0.1$ and $\delta^2 = 0.01$.

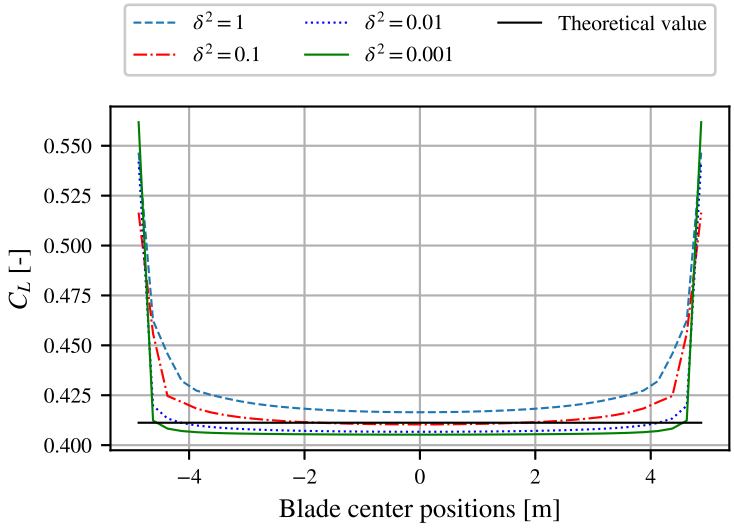


Figure 3.4: Lift coefficient for different cutoff radii function of blade center positions.

Wake analysis

A second tool used for validating the vortex solver is a qualitative indicator based on wake generation. When analyzing the vortices shed in the wake behind the elliptical wing, two key phenomena should be observed: tip vortices and starting vortices. Tip vortices, as the name suggests, form at the wing tips and result from the pressure gradient between the intrados and extrados surfaces. The higher pressure beneath the wing flows around to the lower pressure above, creating a swirling motion. The starting vortex, on the other hand, forms at the wing's trailing edge when the flow begins. This vortex is created to satisfy the conservation of vorticity, as the wing generates lift by inducing circulation around itself. The starting vortex carries an equal and opposite circulation to the bound vortex on the wing and is convected

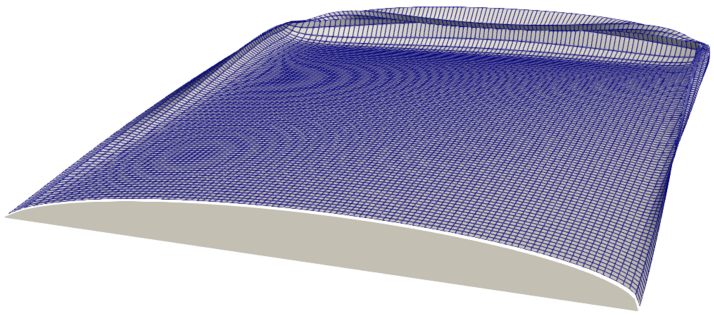


Figure 3.5: Static elliptical wing wake obtained through Pitchou.

downstream as the flow progresses. Figure 3.5 shows the wake after 6 seconds of simulation, clearly illustrating both the starting and tip vortices.

3.3.2.2 Elliptical wing undergoing pitch change

The second test case builds upon the static elliptical wing validation and extends it to a dynamic scenario where the wing undergoes a pitch change. This test case is essential for studying unsteady aerodynamic effects, which are accounted for through the shed filaments. The theoretical lift coefficient expression (3.27) is again used to verify if the aerodynamic coefficients are correctly evaluated when the system is in motion. For this case, an elliptical wing with an aspect ratio of $AR = 18$ and 40 spanwise nodes is simulated, undergoing a pitch change from $\alpha = 2°$ to $\alpha = 8°$. The inflow velocity is $U_\infty = 1$ m/s, with a total simulation time of $t = 20s$, and the pitch change imposed at $t = 10s$. Figure 3.6 shows the lift response to the pitch change: for the first 10 seconds, the lift coefficient converges to approximately $C_L = 0.2$, matching the theoretical value for $\alpha = 2°$. At the 10-second mark, a sharp increase in the lift coefficient is observed due to the pitch change, with the new value converging to $C_L = 0.8$ by the end of the simulation, with a slight overestimation compared to the theoretical value.

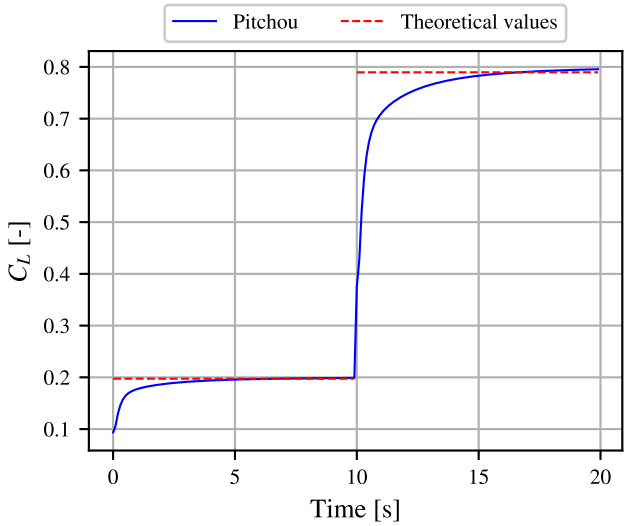


Figure 3.6: Lift response for elliptical blade undergoing pitch change.

3.3.2.3 Three-bladed rotor

The third case study extends the vortex code to simulate three-bladed horizontal axis wind turbines. A code-to-code comparison was performed against the FVW solver CASTOR. The validation test case is based on the "Mexico" rotor and "NewMexico" experiments [134], against which CASTOR has already been benchmarked in a prior study [71]. This simulation involves an axial inflow scenario with a wind velocity of 15 m/s, rotational velocity of 44.5 rpm, and blade pitch angle of $\theta = -2.3°$, with the wake discretised using vortex filaments. Figure 3.7 illustrates the lineic aerodynamic forces along the blades, showing good agreement between Pitchou, CASTOR, and the experimental data from the NewMexico tests. Both the normal force (F_n) and the tangential force (F_t) exhibit a good match across all data sources. Figure 3.8 presents the wake structure generated by the vortex filaments, clearly showing the inward rolling motion at the blade tips, consistent with both theoretical expectations and previous simulations, as observed in the elliptical wing case.

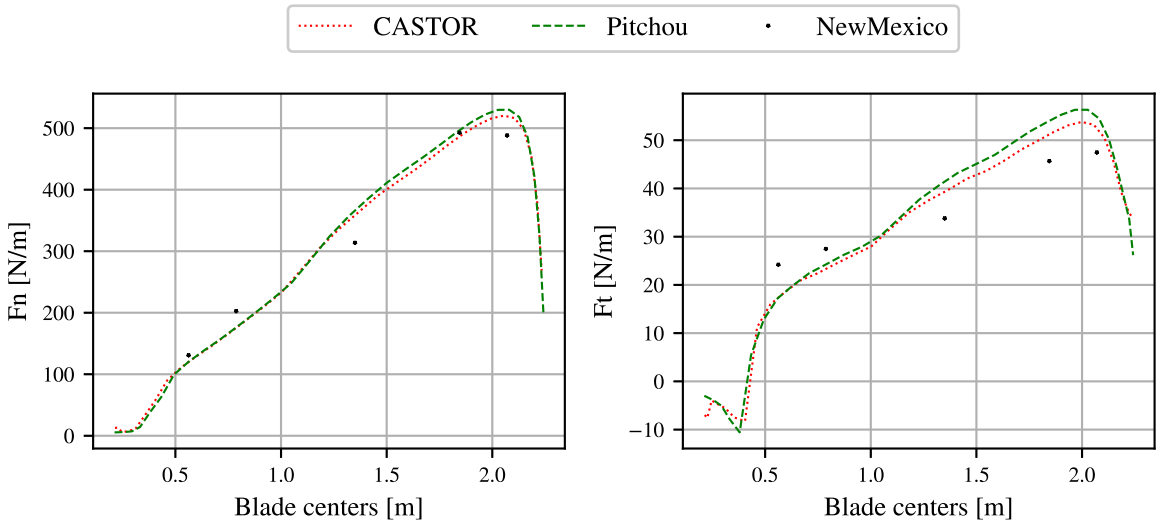
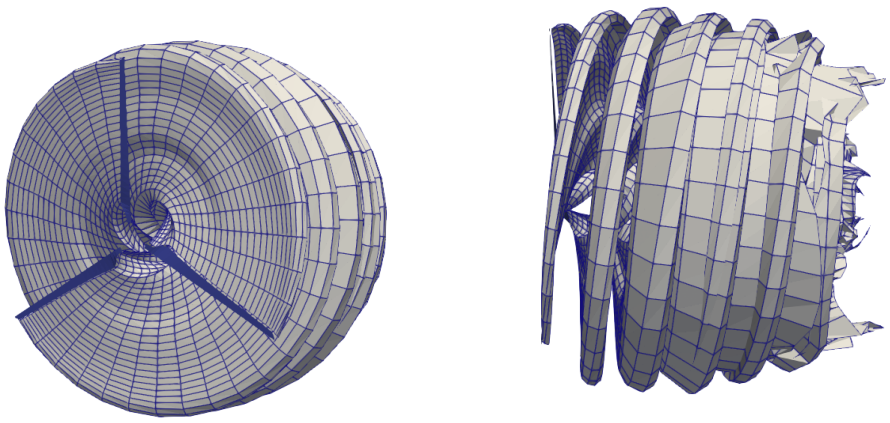


Figure 3.7: Code-to-code comparison of lineic (left) normal forces and (right) tangential forces function of blade center positions obtained by CASTOR and Pitchou.



(a) (b)

Figure 3.8: Front and side facing views of aerodynamic wake obtained through Pitchou.

3.3.3 Outcomes

The two vortex method solvers, CASTOR and Pitchou were presented, detailing the algorithmic processes involved and the necessary computational steps required

to accurately compute aerodynamic forces on wind turbine blades using lifting line methods and filament-based wake discretisations. To improve the efficiency of vortex methods and make them more comparable in computational time to Blade Element Momentum (BEM) methods, various wake merging techniques available in CASTOR were also introduced, highlighting approaches to reduce computational costs. Additionally, the development and validation of the Python-based solver Pitchou were discussed, showcasing its ability to produce results comparable to CASTOR, while offering a simpler and more user-friendly code. The next section will present a study on wake discretisations, analysing their sensitivity to time discretisation as well as to different desingularisation models. This will provide a deeper understanding of their influence and offer guidance on how to optimally configure these wake parameters for accurate computations.

3.4 Wake sensitivity in FVW methods

Wake modelling and and its discretisation in FVW methods are closely tied to the accuracy of computed aerodynamic forces. In free vortex wind turbine simulations, several factors can influence this accuracy, including the wake's length behind the rotor, its time discretisation, desingularisation methods, and filament core sizes. In [40], a link between the time discretisation of the wake and the vortex core radius was discussed, highlighting that a combination of a large vortex core size and a fine temporal discretisation can result in errors. Specifically, it was noted that a filament entering the core radius zone of a previously emitted filament could cause incorrect evaluation of the temporal variation of lift. To address these issues, the following sections will present an anlysis where the effects of these parameters were studied. The first section will begin by presenting the impact of CASTOR's desingularisation models and filament core sizes on induced velocity. This will be followed by an investigation into the combined effects of core sizes and temporal wake discretisation, with all studies carried out using the CASTOR solver.

3.4.1 Desingularisation models

Building on the work of [40], a simplified model consisting of one or two sequential vortex filaments was implemented to study the effects of CASTOR's desingularisation models on induced velocities. In fig. 3.9, the first two cases (left of fig. 3.9a and fig. 3.9b) feature a single vortex filament placed symmetrically around the origin, while the remaining cases involve two sequential filaments aligned along the origin. Two filament core sizes are considered, with $r_c = 0.02$ in fig. 3.9a and $r_c = 0.2$ in fig. 3.9b. When comparing the results for a single filament, it is evident that the Vatistas model produces a significantly higher maximum induced velocity than the Offset model, regardless of the core radius used. This behaviour is also observed with two filaments, where the Vatistas model continues to produce higher induced velocities, and the Offset model shows an additional spike at the transition between the filaments. These findings align with those reported in [40], where it was argued that such a spike should not occur, as the induced velocity from a single larger filament should match that of two consecutive smaller filaments, resulting in a smooth

transition between them. To further investigate these effects, the influence of desingularization models and filament core sizes will be evaluated in a more complex wind turbine simulation, providing insight into their impact in a realistic scenario.

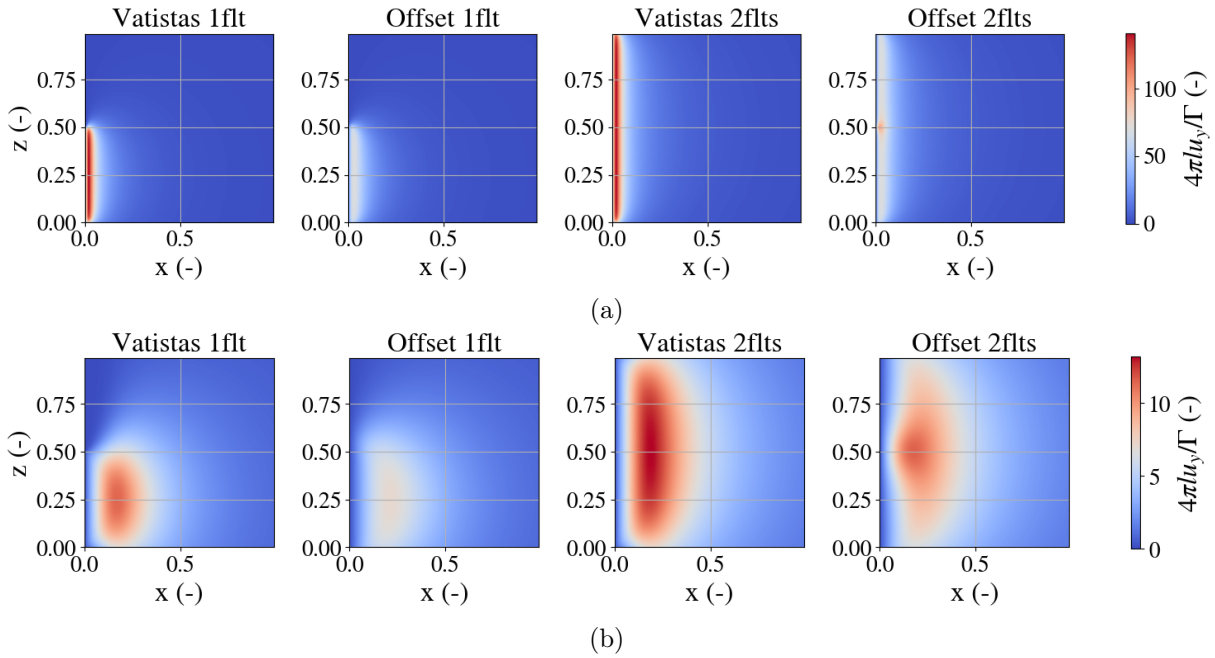


Figure 3.9: Wake using Offset desingularisation method with $r_c = 0.1$ for azimuth discretisations (a) $\delta_{Azi} = 1^{\circ}$ and (b) $\delta_{Azi} = 5^{\circ}$.

3.4.2 Effects of desingularisation model and time discretisation on wind turbine case

The impact of both time discretisation, referred to here as 'azimuth' discretisation (or δ_{Azi}), along with desingularisation models and cutoff radii, is analysed using a 15 MW IEA wind turbine model. In this case, the wind turbine operates under a steady wind condition of $10.6m/s$ with an average rotational velocity of $7.5rpm$. In the first scenario, the Offset and Vatistas models are applied with a constant cutoff radius of $r_c = 0.1$, while the azimuth discretisation varies from a fine resolution of $\delta_{Azi} = 1^{\circ}$ to more typical values suggested in the literature [109, 40], with $\delta_{Azi} = 5^{\circ}$ and $\delta_{Azi} = 10^{\circ}$. In the second case, the effects of a variable cutoff radius ranging from $r_c = 0.01$ to $r_c = 0.1$ are studied for both the Vatistas and Offset models, across all previously described azimuth discretisations.

3.4.2.1 Effects of azimuth discretisation

Figures 3.10 to 3.12 present the effects of azimuth discretisation on normal aerodynamic forces and the resulting wake. In particular, fig. 3.11a illustrates how the finest discretisation with $\delta_{Azi} = 1°$, combined with the Offset desingularisation model, causes overlapping of the filaments near the blade root. This overlapping can influence the accuracy of aerodynamic force computations, as reflected in fig. 3.10 (left), where the time variation of sectional normal forces at the blade's mid-span shows that the Offset model with a 1° discretisation produces less smooth force plots compared to coarser discretisations. This behaviour is not observed with the coarser discretization of $\delta_{Azi} = 5°$ when using the Offset model, nor is it present with the Vatistas model, regardless of whether fine or coarse azimuth steps are applied as shown in fig. 3.11. This leads to smoother time variations in sectional forces across all these cases. In fig. 3.10 (right), the distribution of lineic forces along the blade is shown, revealing that neither the azimuth discretisations nor the desingularisation model significantly affect the force distribution enough to distinguish the cases from one another. This contrasts with the results presented in section 3.4.1, where the Vatistas model produced higher induced velocities than the Offset model, potentially leading to higher aerodynamic forces.

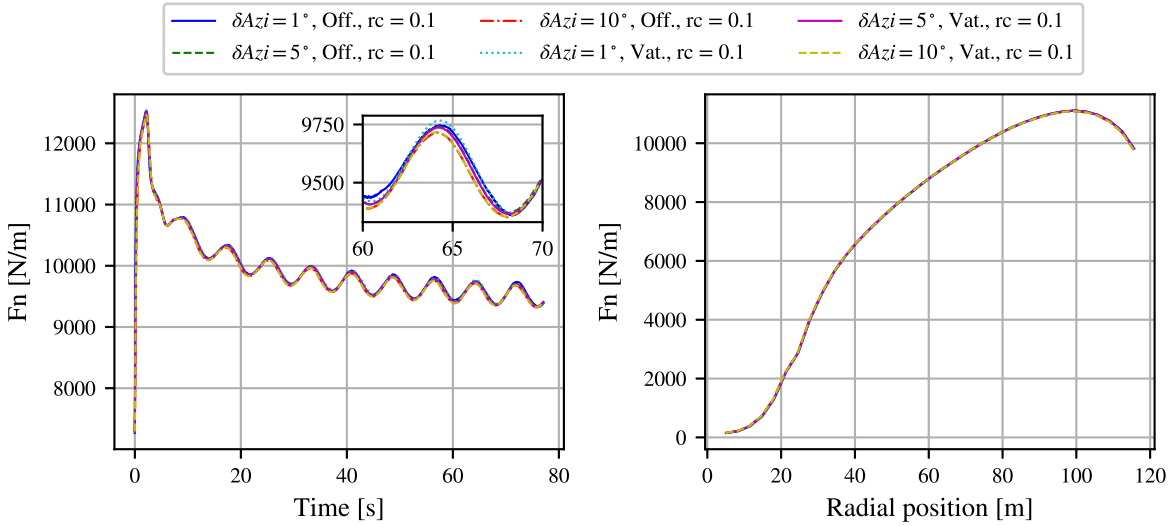
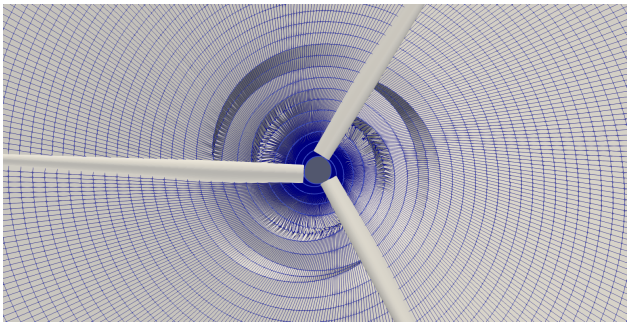
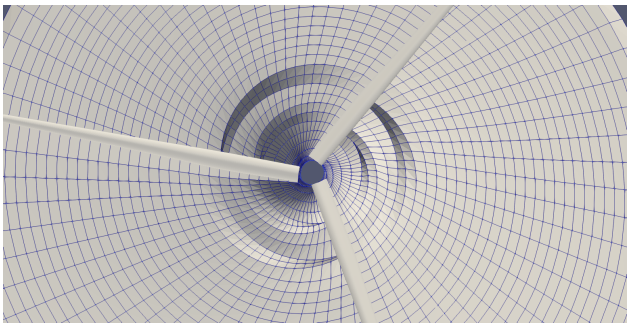


Figure 3.10: Effects of azimuth discretisations ranging from 1° to 10° using the Vatistas and Offset models with a constant cutoff radius $r_c = 0.1$: (left) time variation of sectional normal aerodynamic forces at the blade's mid-span, and (right) distribution of lineic normal aerodynamic forces along the blade's radial positions.

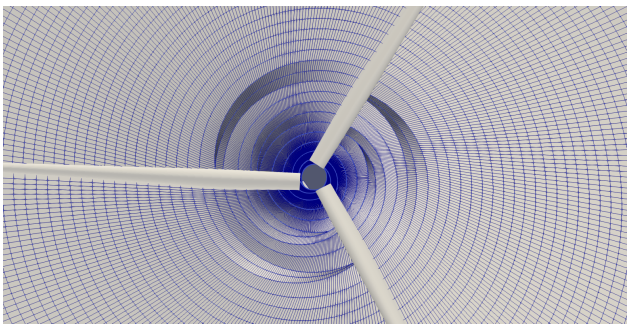


(a)

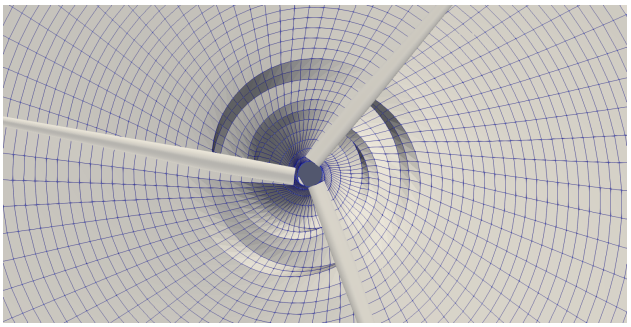


(b)

Figure 3.11: Wake using Offset desingularisation method with $r_c = 0.1$ for azimuth discretisations (a) $\delta_{Azi} = 1^{\circ}$ and (b) $\delta_{Azi} = 5^{\circ}$.



(a)



(b)

Figure 3.12: Wake using Vatistas desingularisation method with $r_c = 0.1$ for azimuth discretisations (a) $\delta_{Azi} = 1^{\circ}$ and (b) $\delta_{Azi} = 5^{\circ}$.

3.4.2.2 Effects of core sizes

Figures 3.13 to 3.17 examine the effects of varying core radii for both desingularisation methods with constant azimuth discretisation. The Offset model appears to be the most sensitive to core radius size. As shown in figs. 3.14 and 3.16 using a core radius of $r_c = 0.01$ significantly affects the wake for all azimuth steps, especially with finer azimuth discretisation, where filament overlap extends beyond the blade root and propagates throughout the wake span. This impact is also evident in the time variation of sectional forces, where the force plots become notably less smooth for all time discretisations when $r_c = 0.01$ is applied in the Offset model. In contrast, filament overlap with the Vatistas model, shown in figs. 3.15 and 3.17, is only mildly noticeable for $\delta_{Azi} = 1^\circ$ and $r_c = 0.01$, and this phenomenon is absent for larger azimuthal steps. For this reason, in subsequent parts when using the Offset method, a core radius of $r_c = 0.1$ will be applied. Since the Vatistas method is less sensitive to core size, it can operate with smaller core radii. In both cases, azimuth steps of $\delta_{Azi} = 5^\circ$ to $\delta_{Azi} = 10^\circ$ are recommended, as suggested in the literature [40].

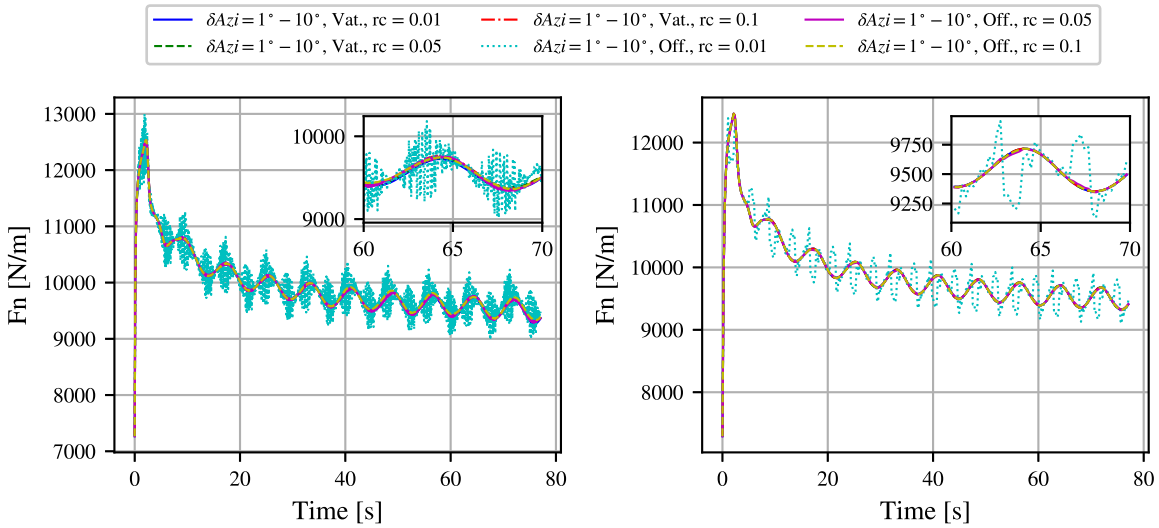
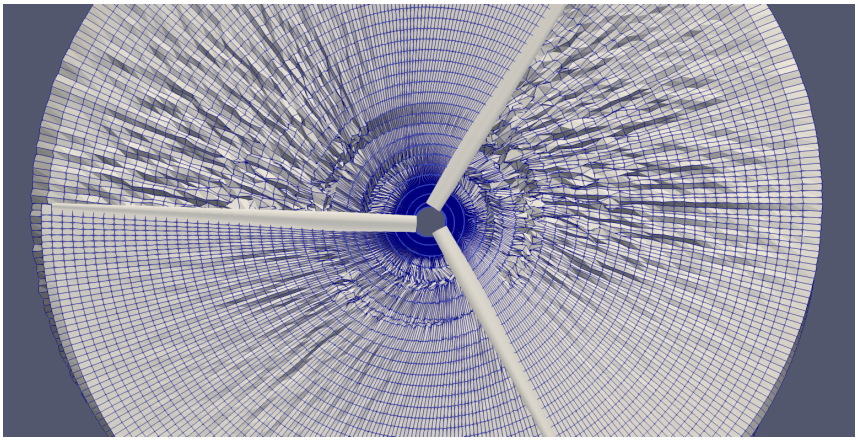
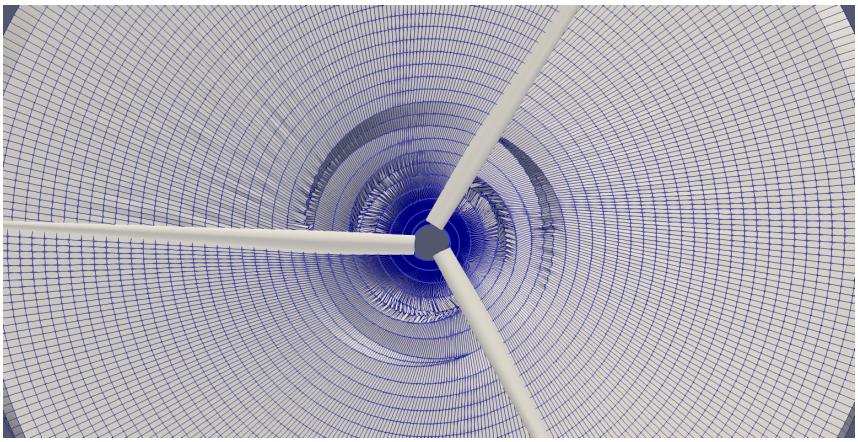


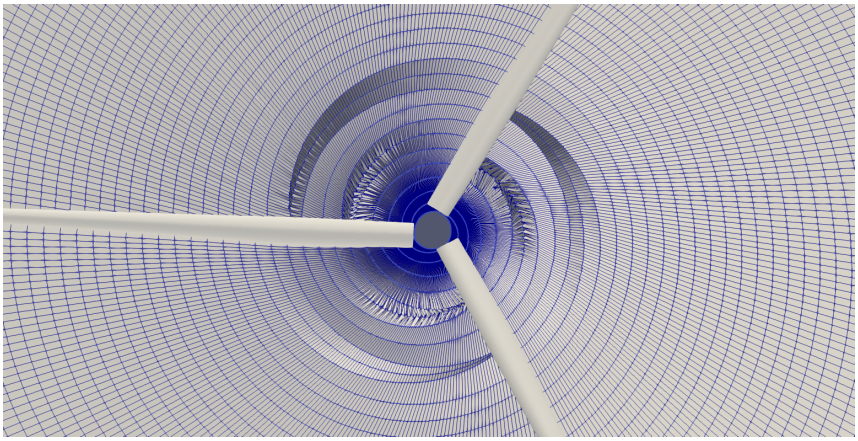
Figure 3.13: Effects of cutoff radius ($r_c = 0.01$ to $r_c = 0.1$) on the time variation of sectional aerodynamic forces at mid-span, using the Vatistas and Offset models with constant azimuth discretisation: (left) $\delta_{Azi} = 1^\circ$, and (right) $\delta_{Azi} = 10^\circ$.



(a)

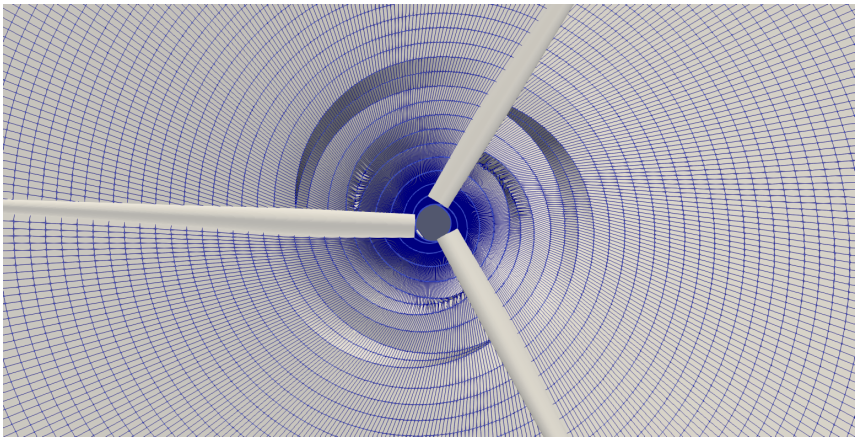


(b)

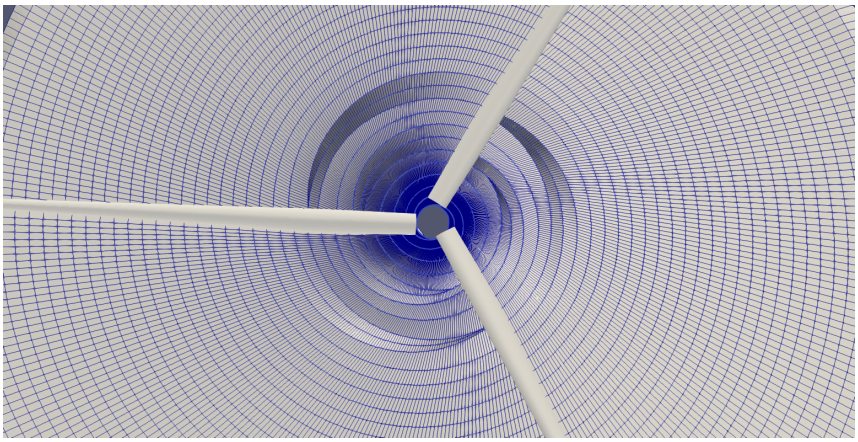


(c)

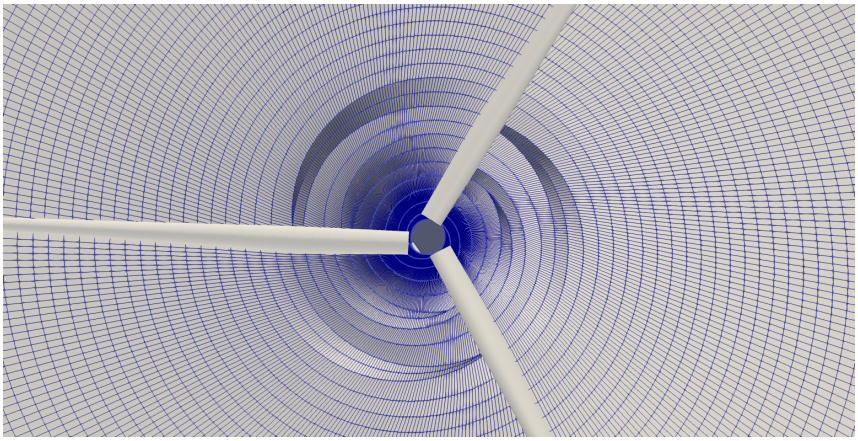
Figure 3.14: Wake for azimuth discretisation $\delta_{Azi} = 1^{\circ}$ using Offset model with: (a) $r_c = 0.01$, (b) $r_c = 0.05$ and (c) $r_c = 0.1$.



(a)

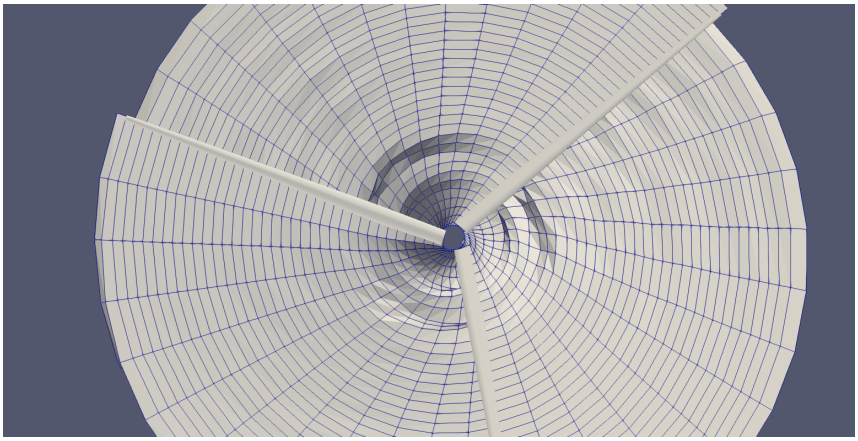


(b)

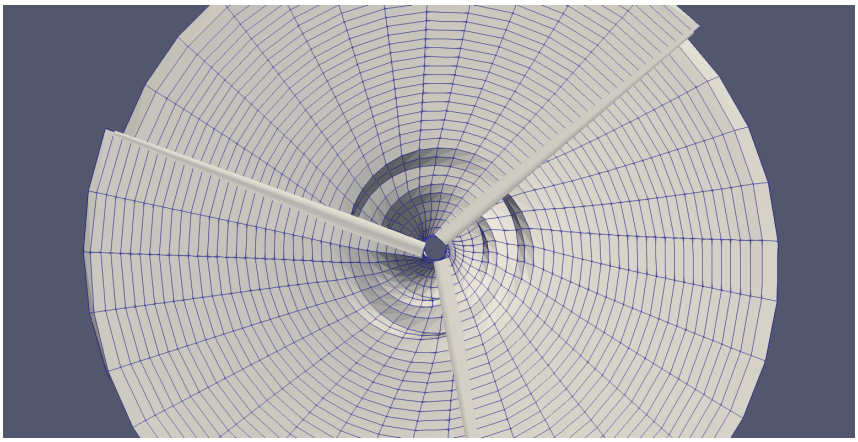


(c)

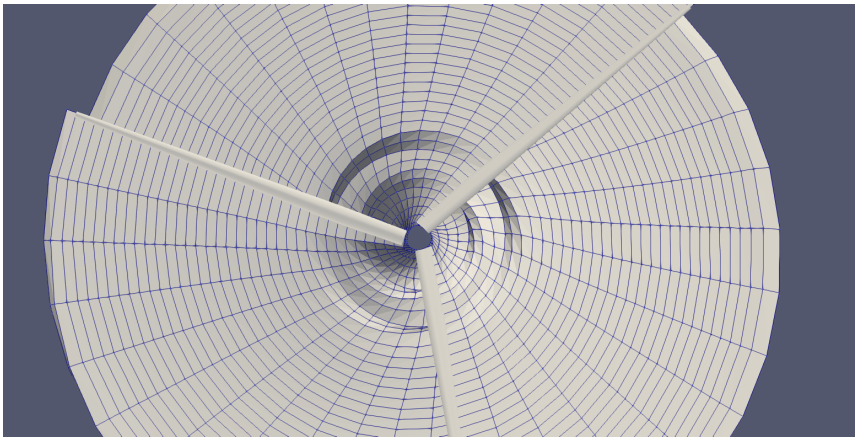
Figure 3.15: Wake for azimuth discretisation $\delta_{Azi} = 1^{\circ}$ using Vatistas model with: (a) $r_c = 0.01$, (b) $r_c = 0.05$ and (c) $r_c = 0.1$.



(a)

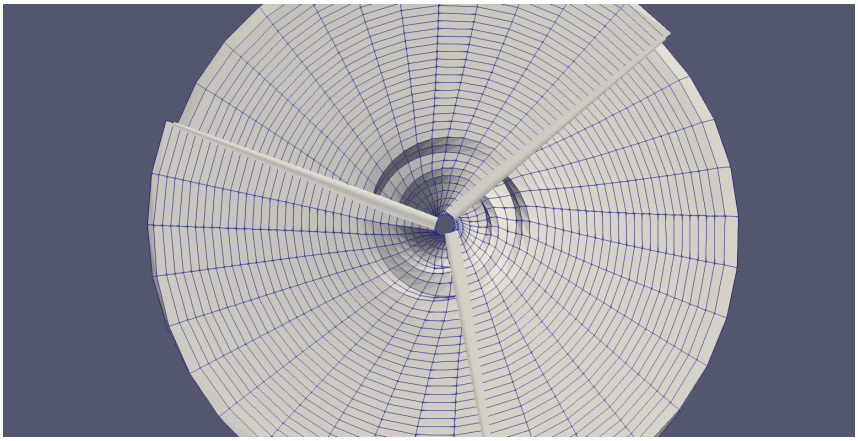


(b)

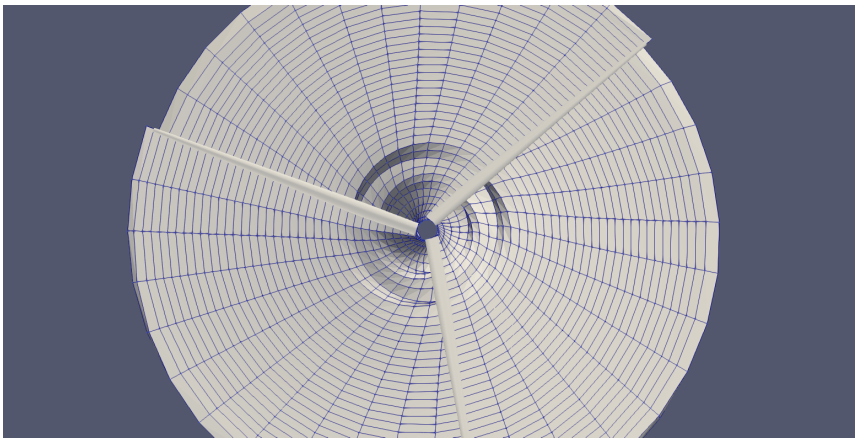


(c)

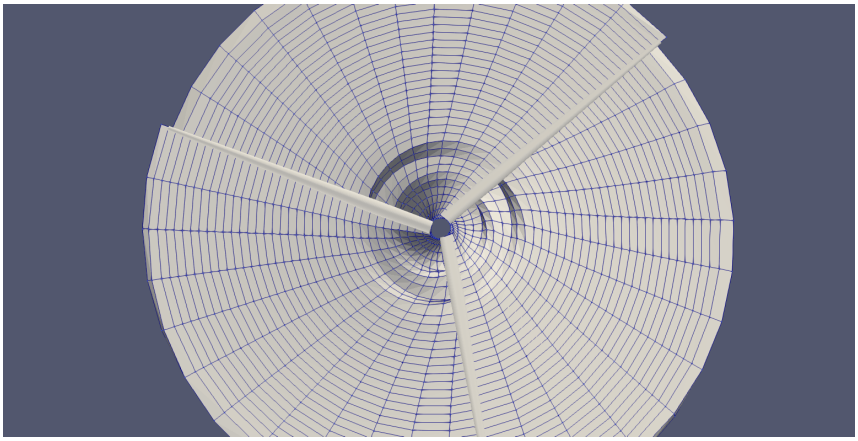
Figure 3.16: Wake for azimuth discretisation $\delta_{Azi} = 10°$ using Offset model with: (a) $r_c = 0.01$, (b) $r_c = 0.05$ and (c) $r_c = 0.1$.



(a)



(b)



(c)

Figure 3.17: Wake for azimuth discretisation $\delta_{Azi} = 10°$ using Vatistas model with: (a) $r_c = 0.01$, (b) $r_c = 0.05$ and (c) $r_c = 0.1$.

3.4.3 Outcomes

This chapter focused on the Free Vortex Wake (FVW) method, with particular emphasis on the lifting line theory using filament-based wake discretisations, which forms the foundation of both the FVM solvers CASTOR and Pitchou. The theoretical framework behind this approach was detailed in the first section, providing insight into its role in simulating aerodynamic behaviour. The second part of the chapter explored the numerical implementation of these methods, offering an algorithmic breakdown of the C++ solver CASTOR and a step-by-step description of the development process for the newly implemented Python-based solver, Pitchou. In addition to the theoretical and numerical discussions, this chapter presented a comparative study of the effects of azimuth discretisation and core radius variation on aerodynamic force predictions and wake behavior, particularly in the context of desingularisation models. This analysis highlighted the sensitivity of the Offset and Vatistas methods to core radius size and discretisation steps, providing insights that will guide the use of these models in the next chapter where FVW methods will be used in aeroelastic simulation purposes.

Chapter 4

Coupling schemes' effect on different coupled problems

This chapter analyses the effects of various coupling schemes introduced in chapter 2 on different model problems, aiming to clarify their impact on the behaviour and performance of aeroelastic system simulations. The first section focuses on a two-dimensional pitch-heave problem, providing a foundational example for fluid-structure interaction analysis. It includes a step-by-step solution using a simplified finite element method for the structural response, coupled with the Free Vortex Wake code, Pitchou, to simulate aerodynamic forces. The second section shifts to a more complex scenario, evaluating the influence of different coupling schemes on a large-scale wind turbine model. Specifically, the IEA 15MW reference wind turbine is modelled using the aero-servo-elastic tool DeepLines Wind™, with the use of the FVW tool CASTOR to simulate the aerodynamics. This section highlights the effects of the coupling schemes on stability, accuracy, and computational efficiency in aeroelastic wind turbine simulations using vortex methods. Through these case studies, the chapter provides insights into the role of coupling methodologies in both simplified and complex aeroelastic simulations. The results in the wind turbine case show that using alternative coupling techniques allows aeroelastic simulations with FVW methods to significantly reduce computational time, potentially making them viable alternatives to the BEM approach.

4.1 Numerical prototype: Pitchou/structure coupling

In chapter 3, the methodology behind the vortex method for computing aerodynamic forces was thoroughly presented, leading to its numerical implementation of the Pitchou code. In this section, Pitchou is used to model a simplified fluid/structure, two-degree-of-freedom, coupling problem. The goal is to detail the process of coupling a simplified finite element method with Pitchou to model a pitch-heave problem. For that matter, this section is organised as follows: first, the pitch-heave problem and its parameters will be described. Next, the development of the simplified finite element method will be outlined, followed by the numerical implementation and the time-marching algorithm that leads to the fluid-structure coupling. In the fourth part, validation results of this coupling approach will be presented. Finally,

various coupling schemes, including CSS and subcycling, will be analysed, providing preliminary results before transitioning to the next section, which covers a more realistic aeroelastic simulation of the IEA 15MW wind turbine.

4.2 Pitch-heave problem description

The aeroelastic behaviour of straight wings with infinite span can be studied through a two-dimensional airfoil section, as depicted in fig. 4.1. This model is characterised by two degrees of freedom, allowing the airfoil to undergo vertical displacement and rotational motion about its elastic center E. This representation is particularly useful for capturing the coupling of these two critical motions: pitch, denoted by the angle θ, and heave, represented by the vertical displacement h. These motions are interconnected by the aeroelastic properties of the wing, which are modelled using torsional and flexion springs. This approach provides a simplified description of the wing's elastic response under aerodynamic loads. The aeromechanical system under consideration can be described by a limited set of parameters that collectively determine its response to aerodynamic forces. The first of these parameters is the mass of the airfoil profile, denoted by m, which influences the inertial forces acting on the system. Secondly, the flow speed U_∞, representing the velocity of the airflow relative to the airfoil, along with the air density ρ_∞, which affects the magnitude of the aerodynamic forces. The aerodynamic centre A, located at the quarter chord (c) of the profile, provides the point where lift forces L are computed (to simplify, drag forces are neglected here). The elastic center E is the point associated with the springs' action, where resistance to rotational and vertical motions is described through torsional and flexion stiffnesses k_θ and k_h respectively. In this model, the airfoil is treated as a rigid body, meaning that it is assumed to undergo no elastic deformations. This simplification allows the focus to be on the solid-body motions of the airfoil, specifically its rotation about the elastic center and its vertical displacement.

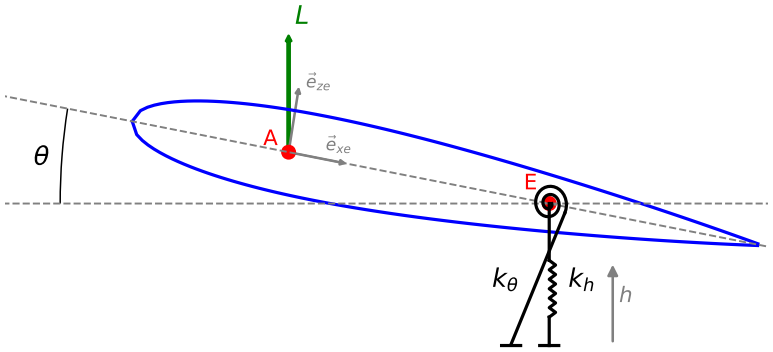


Figure 4.1: Schematic representation of the pitch-heave problem

4.3 Structural solving method: a simplified FEM

This section outlines the step-by-step development of a simplified finite element method, providing the reader with an understanding of how the structural quantities were derived in this work for their use in fluid-structure coupling problems using Pitchou. The first part introduces the application of Euler-Lagrange equations to the pitch-heave problem, along with the kinetic and potential energy relations and the principle of virtual work, which are used to derive the equations of motion for the problem. These equations are then discretised in time, demonstrating how the coupling with Pitchou is achieved and identifying the key quantities needed for transferring information between the fluid and structural problems. This forms the basis for the next section, which details the iterative process of the coupling algorithm.

4.3.1 Lagrangian and Euler-Lagrange equations

The Lagrangian function is used to describe a system's dynamics, defined as the difference between the system's total kinetic energy T and potential energy U:

$$\mathcal{L}(q, \dot{q}, t) = T(q, \dot{q}, t) - U(q, t), \tag{4.1}$$

where $\mathcal{L}$ depends on the system's generalised coordinates q_i, generalised velocities $\dot{q}_i$ and time t.

The Euler-Lagrange equations, derived from the Lagrangian function, are used to determine the equations of motion for a mechanical system. For a system with n degrees of freedom, this results in a set of n second-order differential equations:

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i, \tag{4.2}$$

where q_i represents the generalized coordinates, $\dot{q}_i$ the generalized velocities, and Q_i the generalized external forces.
In the case of the pitch-heave problem, the equations become

$$\frac{d}{dt}\left(\frac{\partial (T-U)}{\partial \dot{q}_i}\right) - \frac{\partial (T-U)}{\partial q_i} = Q_i, \tag{4.3}$$

with $q_i = [h, \theta]$ representing the system's degrees of freedom, where h is the heave displacement and θ is the pitch angle. T and U denote the kinetic and potential energy, respectively, while $Q_i = [Q_h, Q_\theta]$ represents the external aerodynamic forces acting on the structural system due to displacements in heave and pitch.

4.3.2 Potential and kinetic energy

To derive the equations of motion for the pitch-heave problem using the Euler-Lagrange framework, it is necessary to define the system's potential and kinetic energy as shown in eq. (4.3). The potential energy for this problem is given by:

$$U = \frac{1}{2}k_h h^2 + \frac{1}{2}k_\theta \theta^2, \tag{4.4}$$

where k_h and k_θ represent the spring stiffnesses for heave and pitch, as illustrated in fig. 4.1. The kinetic energy of the system is defined as:

$$T = \frac{1}{2}mV_A^2 + \frac{1}{2}I_A\dot{\theta}^2, \tag{4.5}$$

where V_A is the velocity at the aerodynamic center, and I_A is the moment of inertia about the same point. A relation is needed to express the velocity at the aerodynamic center in terms of the velocity at the elastic center, when they are considered distinct points and can be expressed through:

$$\vec{V_A} = \vec{V_E} + \vec{AE} \wedge \vec{\Omega}, \tag{4.6}$$

where $\vec{V_E} = \dot{h}\vec{e_z}$ represents the heave velocity at the elastic center, $\vec{AE} = \frac{1}{4}c\vec{e_{xe}}$ is the vector between the aerodynamic and elastic centers, and $\vec{\Omega} = \dot{\theta}\vec{e_y}$ denotes the rotational velocity. Using the relation between velocities at the aerodynamic and elastic centers, along with small-angle approximations (i.e., $\cos\theta \approx 1$ and $\sin\theta \approx \theta$), the kinetic energy simplifies to:

$$T = \frac{1}{2}m\left(\dot{h}^2 + \frac{1}{2}c\dot{h}\dot{\theta} + \frac{1}{16}c^2\dot{\theta}^2\right) + \frac{1}{2}I_A\dot{\theta}^2. \tag{4.7}$$

4.3.3 Virtual work principle

The virtual work principle allows for the incorporation of external forces, such as lift and moment, into the analysis by considering the work done by these forces during virtual displacements. In the context of the pitch-heave problem, the velocity at the aerodynamic center is given by:

$$\vec{V}_A = \left(\dot{h} + \frac{1}{4}c\dot{\theta}\right)\vec{e}_z. \tag{4.8}$$

Virtual displacements, representing small virtual changes in the system's position, can be expressed as follows:

$$\delta\vec{d}_A = \left(\delta h + \frac{1}{4}c\delta\theta\right)\vec{e}_z. \tag{4.9}$$

The virtual work due to lift for a displacement $\delta\vec{d}_A$ at point A and aerodynamic moment about the elastic centre can be expressed by:

$$\delta W = L\vec{e}_z\delta\vec{d}_A + M_A\delta\theta. \tag{4.10}$$

Substituting the expression for the virtual displacements into the virtual work equation yields:

$$\delta W = L\delta h + \left(M_A + \frac{1}{4}cL\right)\delta\theta. \tag{4.11}$$

The generalised external forces acting on the system can thus be identified as:

$$\begin{cases} Q_h = L \\ Q_\theta = M_A + \dfrac{1}{4}cL. \end{cases}$$

These expressions for Q_h and Q_θ provide the external forces acting on the system in terms of the aerodynamic lift and moment, which will be used to derive the equations of motion.

4.3.4 Equation of motion

For a system with n degrees of freedom, there are n corresponding Euler-Lagrange equations that describe its motion. In this case, with two degrees of freedom (h and θ), substituting the previously derived quantities into the Euler-Lagrange equations results in two equations of motion. These equations can be compactly represented in matrix form as:

$$\underbrace{\begin{bmatrix} m & \frac{1}{4}mc \\ \frac{1}{4}mc & \frac{1}{16}mc^2 + I_A \end{bmatrix}}_{\underline{\underline{M}}} \ddot{\boldsymbol{q}} + \underbrace{\begin{bmatrix} k_h & 0 \\ 0 & k_\theta \end{bmatrix}}_{\underline{\underline{K}}} \boldsymbol{q} = \underbrace{\begin{bmatrix} 1 & 0 \\ \frac{1}{4}c & 1 \end{bmatrix}}_{\underline{\underline{T}}} \begin{bmatrix} L \\ M_A \end{bmatrix}. \tag{4.12}$$

where $\boldsymbol{q} = [h, \theta]^T$ is the vector of generalized coordinates, $\ddot{\boldsymbol{q}}$ is the vector of accelerations for heave and pitch, the mass matrix $\underline{\underline{M}}$ captures the system's inertial

properties, the stiffness matrix $\underline{\underline{K}}$ includes the stiffness coefficients k_h and k_θ, and the transformation matrix $\underline{\underline{T}}$ relates the lift aerodynamic forces L and moment M_A from the aerodynamic to the structural coordinates.

4.3.4.1 Time discretisation

To solve the equations of motion numerically, it is necessary to discretise them in time. The above equation of motion can be rewritten for a specific time step $n+1$ as:

$$\underline{\underline{M}}\ddot{\boldsymbol{q}}_{n+1} + \underline{\underline{K}}\boldsymbol{q}_{n+1} = \underline{\underline{T}}\boldsymbol{Q}_{n+1}. \tag{4.13}$$

Note that this is one type of discretisation, where all components are expressed at time step $n+1$ as the arodynamics is the first solved problem allowing the structural problem to be solved implicitly. Once the structural equations of motion are established, and the aerodynamic forces are determined by the fluid solver, a time integration scheme is required to compute the displacements and velocities of the structure at time step $n+1$. In this case, the chosen time integration scheme for is the Newmark method [99], defined as:

$$\begin{cases} \dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}_n + (1-\gamma)\,\Delta t\ddot{\boldsymbol{q}}_n + \gamma\Delta t\ddot{\boldsymbol{q}}_{n+1} \\ \boldsymbol{q}_{n+1} = \boldsymbol{q}_n + \Delta t\dot{\boldsymbol{q}}_n + \Delta t^2\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{q}}_n + \Delta t^2\beta\ddot{\boldsymbol{q}}_{n+1}, \end{cases} \tag{4.14}$$

where $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$ for an unconditionally stable time integration scheme.
By substituting the Newmark expression for displacement at time step $n+1$ into equation (4.13), the following explicit expression for the acceleration at time step $n+1$ is obtained as a function of the system's state at time step n:

$$\ddot{\boldsymbol{q}}_{n+1} = \left(\underline{\underline{M}} + \beta\Delta t^2\underline{\underline{K}}\right)^{-1}\left[\underline{\underline{T}}\boldsymbol{Q}_{n+1} - \underline{\underline{K}}\left(\boldsymbol{q}_n + \Delta t\dot{\boldsymbol{q}}_n + \Delta t^2\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{q}}_n\right)\right]. \tag{4.15}$$

This expression can then be substituted back into eq. (4.14) to calculate both the displacements and velocities at time step $n+1$ induced from aerodynamic forces from the same time step.

4.3.5 Fluid/structure coupling algorithm using Pitchou

Algorithm 2 outlines the fluid-structure coupling process, presented in a subcycling form where the structural problem is subcycled. This approach can be adjusted into a classical staggered scheme (CSS) by setting the number of sub-cycles to zero ($subIt = 0$), or it can be adapted to subcycle the fluid problem instead. The process begins at time step $n+1$, where aerodynamic forces are computed using the FVW solver Pitchou, using updated node positions provided by the structure at time step n. These forces are then transferred from the aerodynamic center nodes to the elastic center nodes if they are distinct points. A sub-iteration loop follows, where the structural problem is solved using Newmark's scheme for each sub-iteration k. If subcycling is not applied, Newmark's scheme is only used once. After exiting the loop,

prediction terms are applied to the computed structural displacements to anticipate their behavior at the next time step $n+2$ to best align with the aerodynamic solver's time step. Finally, the structural displacements are transferred back from the elastic center nodes to the aerodynamic center nodes for the calculation of aerodynamic forces at time step $n+2$.

Algorithm 2: Coupling algorithm for pitch-heave problem modelling

Data: Wing geometry and kinematic
Result: Wing loads, rigid displacements and wake geometry
Compute aerodynamic forces at aerodynamic center nodes following 1
Transfer aerodynamic forces from aerodynamic to elastic center nodes
while $k < subIt$ **do**
    Compute accelerations $\ddot{q}_{n+1,k}$ using eq. (4.15)
    Apply Newmark's scheme to compute $\dot{q}_{n+1,k}$ and $q_{n+1,k}$
end
Apply prediction terms from eqs. (2.18) and (2.19)
Transfer structural displacements from elastic center nodes to aerodynamic center nodes

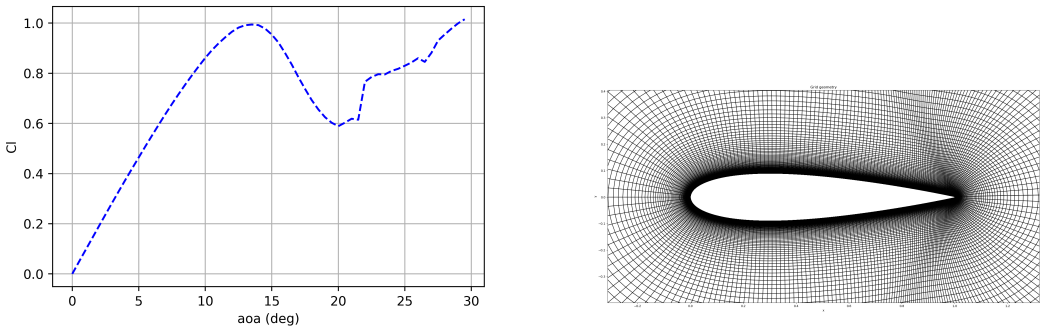
4.3.6 Coupling validation

The validation results for the fluid-structure coupling using the Pitchou solver are presented next, covering two test cases. In the first case, the structure was subjected to a forced sinusoidal movement. This allowed for a code-to-code comparison with CFD results from the SU2 solver, as well as with the analytical solution given by Theodorsen's method. The second test case examined a freely oscillating structure subject to an initial disturbance, where various oscillation frequencies were analysed to evaluate the stability of the coupled system. The insights from this second case help to identify the optimal parameters for the next part, ensuring that the analysis of different coupling schemes is conducted on a stable coupled system.

4.3.6.1 Forced pitching motion

The test case involving a structure subjected to forced displacements is based on the approach described in [135], which examines a pitching airfoil at low angles of attack in a constant free stream. The selected airfoil profile is the NACA 0018, with imposed pitching displacements defined by $\alpha(t) = 2° + 2° sin(\omega t)$ where $\omega = 2\pi f$ represents the angular frequency. Simulations were conducted for two different reduced frequencies, $k = \frac{\omega c}{u}$, specifically $k_1 = 0.08$ and $k_2 = 0.263$, with an inflow velocity of $u = 4.5\,\mathrm{m/s}$. A code-to-code comparison of the results was performed using the SU2 solver [136], alongside an analytical comparison with a solver implementing Theodorsen's method. The SU2 solver was first used as a steady-state CFD tool to compute the aerodynamic polars, which were then used in Pitchou for lift coefficient identification. Subsequently, SU2 was employed as an unsteady CFD tool to model the oscillating pitching motion of the NACA 0018 airfoil. Figure 4.2 shows the simulation mesh, featuring an O-shaped domain with a radius of 500 chord lengths and 300 grid cells in the direction perpendicular to the flow. The lift coefficient polar

obtained from the angle-of-attack sweep during the steady-state computations with SU2 is shown in Figure 4.2.



(a) (b)

Figure 4.2: (a) aerodynamic lift polar obtained through SU2 simulation and (b) airfoil mesh used in SU2

Figure 4.3 compares the hysteresis loops obtained using Theodorsen's theory, SU2, and Pitchou. These loops start at 2°, increase to a maximum angle of attack of 4°, and then return to the starting value. For a low reduced frequency $k_1 = 0.08$, the lift coefficient results are similar across all three methods and align well with the findings of [135]. At this low reduced frequency, the slope of the CFD curve is close to 2π, indicating that quasi-steady effects dominate over unsteady lift effects. The vortex method also agrees closely with the CFD results, as both exhibit comparable slopes. In contrast, Theodorsen's method predicts a slight increase in the lift curve. Despite this difference, the amplitude of the hysteresis loops is similar across all three approaches. At a higher reduced frequency $k = 0.263$, both the CFD and vortex methods predict a horizontal tilt in the slope compared to the lower reduced frequency, consistent with experimental results from [135]. The trend of increased hysteresis with rising k is observed in both the vortex method and Theodorsen's theory, but appears to be absent or negligible in the CFD results.

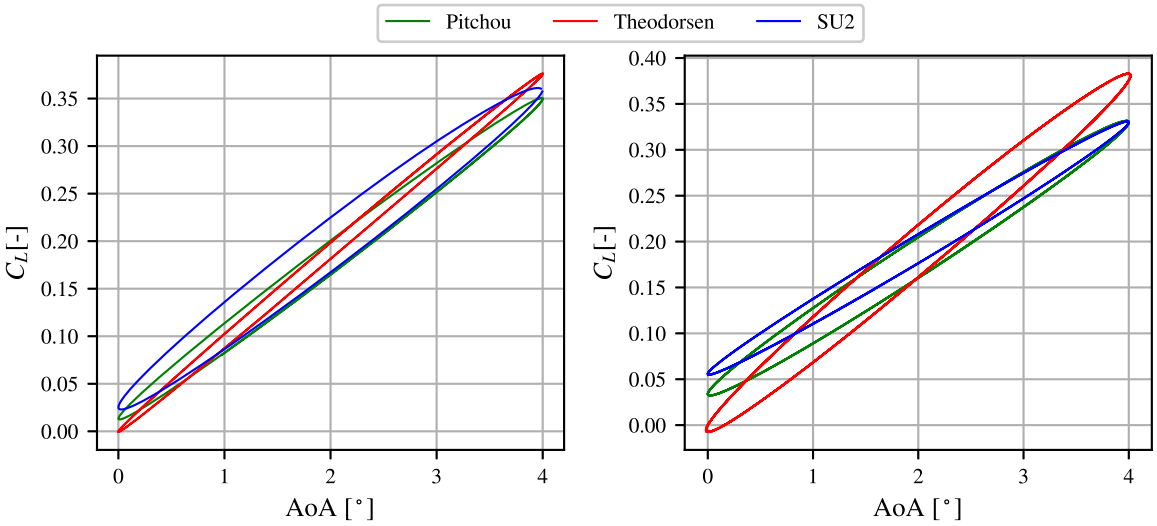


Figure 4.3: Aerodynamic lift coefficients using different solvers for imposed structural motion $\alpha(t) = 2° + 2°sin\,(\omega t)$ where (left) $k_1 = 0.08$ and (right) $k_2 = 0.263$

4.3.6.2 Free movement case

In this section, the structure is no longer subjected to forced displacements. Instead, it experiences an initial disturbance, and the subsequent coupling behaviour is studied. This shifts the focus to the aeroelastic stability of the two-degree-of-freedom airfoil system. In such scenarios, not all aeroelastic responses will result in a stable system. Flutter is a dynamic aeroelastic phenomenon in which the interaction between aerodynamic, elastic, and inertia forces can cause energy to be extracted from the airflow, potentially leading to instability [137]. The critical speed is defined as the minimum inflow velocity above which the structure transitions from stable, simple harmonic oscillations to divergent behaviour. Below this threshold, the system remains stable. Studies such as [137] and [138] provide the flutter limits using Theodorsen's method and compare computational results to assess the consistency of the critical speed given a specific set of parameters. These parameters include ω_h and ω_α, the uncoupled plunge and pitch radian frequencies, U_∞ the airspeed, x_α, the distance from the elastic center of the airfoil to its center of mass, and r_α, the radius of gyration. The stability boundary is plotted as $V = U_\infty / b\omega_\alpha$ (where b is the semi-chord) against ω_h/ω_α. In the first computation, with $\omega_\alpha = 1$ and $\omega_h = 1$, the inflow velocity is set to $U_\infty = 4\,\mathrm{m/s}$, $x_\alpha = 0$ and $r_\alpha^2 = 0.25$. The graph below shows the evolution of the pitching angle of the airfoil and the associated aerodynamic forces over time. This case clearly demonstrates stable aeroelastic conditions: the pitching displacements exhibit damped harmonic motion, gradually decreasing over time. This result aligns with the findings of [138], which predict unconditional stability for these parameter values. Figure 4.5 illustrates the same test case with an inflow velocity of $U_\infty = 6\,\mathrm{m/s}$, where the aerodynamic forces maintain a decreasing harmonic behaviour despite the increased velocity, consistent with the unconditionally stable region.

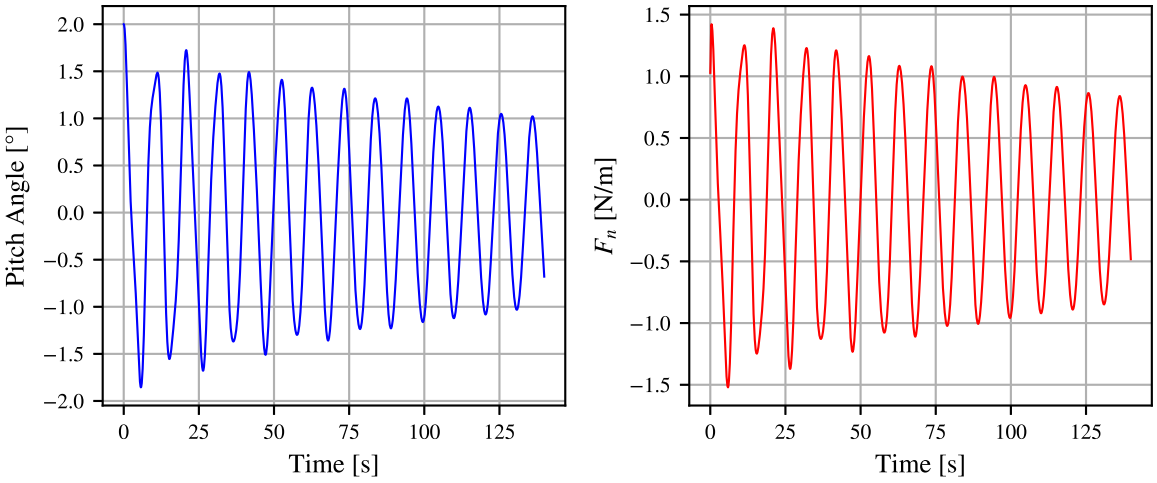


Figure 4.4: (left) Pitch displacements and (right) normal aerodynamic forces evolution for unconditionally stable case $U_\infty = 4\,\mathrm{m/s}$

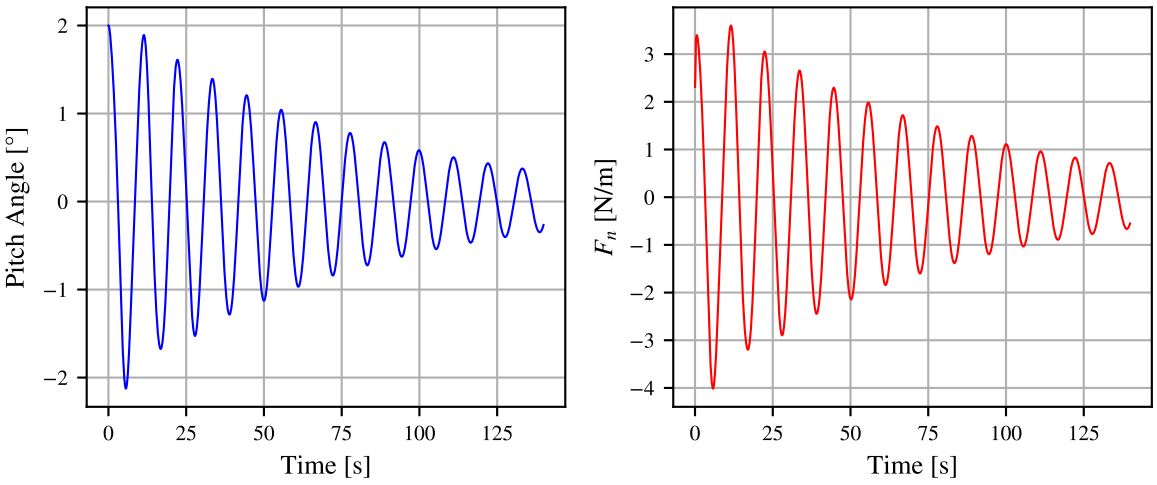


Figure 4.5: Time evolution of (left) Pitch angle and (right) normal aerodynamic forces for unconditionally stable case $U_\infty = 6\,\mathrm{m/s}$

The second case studied involves $\omega_\alpha = 1.$ and $\omega_h = 0.4$. According to bibliographic references, this configuration is expected to have a defined flutter boundary. A critical speed, U_{crit}, was identified around $3\,\mathrm{m/s}$, which is slightly lower than the value reported in [138]. Above this critical speed, the system becomes unstable, with the amplitudes of harmonic oscillations increasing and eventually becoming divergent. Figure 4.6 illustrates this scenario for the same inflow velocity as in fig. 4.4 of $U_\infty = 4\,\mathrm{m/s}$. In the upcoming sections, the parameters from the second stable case in fig. 4.5 will be used for the analysis of different coupling schemes on this model problem.

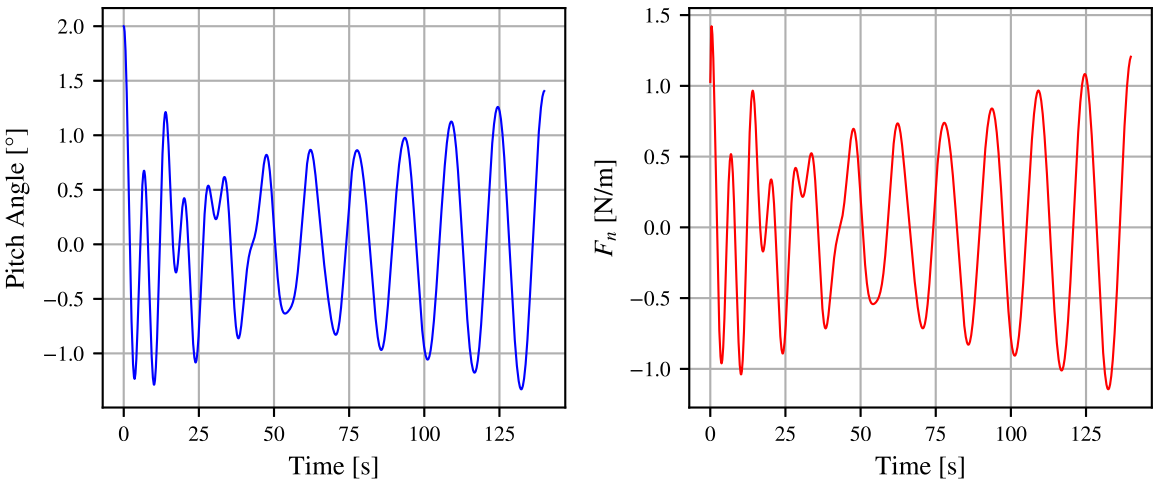


Figure 4.6: Time evolution of (left) Pitch angle and (right) normal aerodynamic forces for unconditionally stable case $U_\infty = 4\,\mathrm{m/s}$ $\omega_h = 0.4$

4.3.7 Coupling schemes analysis on pitch-heave problem

This section analyses the effects of different coupling schemes on the pitch-heave problem. First, the results for the CSS coupling scheme will be presented, focusing on time-step sensitivity analysis and its impact on computational time, pitch motion, and aerodynamic force calculations. This analysis will also include an assessment of the prediction terms introduced in section 2.2.2.1. In the second part, the focus shifts to evaluating the impact of a subcycling scheme, where the structural dynamics are subcycled relative to the aerodynamics.

4.3.7.1 CSS

In this part, the test case uses the same parameters as those from fig. 4.5. A time-step sensitivity analysis was conducted to examine the differences in normal aerodynamic forces and pitch displacements when varying the time step from $\Delta t = 0.5\,\mathrm{s}$ to $\Delta t = 0.025\,\mathrm{s}$ using a CSS coupling scheme. The results are displayed in fig. 4.7. Both aerodynamic forces and pitch displacements exhibit similar behaviour: larger time steps lead to underestimated forces and pitch angles, while the discrepancies in amplitude decrease as the time step becomes smaller. The total computation time for each discretisation is provided in table 4.1. It shows a significant increase in total simulation time as the time step decreases. This increase is mainly due to the FVW solver, whose computational time scales quadratically with the number of filaments emitted in the wake. As the number of filaments increases from 120 for a time step of $\Delta t = 0.5\,\mathrm{s}$ to 2400 for the smallest time step of $\Delta t = 0.025\,\mathrm{s}$, the total simulation time increases substantially.

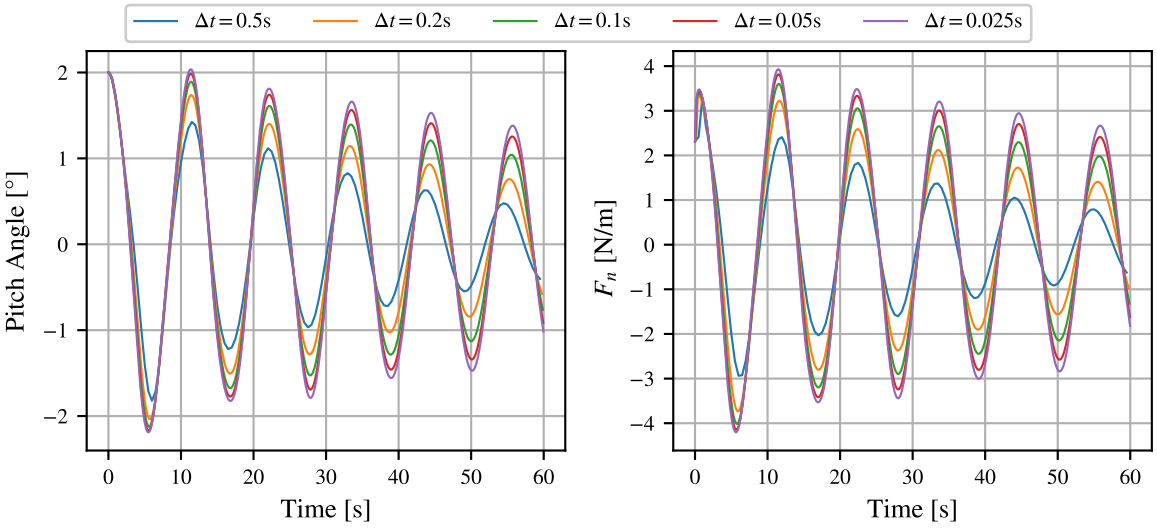


Figure 4.7: Time step sensitivity on (left) pitch angle and (right) normal aerodynamic forces evolution for stable case $U_\infty = 3\,\mathrm{m/s}$

Time step (s)	0.5	0.2	0.1	0.05	0.025
Total comp. time	8s	39s	3min31s	23min18s	2h44min

Table 4.1: Total computation time for a 60 second simulation using different time steps.

Figure 4.8 presents the Root Mean Square Error (RMSE) values of aerodynamic force computations across different time steps, comparing the CSS scheme without prediction terms to CSS schemes using prediction 1 and prediction 2, as defined in eq. (2.18) and eq. (2.19). The plot shows that incorporating either prediction 1 or 2 reduces the RMSE values compared to the original CSS scheme without predictions, achieving better accuracy for larger time steps without adding extra computational time.

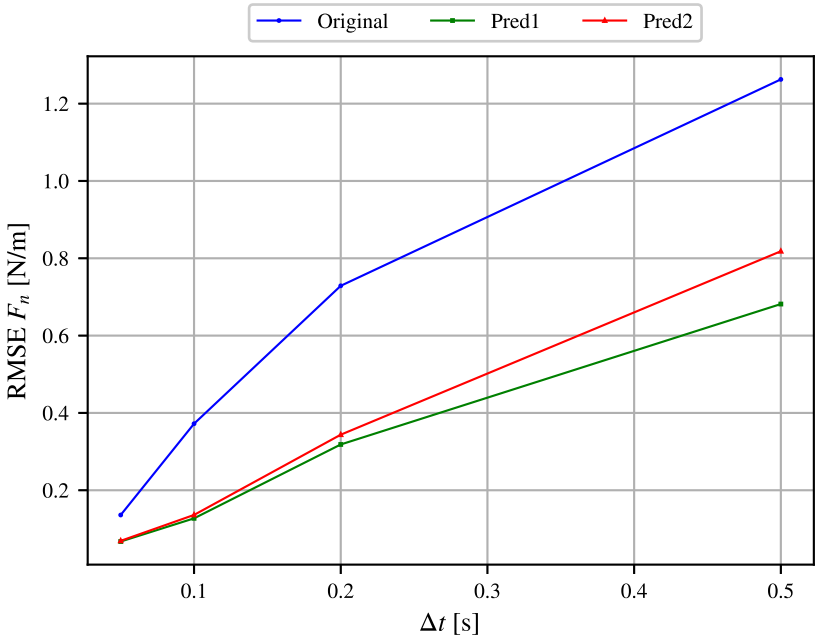


Figure 4.8: Comparison of RMSE values for aerodynamic forces across different time steps for CSS with and without prediction terms.

4.3.7.2 Subcycling

Subcycling iterations were applied to the structural part of the problem, allowing the use of a coarser time step for the fluid resolution $\Delta t_f = 0.2\,\text{s}$ while introducing 1 to 7 sub-iterations for the structural computations. This approach achieved structural time steps ranging from $\Delta t_s = 0.1\,\text{s}$ to $\Delta t_s = 0.025\,\text{s}$. Results comparing the CSS scheme with time steps $\Delta t = 0.2\,\text{s}$ and $\Delta t = 0.025\,\text{s}$ to the subcycling schemes are shown in fig. 4.9. The RMSE values are also plotted, highlighting the differences in error between the CSS scheme and subcycled schemes with varying numbers of subcycles, using the CSS scheme with the finest time step as the reference solution. The results indicate that sub-iterations on the structural part improve the accuracy of the aerodynamic force amplitudes compared to the standard CSS case. This is further shown by the RMSE results, where the error relative to the reference solution is minimised when the maximum number of sub-iterations is used.

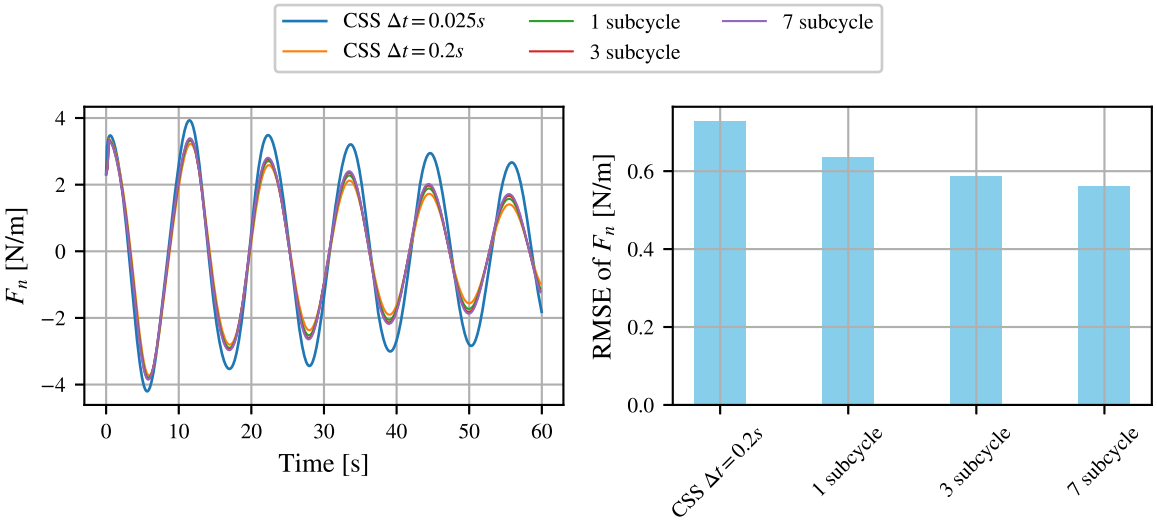


Figure 4.9: Effects of structural subcycles on (left) normal sectional aerodynamic forces and its (right) RMSE

4.4 IEA 15MW aero-servo-elastic case

The focus now shifts to the large wind turbine framework, specifically the aeroelastic modelling of the IEA 15 MW turbine using various partitioned coupling schemes, for which results were described in [2]. For this purpose, the aero-servo-elastic modelling tool DeepLines Wind™ is employed, using CASTOR as the aerodynamic solver, coupled with the Finite Element code Deeplines [7], all managed by the ROSCO controller [70]. To analyse the effects of different coupling schemes on this more realistic test case, the next sections will proceed as follows: first, the characteristics of the coupling schemes are discussed, including an analysis of the coupling types to draw parallels with section 2.1.2.3. This is followed by an examination of reference aeroelastic solutions, focusing on the effects of varying wake lengths, with a discussion on the implications of different time step sizes for both the structural and fluid solvers. These analyses provide the basis for defining the conditions under which, in the final section, a stability and accuracy analysis is conducted. This analysis compares the performance of the CSS and subcycling schemes using forecasting techniques from section 2.4.3, for a wind turbine operating under both steady and turbulent wind conditions.

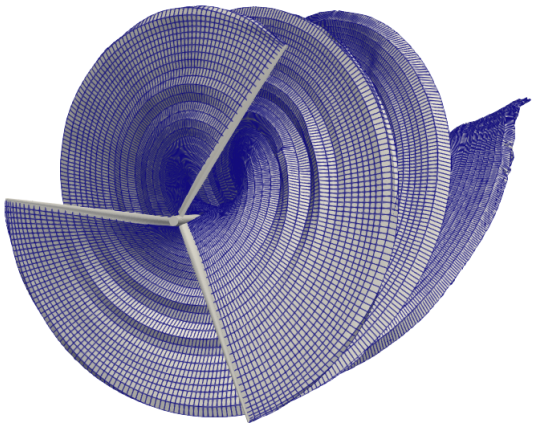


Figure 4.10: Wake generation on the IEA 15MW wind turbine

4.4.1 Aerodynamic effects on coupled problem

The aim of this first part is to assess the impact of including aerodynamic effects in comparison to a purely elastic simulation where no coupling with an aerodynamic model is considered. To achieve this, two scenarios were simulated using the IEA 15MW wind turbine in a non-rotating state. In both scenarios, a white noise force of 3 N was applied along the blade span over a one-hour simulation period. The first scenario incorporates aerodynamic effects with a constant wind speed of $U_\infty = 10.6\,\mathrm{m/s}$, allowing for the analysis of fluid/structure coupling. The second scenario represents a purely structural case, without any aerodynamic interactions. Figure 4.11 shows the frequency response for both scenarios, derived from a Fourier Transform (FT) of the blade's tip displacements (u and v). In the structural-only case, distinct natural frequencies of the blade modes are evident around 0.5 Hz, corresponding to the first flapwise tilt mode, the first collective flap mode, and the first collective edgewise mode. Additionally, the second collective edgewise mode appears around 1.5 Hz. In the aeroelastic case, these natural frequencies remain visible but with reduced amplitude. The comparison between the two scenarios highlights differences mainly in the amplitude of the Fourier Transform rather than in the frequencies themselves. This indicates that aerodynamic effects primarily introduce a damping effect on the structural response while having minimal impact on the natural frequencies. This observation aligns with the damping coupling framework defined in section 2.1.2.3.

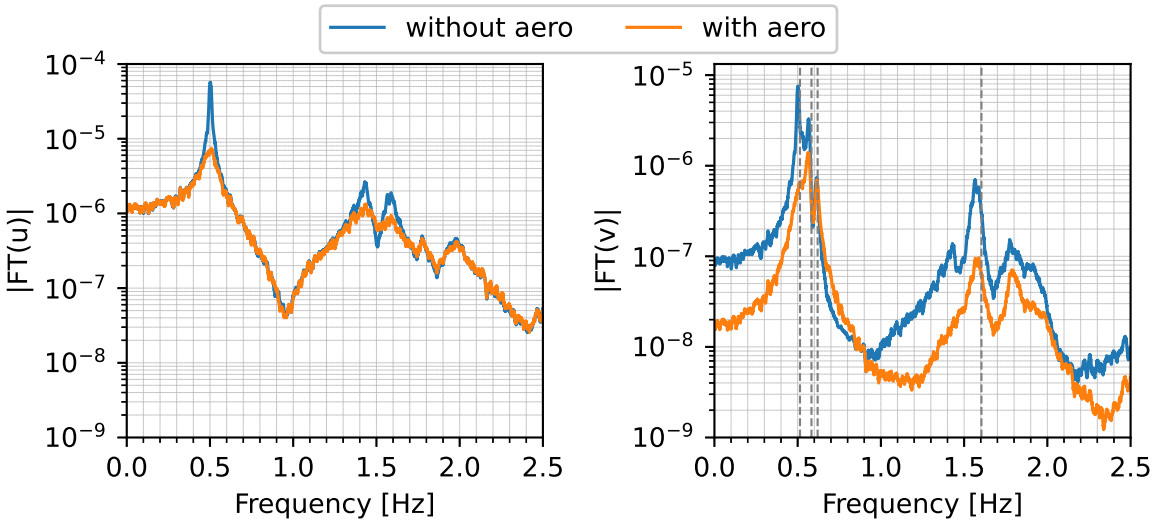


Figure 4.11: Frequency response to white noise force applied on a blade.

4.4.2 Discussion on parameters for aeroelastic reference solution

In the previous chapter, several parameters, including azimuthal discretisation and the core radius of different desingularisation methods, were analyzed to understand their impact on aerodynamic simulations using a Free Vortex Wake (FVW) solver. The results demonstrated that the time variation of aerodynamic forces is highly sensitive to these parameters, making their proper tuning crucial for aeroelastic simulations where aerodynamic forces are transferred to the structural solver to model the coupled behaviour. For the following simulations, an Offset model with a core radius of $r_c = 0.1$ is used. Choosing an appropriate azimuthal discretisation for aeroelastic simulations using a CSS coupling technique is more challenging, as it depends on both the aerodynamic and structural time discretisations. As discussed earlier, the CSS scheme enforces a common time step for the integration of both physical problems, which necessitates selecting the most restrictive characteristic time step. This often results in an over-resolution of the less restrictive physics. In this study, the first blade modes of the IEA 15MW turbine exhibit eigenfrequencies around 0.5 Hz, as shown in fig. 4.11. This corresponds to a structural time step of approximately $\Delta t_s = 0.01\,\mathrm{s}$, which would require an azimuthal discretisation of less than 1° per time step for a wind turbine operating at its nominal speed. This level of discretisation is significantly finer than the recommended 5° to 10° per time step suggested in [40], leading to increased computational costs and potentially affecting the accuracy of the solution. Nevertheless, the CSS scheme with this finer discretisation is used for the reference solution, owing to the absence of a well-established alternative and its established use in previous studies [1]. Furthermore, the accuracy of the aerodynamic force calculations using the free vortex wake method is influenced by the wake length, which is determined by the number of vortex filaments retained behind the rotor [40]. A convergence analysis of the aerodynamic forces as a function of wake length was conducted using a subcycling algorithm with

7 sub-iterations. Given the high computational expense of the CSS approach and the comparable results achieved with subcycling (as discussed in the next section), the subcycling method serves as a reliable estimate of the converged wake length, but at a lower computational cost. The results of this analysis are presented in fig. 4.12, showing that the sectional normal forces converge when the wake retains vortex filaments equivalent to 7 to 15 rotor rotations. For all subsequent simulations in this study, a fixed wake length corresponding to 13.5 rotor rotations is adopted.

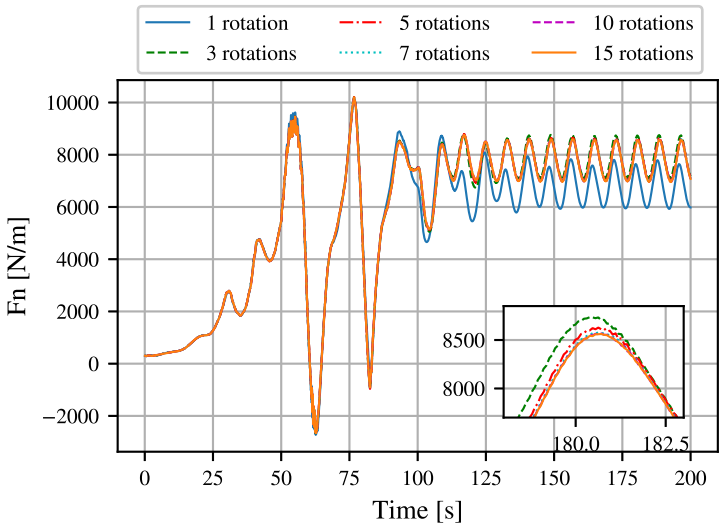


Figure 4.12: Convergence of sectional aerodynamic forces at radius location $r = 70\,\mathrm{m}$ function of different wake lengths.

4.4.3 Constant wind case

The IEA 15MW wind turbine is subjected to a constant wind with a velocity of $U_\infty = 10.6\,\mathrm{m/s}$, resulting in a tip speed ratio of $\lambda = 8.5$. In this first case, a 200 s aeroelastic simulation was performed, coupling DeepLines with CASTOR, and using various coupling algorithms. As discussed above, the reference case employs a CSS scheme, where the coupling time step is set to $\Delta t_c = 0.01\,\mathrm{s}$. For comparison, a subcycling algorithm was implemented with an increasing number of regression iterations using constant forecasting. The sectional aerodynamic forces obtained from both the CSS and subcycling methods are shown in fig. 4.13. As the number of sub-iterations in the coupling algorithm increases, a slight difference in amplitude between the aerodynamic forces of the subcycling methods and the reference case can be observed. Additionally, a small phase shift is introduced with the sub-iterations, potentially due to the estimation of induced displacements based on the constant forecasted aerodynamic forces received by the structural solver. This behaviour is observed up to a subcycled structure with 9 iterations, allowing the aerodynamic integration time step to be increased to $\Delta t_f = 1\,\mathrm{s}$, beyond which the coupling algorithm becomes unstable. In fig. 4.14, the results for subcycling with more than 9 sub-iterations are presented. While the computation converges with 10 sub-iterations, the aerodynamic forces are poorly resolved. With 12 sub-iterations, however, the computation

diverges completely, halting around 60 seconds during the turbine's startup phase. Table 4.2 presents the total computational times for the test case using the various coupling algorithms. By introducing subcycles to the coupling scheme and consequently enabling a slower time step for the fluid problem, the total computational time is significantly reduced. The use of a 9-iteration subcycled scheme results in a computational time more than 400 times faster than the CSS method. This efficiency gain is attributed to the reduced frequency of wake filament emissions in the vortex method and the direct use of a meta-model for aerodynamic data communication.

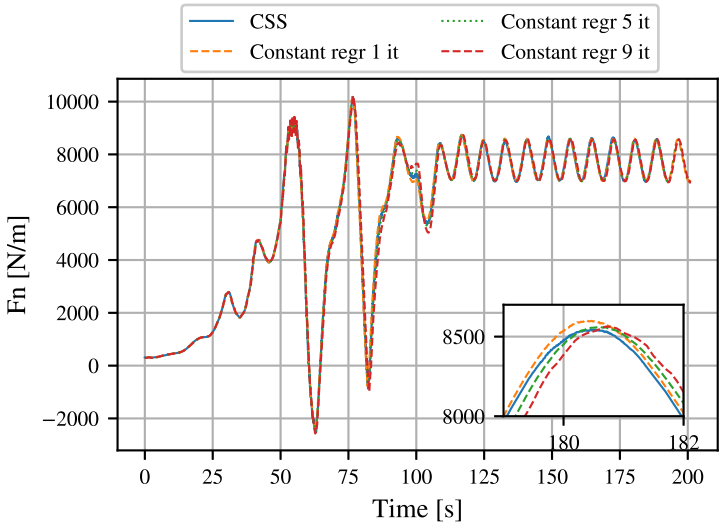


Figure 4.13: Sectional aerodynamic forces comparison for CSS method and subcycling scheme using constant regression at radius location $r = 70\,\mathrm{m}$.

Method	CSS	1 sub-it	2 sub-it	4 sub-it	7 sub-it	9 sub-it
Total comp. time	10d16h	35h06min	11h24min	2h55min	57min	36min

Table 4.2: Total computational times for CSS and subcycled schemes.

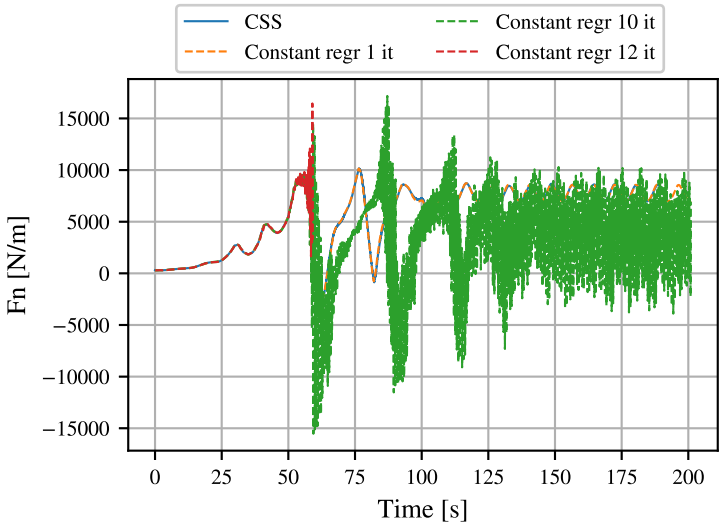


Figure 4.14: Sectional aerodynamic forces comparison for CSS method and unstable subcycling scheme using constant regression at radius location $r = 70\,\mathrm{m}$.

The results of the subcycling scheme with predictions using linear and second-order polynomial regressions are presented in fig. 4.15. Both forecasting techniques exhibit similar trends to those observed with constant regression: adding subcycles leads to variations in the amplitude of aerodynamic forces and introduces a slight phase lag compared to the CSS approach. In this analysis, the maximum number of stable subcycles for the linear and polynomial forecasting methods is 4 and 2, respectively, beyond which the subcycling algorithm becomes unstable. Tables 4.3 and 4.4 summarise the relative errors in amplitude and mean height compared to the CSS solution for the different subcycled algorithms. The results indicate that all subcycling schemes tend to slightly underestimate both the amplitude and the mean height of the normal sectional aerodynamic forces when compared to CSS. Additionally, for a given number of subcycles, the relative error in mean height is smallest when using the constant regression scheme, whereas polynomial forecasting yields the highest relative error.

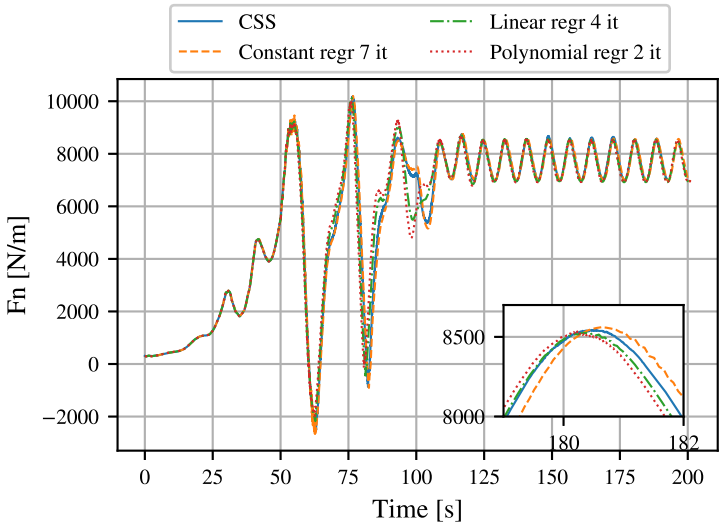


Figure 4.15: Sectional aerodynamic forces comparison for CSS method and subcycling scheme using linear and polynomial forecasting at radius location $r = 70\,\mathrm{m}$.

Sub-it number	1	2	4	7	9
Constant regr err (%)	3.8	3.38	3.2	2.26	2.41
Linear regr err (%)	3.0	3.48	3.14	-	-
Polynomial regr err (%)	1.85	2.17	-	-	-

Table 4.3: Relative error of amplitude between times 142s and 152s compared to CSS of sub-iterations methods with constant, linear and polynomial regression.

Sub-it number	1	2	4	7	9
Constant regr err (%)	0.78	0.77	0.75	0.79	0.72
Linear regr err (%)	1.09	1.33	1.30	-	-
Polynomial regr err (%)	1.72	1.55	-	-	-

Table 4.4: Relative error of mean height between times 142s and 152s compared to CSS of sub-iterations methods with constant, linear and polynomial regression.

4.5 Turbulent wind case

In a second scenario, the IEA 15MW wind turbine was subjected to turbulent wind conditions generated using the TurbSim simulator [139]. The mean wind speed at hub height was set to 8 m/s, with a turbulence intensity of 8%. Under these conditions, the subcycling scheme with constant, linear, and polynomial regressions performed comparably well to the constant wind case. The results for each method are shown below, where it can be seen that the maximum number of sub-iterations remains the same as in the previous case, except for the constant regression, where an additional sub-iteration could be applied. From table 4.5, it can be seen that the relative errors in amplitude remain minimal, with a maximum error around 6% for the constant regression case at the maximum number of sub-iterations, and lower errors for all other cases. This further demonstrates that this method is effective even under turbulent wind conditions. Owing to the already satisfactory performance of the constant, linear, and polynomial regression methods in the subcycling algorithm, the ARIMA model, which was originally considered, did not yield any additional benefits. Initial attempts with the ARIMA model have so far been computationally expensive and have not produced satisfactory results under these conditions. Given its higher complexity and lack of improvement over the other methods, it was not pursued further at this stage.

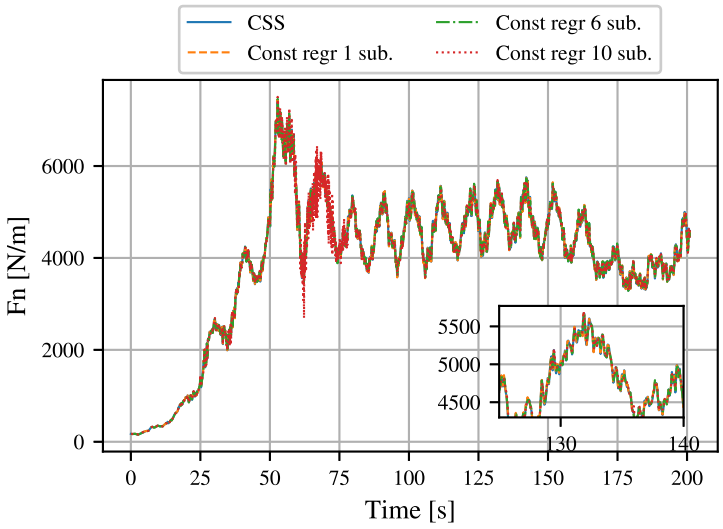


Figure 4.16: Sectional aerodynamic forces comparison for CSS method and subcycling scheme using constant regression at radius location $r = 70\,\mathrm{m}$ in turbulent wind case.

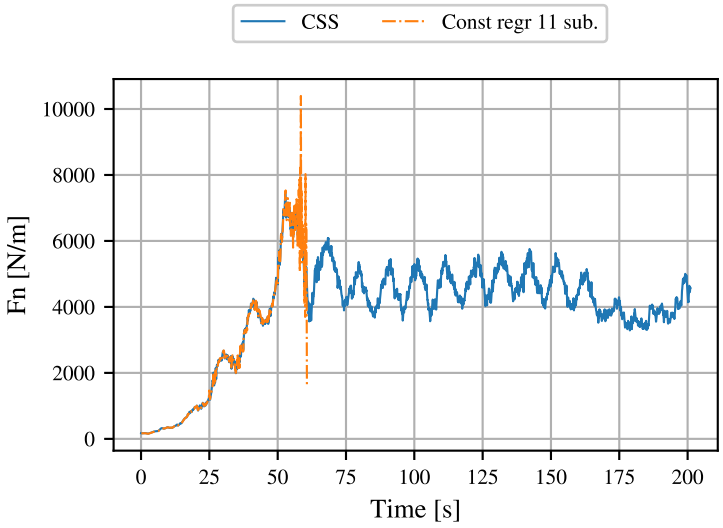


Figure 4.17: Sectional aerodynamic forces comparison for CSS method and unstable subcycling scheme using constant regression at radius location $r = 70\,\mathrm{m}$ in turbulent wind case.

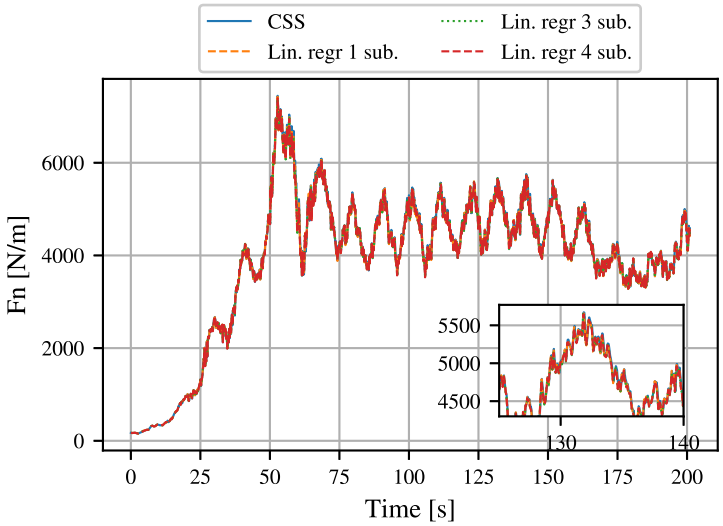


Figure 4.18: Sectional aerodynamic forces comparison for CSS method and subcycling scheme using linear regression at radius location $r = 70\,\mathrm{m}$ in turbulent wind case.

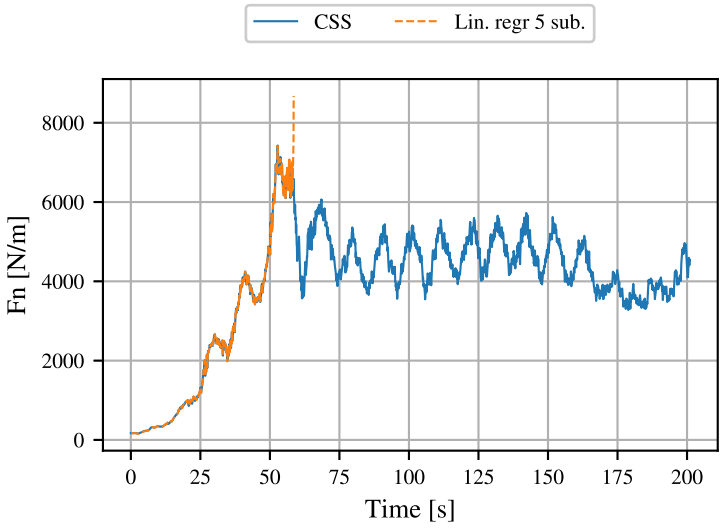


Figure 4.19: Sectional aerodynamic forces comparison for CSS method and unstable subcycling scheme using linear regression at radius location $r = 70\,\mathrm{m}$ in turbulent wind case.

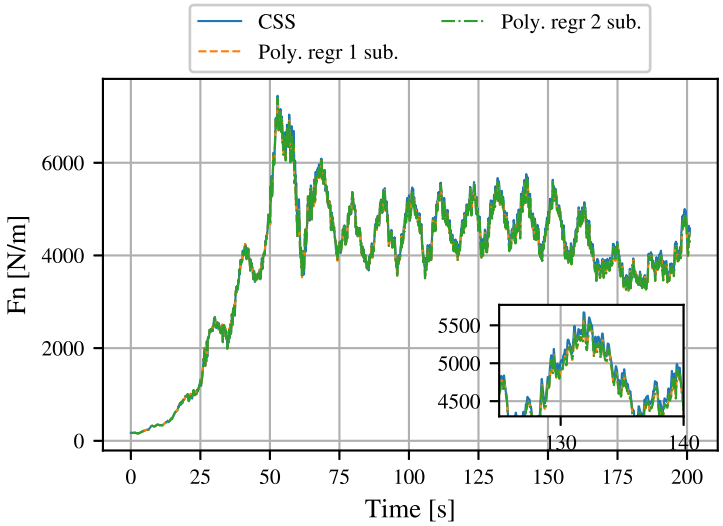


Figure 4.20: Sectional aerodynamic forces comparison for CSS method and subcycling scheme using polynomial regression at radius location $r = 70\,\mathrm{m}$ in turbulent wind case.

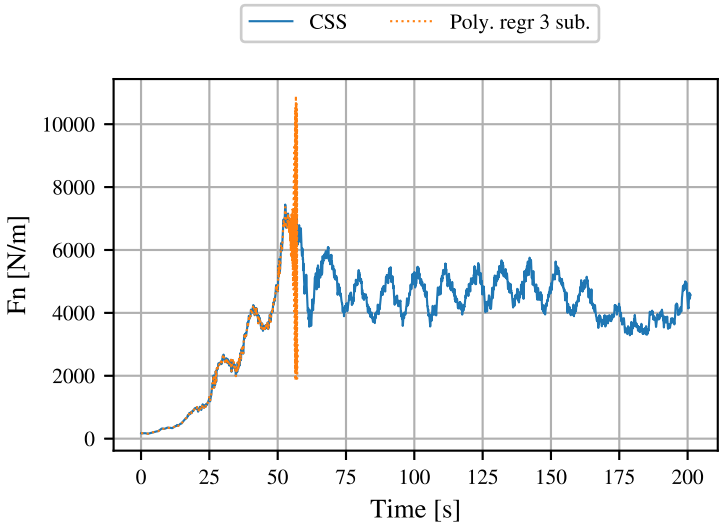


Figure 4.21: Sectional aerodynamic forces comparison for CSS method and unstable subcycling scheme using polynomial regression at radius location $r = 70\,\mathrm{m}$ in turbulent wind case.

Sub-it number	1	2	4	6	10
Constant regr err (%)	1.01	0.16	1.23	2.09	6.75
Linear regr err (%)	0.11	0.63	0.52	-	-
Polynomial regr err (%)	1.2	1.2	-	-	-

Table 4.5: Relative error of amplitude between times 142s and 152s compared to CSS of sub-iterations methods with constant, linear and polynomial regression in turbulent wind case.

4.5.1 Outcomes

In chapter 2, the subcycling scheme using constant and linear regression methods was applied to a simple one-degree-of-freedom linear system involving two coupled oscillators. Although this provided useful insights, it is important to note that this case represents a much simpler system than the full wind turbine model in the present chapter, yet some parallels can still be drawn. In the oscillator case, where coupling was defined by damping effects, the subcycling scheme allowed for a larger critical coupling time step compared to the critical time step in the CSS scheme. Similarly, in the wind turbine case, aerodynamic effects introduce a damping-like influence on the structure, drawing a parallel with the damping coupling observed in the oscillator scenario. The introduction of subcycles on the structural side in the wind turbine case allows for a larger coupling time step in the subcycling scheme, enabling the aerodynamic solver to be called at a slower rate, which could not initially be achieved with the CSS scheme. Aerodynamic forces are well computed up to a certain number of sub-iterations, after which numerical instability can occur, causing the aeroelastic computation to diverge. This may be due to the meta-model forecast,

which provides an approximation of the aerodynamic forces at each sub-iteration to the structural solver, deviating too much from the actual solution provided by the aerodynamic solver at each coupling time step, leading to discrepancies in the aerodynamic forces transmitted to the structural solver. This was also observed in the coupled oscillator case, where, at a certain critical coupling time step using the subcycling scheme, a numerical stability limit was reached. The subcycling algorithm in the wind turbine framework also leads to substantial computational improvements, with a simulation speed up of more than 400 times compared to the original CSS scheme, going from more than ten days of simulation to only thirty six minutes. Such computational gains are significant and could be further enhanced if combined with other methods, such as wake coarsening, described in chapter 3. The ARIMA model, although initially considered, has not yet provided satisfactory results. Its computational expense and lack of improvement over the simpler methods suggest that further evaluation is required before it can be effectively implemented in this context.

Chapter 5

Conclusions and perspectives

The increasing deployment of wind turbines, both onshore and offshore, plays a crucial role in the transition to renewable energy and the mitigation of climate change. However, the evolution in turbine design presents significant challenges, particularly in terms of aeroelastic modelling and the computational resources needed to simulate these interactions with sufficient accuracy. On the aerodynamic side, various modelling methods can be employed, with the Blade Element Momentum method being one of the most practical for design purposes due to its low computational cost. Despite this advantage, BEM is a low-fidelity method and requires a set of correction techniques to account for unsteady effects, which must be captured more accurately, especially with the growing importance of future floating wind turbines. In this work, the lifting line Free Vortex Wake (FVW) method is used as a higher-fidelity alternative to BEM. While it requires more computational resources, it remains significantly less demanding than fully resolved CFD methods. Using FVW methods in aeroelastic simulations remains too computationally intensive for it to be considered a viable design tool at present. This work aimed to explore potential methods of solver coupling to reduce computational costs without compromising the accuracy of aeroelastic simulations. To achieve this, various coupling strategies were assessed. It was found that the two most common approaches to solving aeroelastic problems are monolithic methods and partitioned coupling techniques. The former, which involves using a single solver to handle both physics simultaneously, is not manageable for realistic problems such as wind turbine simulations due to its complexity and computational demands. Additionally, in the context of design, this approach is impractical, as the entire problem would need to be re-adapted for each new design, leading to prohibitively high costs. In contrast, partitioned methods are more suitable for simulating coupled systems for design purposes. These methods allow each solver, whether for aerodynamic or structural physics, to be developed, optimised, and adapted independently. This flexibility is particularly valuable in the design process, where different aspects of the simulation may need to be tailored for specific configurations, making partitioned methods more practical for realistic applications. The different partitioned coupling techniques were presented, and their effects were initially evaluated using a simple linear model where a structural oscillator was coupled with a fluid oscillator. This preliminary analysis led to several key observations. Primarily, it was found that when using a Conventional Serial Stag-

gered (CSS) coupling scheme, the numerical stability limit is often dictated by the time step of the most restrictive physics. However, by implementing a subcycling scheme and thus allowing the two oscillators to integrate using different time steps, the numerical stability could be extended without accuracy deterioration. Building on this, a subcycling technique was implemented into DeepLines Wind™ to analyse the effects of partitioned schemes on a more realistic test case. The subcycling algorithm was modified to fit within the solver's coupling architecture, utilising an intermediate meta-model to handle communication between the aerodynamic and structural solvers. This meta-model forecasts aerodynamic forces by extrapolating data from previous iterations of the aerodynamic solver. This architecture also leaves room for the integration of more advanced forecasting techniques in the future. In this study, constant, linear, and quadratic polynomial regression techniques were applied within the subcycling scheme. The impact of the subcycling scheme was then assessed through aeroelastic simulations of a large-scale wind turbine, specifically the IEA 15MW model. The turbine was first subjected to steady wind conditions, followed by a turbulent wind case to evaluate the algorithm's performance for a wind turbine under more realistic operational scenarios. In both cases, the subcycling algorithm proved to be efficient and showed similar conclusions to the ones drawn in the simpler model. The coupling time step could be extended, allowing the aerodynamic problem to be solved at a slower rate, which significantly reduced computational costs. For example, an aeroelastic simulation that would take over 10 days to compute using a CSS scheme was reduced to just 36 minutes with the subcycling approach, which represents a significant speed-up of over 400 times, while maintaining minimal error in the aerodynamic force computations compared to the CSS method. This computational time gain can be attributed to the Free Vortex Wake method and the filament discretisation used to model the wake. By solving the aerodynamic problem with a less restrictive time step, the wake becomes less filament-dense, which directly reduces computational costs. Furthermore, it was observed that the FVW method is sensitive to several key parameters. To better understand the FVW method and the factors influencing aerodynamic force computations, a new Python-based Free Vortex Wake code was developed. This user-friendly code, intended to be made open source, facilitated the assessment of key parameters that require careful adjustment when using the method. Further analysis using the FVW C++ solver revealed that the method is particularly sensitive to factors such as the azimuthal discretisation of the wake, the wake length, the desingularisation technique applied in the Biot-Savart equation, and the vortex core radius. These elements must be chosen with care to ensure accurate aerodynamic force predictions.

In future works, these coupling techniques could be applied to fully coupled aero-hydro-servo-elastic simulations using the hydrodynamics model in DeepLines Wind™. This would allow for more efficient simulations of floating offshore wind turbines, where interactions between aerodynamic forces, structural deformation, and wave motion are crucial. Vortex methods are particularly well suited to such cases due to the unsteady inflow conditions, providing a higher-fidelity representation of aerodynamic forces.

To further reduce computational costs, wake coarsening techniques, as described in [1], could be combined with the subcycled coupling method. This approach would re-

quire careful evaluation of how wake resolution affects the accuracy of aerodynamic force predictions. By combining wake coarsening with subcycling, it may be possible to achieve efficient multi-physics simulations using vortex methods, potentially bringing them closer to being considered as viable design tools, similar to BEM.
In addition to the regression methods currently used in the meta-model architecture, exploring more advanced machine learning techniques, such as Long Short-Term Memory (LSTM) networks [140], could enhance the forecasting of aerodynamic forces under highly dynamic inflow conditions, where simpler approaches may struggle to capture complex temporal variations.
Finally, the Python-based Free Vortex Wake (FVW) solver, Pitchou, could be further refined by incorporating other desingularisation techniques and wake coarsening methods to improve computational efficiency. Coupling Pitchou with a structural solver would enable fully coupled aeroelastic simulations, with the added benefit of a Python-based code, making it more accessible and easier to modify for future developments.

Bibliography

[1] F. Blondel, P.-A. Joulin, and C. Le Guern, "Towards vortex-based wind turbine design using gpus and wake accommodation," in *Journal of Physics: Conference Series*, vol. 2767, p. 052016, IOP Publishing, 2024.

[2] C. Le Guern, F. Blondel, J.-L. Pfister, N. Guy, and M. Bergmann, "Partitioned time couplings of an aero-mechanical wind turbine problem.," in *Journal of Physics: Conference Series*, vol. 2767, p. 022012, IOP Publishing, 2024.

[3] R. Energy *et al.*, "Future of deployment investment technology grid integration and socio-economic aspects," 2019.

[4] E. Gaertner, J. Rinker, L. Sethuraman, F. Zahle, B. Anderson, G. Barter, N. Abbas, F. Meng, P. Bortolotti, W. Skrzypinski, *et al.*, "Definition of the iea 15-megawatt offshore reference wind turbine," 2020.

[5] A. van Garrel, "Development of a wind turbine aerodynamics simulation module," 2003.

[6] J. Jonkman, "Dynamics modeling and loads analysis of an offshore floating wind turbine [ph. d. thesis]," *University of Colorado, Boulder*, p. 237, 2007.

[7] C. Le Cunff, J.-M. Heurtier, L. Piriou, C. Berhault, T. Perdrizet, D. Teixeira, G. Ferrer, and J.-C. Gilloteaux, "Fully coupled floating wind turbine simulator based on nonlinear finite element method: Part i—methodology," in *International Conference on Offshore Mechanics and Arctic Engineering*, vol. 55423, p. V008T09A050, American Society of Mechanical Engineers, 2013.

[8] L. Fingersh, M. Hand, and A. Laxson, "Wind turbine design cost and scaling model," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2006.

[9] P. Veers, K. Dykes, E. Lantz, S. Barth, C. L. Bottasso, O. Carlson, A. Clifton, J. Green, P. Green, H. Holttinen, *et al.*, "Grand challenges in the science of wind energy," *Science*, vol. 366, no. 6464, p. eaau2027, 2019.

[10] P. Veers, C. L. Bottasso, L. Manuel, J. Naughton, L. Pao, J. Paquette, A. Robertson, M. Robinson, S. Ananthan, T. Barlas, *et al.*, "Grand challenges in the design, manufacture, and operation of future wind turbine systems," *Wind Energy Science*, vol. 8, no. 7, pp. 1071–1131, 2023.

[11] B. R. Resor, B. C. Owens, and D. T. Griffith, "Aeroelastic instability of very large wind turbine blades," in *European Wind Energy Conference Annual Event, Copenhagen, Denmark*, 2012.

[12] C. L. Kelley and J. Paquette, "Investigation of flutter for large, highly flexible wind turbine blades," in *Journal of physics: Conference series*, vol. 1618, p. 052078, IOP Publishing, 2020.

[13] W. S. Carron and P. Bortolotti, "Innovative rail transport of a supersized land-based wind turbine blade," in *Journal of Physics: Conference Series*, vol. 1618, p. 042041, IOP Publishing, 2020.

[14] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: theory, design and application*. John Wiley & Sons, 2010.

[15] V. N. Dinh and B. Basu, "On the modeling of spar-type floating offshore wind turbines," *Key Engineering Materials*, vol. 569, pp. 636–643, 2013.

[16] M. H. Hansen, "Aeroelastic instability problems for wind turbines," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 10, no. 6, pp. 551–577, 2007.

[17] J. Jonkman, "Definition of a 5-mw reference wind turbine for offshore system development," *National Renewable Energy Laboratory*, 2009.

[18] C. Bak, F. Zahle, R. Bitsche, T. Kim, A. Yde, L. C. Henriksen, M. H. Hansen, J. P. A. A. Blasques, M. Gaunaa, and A. Natarajan, "The dtu 10-mw reference wind turbine," in *Danish wind power research 2013*, 2013.

[19] F. Zahle, T. Barlas, K. Lonbaek, P. Bortolotti, D. Zalkind, L. Wang, C. Labuschagne, L. Sethuraman, and G. Barter, "Definition of the iea wind 22-megawatt offshore reference wind turbine," tech. rep., National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2024.

[20] R. L. Bisplinghoff, H. Ashley, and R. L. Halfman, *Aeroelasticity*. Courier Corporation, 2013.

[21] E. H. Dowell, *A modern course in aeroelasticity*, vol. 217. Springer, 2014.

[22] A. Collar, "The first fifty years of aeroelasticity," *Aerospce 2.5*, pp. 12–20, 1978.

[23] J. Holierhoek, "An overview of possible aeroelastic instabilities for wind turbine blades," *Wind Engineering*, vol. 37, no. 4, pp. 421–440, 2013.

[24] D. T. Griffith and M. Chetan, "Assessment of flutter prediction and trends in the design of large-scale wind turbine rotor blades," in *Journal of Physics: Conference Series*, vol. 1037, p. 042008, IOP Publishing, 2018.

[25] P. Pourazarm, Y. Modarres-Sadeghi, and M. Lackner, "A parametric study of coupled-mode flutter for mw-size wind turbine blades," *Wind Energy*, vol. 19, no. 3, pp. 497–514, 2016.

[26] X. Hua, Q. Meng, B. Chen, and Z. Zhang, "Structural damping sensitivity affecting the flutter performance of a 10-mw offshore wind turbine," *Advances in Structural Engineering*, vol. 23, no. 14, pp. 3037–3047, 2020.

[27] A. S. Escalera Mendoza, I. Mishra, and D. T. Griffith, "An open-source numad model for the iea 15 mw blade with baseline structural analysis," in *AIAA SCITECH 2023 Forum*, p. 2093, 2023.

[28] J. C. Heinz, N. N. Sørensen, F. Zahle, and W. Skrzypiński, "Vortex-induced vibrations on a modern wind turbine blade," *Wind Energy*, vol. 19, no. 11, pp. 2041–2051, 2016.

[29] B. Dose, H. Rahimi, I. Herráez, B. Stoevesandt, and J. Peinke, "Fluid-structure coupled computations of the nrel 5mw wind turbine blade during standstill," in *Journal of Physics: Conference Series*, vol. 753, p. 022034, IOP Publishing, 2016.

[30] S. Horcas, T. Barlas, F. Zahle, and N. Sørensen, "Vortex induced vibrations of wind turbine blades: Influence of the tip geometry," *Physics of Fluids*, vol. 32, no. 6, 2020.

[31] S. Horcas, N. Sørensen, F. Zahle, G. Pirrung, and T. Barlas, "Vibrations of wind turbine blades in standstill: Mapping the influence of the inflow angles," *Physics of Fluids*, vol. 34, no. 5, 2022.

[32] G. J. Leishman, *Principles of helicopter aerodynamics with CD extra*. Cambridge university press, 2006.

[33] J. Loubeyres, J.-L. Pfister, F. Blondel, and N. Guy, "Stall flutter instabilities on the iea-15 reference wind turbine in idling conditions: Code-to-code comparisons and physical analyses," in *Journal of Physics: Conference Series*, vol. 2265, p. 032019, IOP Publishing, 2022.

[34] K. Wang, V. A. Riziotis, and S. G. Voutsinas, "Aeroelastic stability of idling wind turbines," *Wind Energy Science*, vol. 2, no. 2, pp. 415–437, 2017.

[35] G. Bir and J. Jonkman, "Aeroelastic instabilities of large offshore and onshore wind turbines," in *Journal of Physics: Conference Series*, vol. 75, p. 012069, IOP Publishing, 2007.

[36] M. O. L. Hansen, J. N. Sørensen, S. Voutsinas, N. Sørensen, and H. A. Madsen, "State of the art in wind turbine aerodynamics and aeroelasticity," *Progress in aerospace sciences*, vol. 42, no. 4, pp. 285–330, 2006.

[37] H. Glauert, "Airplane propellers," *Aerodynamic theory*, 1935.

[38] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Wind energy handbook*. John Wiley & Sons, 2011.

[39] A. Ning, G. Hayman, R. Damiani, and J. M. Jonkman, "Development and validation of a new blade element momentum skewed-wake model within aerodyn," in *33rd Wind Energy Symposium*, p. 0215, 2015.

[40] R. Corniglion, *aero-elastic modeling of floating wind turbines with vortex methods*. PhD thesis, Marne-la-vallée, ENPC, 2022.

[41] T. Sebastian, *The aerodynamics and near wake of an offshore floating horizontal axis wind turbine*. PhD thesis, 2012.

[42] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *The computer journal*, vol. 14, no. 4, pp. 422–425, 1971.

[43] L. Prandtl and A. Betz, *Vier abhandlungen zur hydrodynamik und aerodynamik*, vol. 3. Universitätsverlag Göttingen, 2010.

[44] W. Z. Shen, R. Mikkelsen, J. N. Sørensen, and C. Bak, "Tip loss corrections for wind turbine computations," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 8, no. 4, pp. 457–475, 2005.

[45] M. L. Buhl Jr, "New empirical relationship between thrust coefficient and induction factor for the turbulent windmill state," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2005.

[46] J. Leishman and T. Beddoes, "A generalised model for airfoil unsteady aerodynamic behaviour and dynamic stall using the indicial method," in *Proceedings of the 42nd Annual forum of the American Helicopter Society*, vol. 34, pp. 3–17, Washington DC, 1986.

[47] M. H. Hansen, M. Gaunaa, and H. A. Madsen, *A Beddoes-Leishman type dynamic stall model in state-space and indicial formulations*. 2004.

[48] T. Knudsen and T. Bak, "Simple model for describing and estimating wind turbine dynamic inflow," in *2013 American Control Conference*, pp. 640–646, IEEE, 2013.

[49] J. De Vaal, M. L. Hansen, and T. Moan, "Effect of wind turbine surge motion on rotor thrust and induced velocity," *Wind Energy*, vol. 17, no. 1, pp. 105–121, 2014.

[50] S. A. Ning, "A simple solution method for the blade element momentum equations with guaranteed convergence," *Wind Energy*, vol. 17, no. 9, pp. 1327–1345, 2014.

[51] H. Glauert, "A general theory of the autogyro," tech. rep., HM Stationery Office, 1926.

[52] W. Z. Shen, W. J. Zhu, and J. N. Sørensen, "Actuator line/navier–stokes computations for the mexico rotor: comparison with detailed measurements," *Wind energy*, vol. 15, no. 5, pp. 811–825, 2012.

[53] E. S. P. Branlard, "Flexible multibody dynamics using joint coordinates and the rayleigh-ritz approximation: The general framework behind and beyond flex," *Wind Energy*, vol. 22, no. 7, pp. 877–893, 2019.

[54] J. J. Bertin and R. M. Cummings, *Aerodynamics for engineers*. Cambridge University Press, 2021.

[55] B. W. McCormick, *Aerodynamics, aeronautics, and flight mechanics*. John Wiley & Sons, 1994.

[56] A. M. Kuethe and C.-Y. Chow, *Foundations of aerodynamics: bases of aerodynamic design*. John Wiley & Sons, 1997.

[57] W. Johnson, *Rotorcraft aeromechanics*, vol. 36. Cambridge university press, 2013.

[58] E. Branlard, "Wind turbine aerodynamics and vorticity-based methods," 2020.

[59] A. Leonard, "Vortex methods for flow simulation," *journal of computational physics*, vol. 37, no. 3, pp. 289–335, 1980.

[60] J. N. Sørensen and A. Myken, "Unsteady actuator disc model for horizontal axis wind turbines," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 39, no. 1-3, pp. 139–149, 1992.

[61] F. Castellani and A. Vignaroli, "An application of the actuator disc model for wind turbine wakes calculations," *Applied Energy*, vol. 101, pp. 432–440, 2013.

[62] N. Sorensen, J. Michelsen, and S. Schreck, "Navier-stokes predictions of the nrel phase vi rotor in the nasa ames 80-by-120 wind tunnel," in *Wind Energy Symposium*, vol. 7476, pp. 94–105, 2002.

[63] Y. Liu, Q. Xiao, A. Incecik, and C. Peyrard, "Aeroelastic analysis of a floating offshore wind turbine in platform-induced surge motion using a fully coupled cfd-mbd method," *Wind Energy*, vol. 22, no. 1, pp. 1–20, 2019.

[64] P.-A. Joulin, *Modélisation à fine échelle des interactions entre parcs éoliens et météorologie locale*. PhD thesis, 2019.

[65] B. Wiegard, M. König, J. Lund, L. Radtke, S. Netzband, M. Abdel-Maksoud, and A. Düster, "Fluid-structure interaction and stress analysis of a floating wind turbine," *Marine structures*, vol. 78, p. 102970, 2021.

[66] S. Scott, P. Greaves, T. Macquart, and A. Pirrera, "Comparison of blade optimisation strategies for the iea 15mw reference turbine," in *Journal of Physics: Conference Series*, vol. 2265, p. 032029, IOP Publishing, 2022.

[67] M. H. Hansen, "Modal dynamics of structures with bladed isotropic rotors and its complexity for two-bladed rotors," *Wind Energy Science*, vol. 1, no. 2, pp. 271–296, 2016.

[68] A. Staino and B. Basu, "Dynamics and control of vibrations in wind turbines with variable rotor speed," *Engineering Structures*, vol. 56, pp. 58–67, 2013.

[69] A. Cadoret, *Operational Modal Analysis (OMA) for wind turbines health monitoring.* PhD thesis, Université de Rennes, 2023.

[70] N. Abbas, D. Zalkind, L. Pao, and A. Wright, *A reference open-source controller for fixed and floating offshore wind turbines.* PhD thesis, 2021.

[71] F. Blondel, R. Boisard, M. Milekovic, G. Ferrer, C. Lienard, and D. Teixeira, "Validation and comparison of aerodynamic modelling approaches for wind turbines," in *Journal of Physics: Conference Series*, vol. 753, p. 022029, IOP Publishing, 2016.

[72] T. Theodorsen, "General theory of aerodynamic instability and the mechanism of flutter," tech. rep., 1949.

[73] H. Wagner, "Über die entstehung des dynamischen auftriebes von tragflügeln," 1924.

[74] W. Skrzypiński and M. Gaunaa, "Wind turbine blade vibration at standstill conditions—the effect of imposing lag on the aerodynamic response of an elastically mounted airfoil," *Wind Energy*, vol. 18, no. 3, pp. 515–527, 2015.

[75] Y. Yu, X. Amandolese, C. Fan, and Y. Liu, "Experimental study and modelling of unsteady aerodynamic forces and moment on flat plate in high amplitude pitch ramp motion," *Journal of Fluid Mechanics*, vol. 846, pp. 82–120, 2018.

[76] R. T. Jones, *The unsteady lift of a wing of finite aspect ratio*, vol. 681. 1940.

[77] C. Farhat, M. Lesoinne, and P. Le Tallec, "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity," *Computer methods in applied mechanics and engineering*, vol. 157, no. 1-2, pp. 95–114, 1998.

[78] C. Farhat and M. Lesoinne, "Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems," *Computer methods in applied mechanics and engineering*, vol. 182, no. 3-4, pp. 499–515, 2000.

[79] C. Farhat, K. G. Van der Zee, and P. Geuzaine, "Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity," *Computer methods in applied mechanics and engineering*, vol. 195, no. 17-18, pp. 1973–2001, 2006.

[80] S. Piperno, C. Farhat, and B. Larrouturou, "Partitioned procedures for the transient solution of coupled aroelastic problems part i: Model problem, theory and two-dimensional application," *Computer methods in applied mechanics and engineering*, vol. 124, no. 1-2, pp. 79–112, 1995.

[81] S. Piperno and C. Farhat, "Partitioned procedures for the transient solution of coupled aeroelastic problems–part ii: energy transfer analysis and three-dimensional applications," *Computer methods in applied mechanics and engineering*, vol. 190, no. 24-25, pp. 3147–3170, 2001.

[82] J. Degroote, "Partitioned simulation of fluid-structure interaction: Coupling black-box solvers with quasi-newton techniques," *Archives of computational methods in engineering*, vol. 20, no. 3, pp. 185–238, 2013.

[83] F. J. Blom, "A monolithical fluid-structure interaction algorithm applied to the piston problem," *Computer methods in applied mechanics and engineering*, vol. 167, no. 3-4, pp. 369–391, 1998.

[84] F. Ischinger, M. Anthonissen, and B. Koren, "A monolithic fluid-structure interaction method, application to a piston problem," in *7th European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2016*, pp. 1564–1581, National Technical University of Athens, 2016.

[85] C. Michler, S. Hulshoff, E. Van Brummelen, and R. De Borst, "A monolithic approach to fluid–structure interaction," *Computers & fluids*, vol. 33, no. 5-6, pp. 839–848, 2004.

[86] C. A. Felippa, K.-C. Park, and C. Farhat, "Partitioned analysis of coupled mechanical systems," *Computer methods in applied mechanics and engineering*, vol. 190, no. 24-25, pp. 3247–3270, 1999.

[87] J.-D. Garaud, *Développement de méthodes de couplage aéro-thermo-mécanique pour la prédiction d'instabilités dans les structures aérospatiales chaudes*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2008.

[88] M. L. Facchinetti, E. De Langre, and F. Biolley, "Coupling of structure and wake oscillators in vortex-induced vibrations," *Journal of Fluids and structures*, vol. 19, no. 2, pp. 123–140, 2004.

[89] E. De Langre, "Frequency lock-in is caused by coupled-mode flutter," *Journal of fluids and structures*, vol. 22, no. 6-7, pp. 783–791, 2006.

[90] J. C. Butcher, "General linear methods," *Acta Numerica*, vol. 15, pp. 157–256, 2006.

[91] Z. Jackiewicz, *General linear methods for ordinary differential equations*. John Wiley & Sons, 2009.

[92] S. Karaa, "Improved accuracy of linear multistep methods," *Applied Mathematics & Information Sciences*, vol. 7, no. 2, pp. 491–496, 2013.

[93] E. A. Celaya and J. J. Anza, "Bdf-α: A multistep method with numerical damping control," *System*, vol. 500, p. 9, 2013.

[94] J. Chung and G. Hulbert, "A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method," 1993.

[95] O. Brüls and M. Arnold, "The generalized-α scheme as a linear multistep integrator: toward a general mechatronic simulator," 2008.

[96] C. A. Felippa and K.-C. Park, "Staggered transient analysis procedures for coupled mechanical systems: formulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 24, no. 1, pp. 61–111, 1980.

[97] J. C. Heinz, N. N. Sørensen, and F. Zahle, "Partitioned fluid-structure interaction for full rotor computations using cfd," *PhD thesis, Technical University of Denmark*, 2013.

[98] J. C. Butcher, *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[99] N. M. Newmark, "A method of computation for structural dynamics," *Journal of the engineering mechanics division*, vol. 85, no. 3, pp. 67–94, 1959.

[100] A. de Boer, A. H. van Zuijlen, and H. Bijl, "Review of coupling methods for non-matching meshes," *Computer methods in applied mechanics and engineering*, vol. 196, no. 8, pp. 1515–1525, 2007.

[101] N. Dimitrov, M. C. Kelly, A. Vignaroli, and J. Berg, "From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases," *Wind Energy Science*, vol. 3, no. 2, pp. 767–790, 2018.

[102] J. R. Cebral and R. Lohner, "Conservative load projection and tracking for fluid-structure problems," *AIAA journal*, vol. 35, no. 4, pp. 687–692, 1997.

[103] A. Beckert and H. Wendland, "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions," *Aerospace Science and Technology*, vol. 5, no. 2, pp. 125–134, 2001.

[104] F.-E. Fekak, M. Brun, A. Gravouil, and B. Depale, "A new heterogeneous asynchronous explicit–implicit time integrator for nonsmooth dynamics," *Computational mechanics*, vol. 60, pp. 1–21, 2017.

[105] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[106] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[107] S. Bisgaard and M. Kulahci, *Time series analysis and forecasting by example*. John Wiley & Sons, 2011.

[108] S. Wheelwright, S. Makridakis, and R. J. Hyndman, *Forecasting: methods and applications*. John Wiley & Sons, 1998.

[109] K. Shaler, B. Anderson, L. A. Martínez-Tossas, E. Branlard, and N. Johnson, "Comparison of free vortex wake and bem structural results against large eddy simulations results for highly flexible turbines under challenging inflow conditions," *Wind Energy Science Discussions*, vol. 2022, pp. 1–22, 2022.

[110] S. Hauptmann, M. Bülk, L. Schön, S. Erbslöh, K. Boorsma, F. Grasso, M. Kühn, and P. Cheng, "Comparison of the lifting-line free vortex wake method and the blade-element-momentum theory regarding the simulated loads of multi-mw wind turbines," in *Journal of Physics: Conference Series*, vol. 555, p. 012050, IOP Publishing, 2014.

[111] C. Mimeau and I. Mortazavi, "A review of vortex methods and their applications: From creation to recent advances," *Fluids*, vol. 6, no. 2, p. 68, 2021.

[112] H. Lee, B. Sengupta, M. Araghizadeh, and R. Myong, "Review of vortex methods for rotor aerodynamics and wake dynamics," *Advances in Aerodynamics*, vol. 4, no. 1, p. 20, 2022.

[113] G. Cottet, "Vortex methods: Theory and practice," 2000.

[114] L. Vermeer, J. N. Sørensen, and A. Crespo, "Wind turbine wake aerodynamics," *Progress in aerospace sciences*, vol. 39, no. 6-7, pp. 467–510, 2003.

[115] D. A. Griffiths, S. Ananthan, and J. G. Leishman, "Predictions of rotor performance in ground effect using a free-vortex wake model," *Journal of the American Helicopter Society*, vol. 50, no. 4, pp. 302–314, 2005.

[116] H. Glauert, *The elements of aerofoil and airscrew theory*. Cambridge university press, 1983.

[117] B. Montgomerie, "Vortex model for wind turbine loads and performance evaluation," 2006.

[118] K. Boorsma, L. Greco, and G. Bedon, "Rotor wake engineering models for aeroelastic applications," in *Journal of Physics: Conference Series*, vol. 1037, p. 062013, IOP Publishing, 2018.

[119] M.-A. Dufour, G. Pinon, E. Rivoalen, F. Blondel, and G. Germain, "Development and validation of a lifting-line code associated with the vortex particle method software dorothy," *Wind Energy*, 2024.

[120] K. Shaler, E. Branlard, and A. Platt, "Olaf user's guide and theory manual," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2020.

[121] C. Greengard, "Convergence of the vortex filament method," *Mathematics of Computation*, vol. 47, no. 176, pp. 387–398, 1986.

[122] J. T. Beale and A. Majda, "High order accurate vortex methods with explicit velocity kernels," *Journal of Computational Physics*, vol. 58, no. 2, pp. 188–208, 1985.

[123] M. Ramasamy and J. G. Leishman, "Interdependence of diffusion and straining of helicopter blade tip vortices," *Journal of Aircraft*, vol. 41, no. 5, pp. 1014–1024, 2004.

[124] J. Jonkman, "Fast user's guide," *National Renewable Energy Laboratory Technical Report*, 2005.

[125] H. v. Helmholtz, "Über integrale der hydrodynamischen gleichungen, welche den wirbelbewegungen entsprechen.," 1858.

[126] S. D. Poisson, "Remarques sur une équation qui se présente dans la théorie des attractions des sphéroïdes," *Bulletin de la Société Philomathique de Paris*, vol. 3, pp. 388–392, 1813.

[127] J. Katz and A. Plotkin, *Low-speed aerodynamics*, vol. 13. Cambridge university press, 2001.

[128] G. S. Winckelmans, *Topics in vortex methods for the computation of three- and two-dimensional incompressible unsteady flows*. California Institute of Technology, 1989.

[129] G. H. Vatistas, V. Kozel, and W. Mih, "A simpler model for concentrated vortices," *Experiments in Fluids*, vol. 11, pp. 73–76, 1991.

[130] F. Blondel, G. Ferrer, D. Teixeira, and M. Cathelain, "Improving a bem yaw model based on newmexico experimental data and vortex/cfd simulations," in *CFM 2017-23ème Congrès Français de Mécanique*, AFM, Maison de la Mécanique, 39/41 rue Louis Blanc-92400 Courbevoie, 2017.

[131] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, A. Fasih, A. Sarma, D. Nanongkai, G. Pandurangan, P. Tetali, *et al.*, "Pycuda: Gpu run-time code generation for high-performance computing," *Arxiv preprint arXiv*, vol. 911, p. 3456, 2009.

[132] J. Aycock, "A brief history of just-in-time," *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 97–113, 2003.

[133] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A llvm-based python jit compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pp. 1–6, 2015.

[134] H. Snel, J. Schepers, and B. Montgomerie, "The mexico project (model experiments in controlled conditions): The database and first results of data processing and interpretation," in *Journal of Physics: Conference Series*, vol. 75, p. 012014, IOP Publishing, 2007.

[135] C. Stangfeld, C. L. Rumsey, H. Mueller-Vahl, D. Greenblatt, C. Nayeri, and C. O. Paschereit, "Unsteady thick airfoil aerodynamics: experiments, computation, and theory," in *45th AIAA Fluid Dynamics Conference*, p. 3071, 2015.

[136] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "Su2: An open-source suite for multiphysics simulation and design," *Aiaa Journal*, vol. 54, no. 3, pp. 828–846, 2016.

[137] R. Kurniawan, "Numerical study of flutter of a two-dimensional aeroelastic system," *International Scholarly Research Notices*, vol. 2013, 2013.

[138] T. A. Zeiler, "Results of theodorsen and garrick revisited," *Journal of Aircraft*, vol. 37, no. 5, pp. 918–920, 2000.

[139] B. J. Jonkman, "Turbsim user's guide: Version 1.50," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2009.

[140] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.