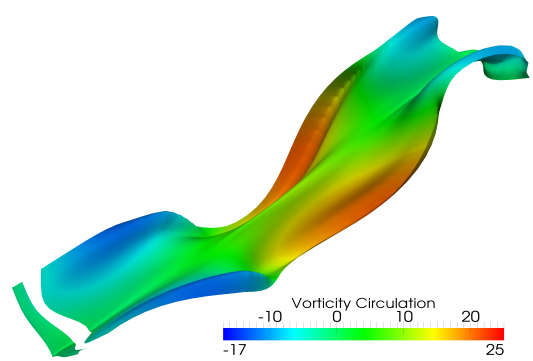
PyFly: Fast, Friendly Framework for Aerodynamic Simulation

Mehdi Ghommem, Nathan Collier, and Victor M. Calo

Physical Sciences and Engineering Division
King Abdullah University of Science and Technology (KAUST)
Thuwal 23955-6900, Kingdom of Saudi Arabia

The performance of multidisciplinary engineered systems (e.g., aerospace vehicles, ships, submarines, offshore structures, and wind turbines) could be significantly enhanced at the earliest stages of design through understanding the governing dynamics and possibly exploiting specific phenomena. Equations governing the performance of these systems may comprise nonlinear partial differential equations (e.g., Navier-Stokes equations). Recent advances in computer hardware and software in terms of speed and storage capacity have enabled the use of numerical simulations whereby, the afore-mentioned equations are discretized and integrated with robust numerical algorithms. While these high-fidelity approaches are powerful in terms of capturing the main physical features, they involve multiple phenomena that are interrelated in a complex manner which require a large number of degrees of freedom to reproduce them. Furthermore, the extensive computational resources and time associated with the use of these tools could limit the capability to simulate a large number of configurations for design purposes. These shortcomings lead to the need for developing simplified simulation tools with significantly reduced computational cost while embodying relevant physical aspects and response characteristics.

In this work, we present a fast and efficient implementation of a potential flow solver based on the unsteady vortex lattice method (UVLM), namely ***PyFly***. This computational tool can be used to simulate the unsteady aerodynamic behavior of moving and deforming bodies such as flapping wings, rotating blades, and swimming fish. UVLM computes the forces generated by pressure differences across the wing surface resulting from acceleration- and circulation-based phe-



Unsteady vortex lattice method simulation of the wake behind the flapping wings.

nomena. This accounts for unsteady effects such as added mass forces, the growth of bound circulation, and the wake. UVLM applies only to ideal fluids, incompressible, inviscid, and irrotational flows where the separation lines are known a priori. Thus, the formulation of UVLM requires that fluid leaves the wing smoothly at the trailing edge (through imposing the Kutta condition), it does not cover the cases of flow separation at the leading-edge and extreme situations where strong wing-wake interactions take place. In spite all of these restrictions, several research efforts have considered the use of UVLM for the design of avian-like flapping wings in forward and hover flights [1–4].

While fast runtime is usually the target of scientific software projects, we recognize that a simple user-interface is also an important aspect of a framework's usage. An efficient framework which is complex to understand and use will not reduce the solution time for an engineer, despite the fact that the resulting code executes quickly. However, languages which are easy to use are frequently orders of magnitude slower in performance. Neither situation is ideal. The goal of ***PyFly*** is to provide a friendly framework for aerodynamic simulations based on the UVLM which is also computationally efficient. We achieve this by using mixed-language programming. We use python [5] for high-level management of grid objects and Fortran for the computational kernels which must run efficiently.

While the numerical method does not change for different applications, the requirements which different applications present can become complex to manage. For example in the case of flapping wings, there is the wing and its wake to manage. For symmetric flights, we must also track the effect of the wing's mirror image. However, in the case of a wind turbine, each rotating blade and its wake is modeled as a separate object which interacts with the entire system.

To simplify user-burden in writing solvers for different scenarios, we implement a base grid class in python which handles the grid itself and its wake. Furthermore it can compute its influence on other grid objects. This abstract approach makes the framework simple to use, while core components run in Fortran for efficiency. Mixed language programming is a growing trend in the field of scientific software [6].

We present and analyze different configurations for flows past wings to demonstrate the ability of the UVLM to appropriately capture the flow around moving bodies and accurately predict the associated aerodynamic loads. Results from these configurations are compared against those obtained from numerical and experimental studies. Furthermore, we show the suitability of ***PyFly*** to perform optimization analyses as required for design purposes.

References

[1] M. Ghommem, M. R. Hajj, D. T. Mook, B. K. Stanford, P. S. Beran, L. T. Watson, Global optimization of actively-morphing flapping wings, Journal of Fluids and Structures 30 (2012) 210–228.

[2] B. K. Stanford, P. S. Beran, Analytical sensitivity analysis of an unsteady vortex-lattice method for flapping-wing optimization, Journal of Aircraft 47 (2010) 647–662.

[3] M. Ghommem, N. Collier, A. H. Niemi, V. M. Calo, Shape optimization and performance analysis of flapping wings, in: Proceedings of The Eighth International Conference on Engineering Computational Technology, 2012.

[4] P. Persson, D. Willis, J. Peraire, Numerical simulation of flapping wings using a panel method and high-order Navier-Stokes solver, International Journal for Numerical Methods in Engineering 89 (2012) 1296–1316.

[5] H. P. Langtangen, Python Scripting for Computational Science, Springer, 2010.

[6] D. I. Ketcheson, K. T. Mandly, A. J. Ahmadia, A. Alghamdi, M. Q. D. Luna, M. Parsani, M. G. Knepley, M. Emmett, Pyclaw: Accessible, extensible, scalable tools for wave propagation problems, SIAM: SIAM Journal on Scientific Computing 34 (2012) 210–231.