Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2020-0246

Vol. **30**, No. 4, pp. 1118-1149
October 2021

A Wavelet-Adaptive Method for Multiscale Simulation of Turbulent Flows in Flying Insects

Thomas Engels[1,4,*], Kai Schneider[2], Julius Reiss[3] and Marie Farge[4]

[1] *Institute of Biosciences, University of Rostock, Rostock, Germany.*
[2] *Aix-Marseille Université, CNRS, I2M UMR 7373, Marseille, France.*
[3] *ISTA, Technische Universität Berlin, Berlin, Germany.*
[4] *LMD UMR 8539 École Normale Supérieure-PSL, Paris, France.*

Received 4 December 2020; Accepted (in revised version) 7 May 2021

Abstract. We present a wavelet-based adaptive method for computing 3D multiscale flows in complex, time-dependent geometries, implemented on massively parallel computers. While our focus is on simulations of flapping insects, it can be used for other flow problems. We model the incompressible fluid with an artificial compressibility approach in order to avoid solving elliptical problems. No-slip and in/outflow boundary conditions are imposed using volume penalization. The governing equations are discretized on a locally uniform Cartesian grid with centered finite differences, and integrated in time with a Runge–Kutta scheme, both of 4th order. The domain is partitioned into cubic blocks with different resolution and, for each block, biorthogonal interpolating wavelets are used as refinement indicators and prediction operators. Thresholding the wavelet coefficients allows to generate dynamically evolving grids, and an adaption strategy tracks the solution in both space and scale. Blocks are distributed among MPI processes and the grid topology is encoded using a tree-like data structure. Analyzing the different physical and numerical parameters allows us to balance their errors and thus ensures optimal convergence while minimizing computational effort. Different validation tests score accuracy and performance of our new open source code, `WABBIT`. Flow simulations of flapping insects demonstrate its applicability to complex, bio-inspired problems.

AMS subject classifications: 65M06, 65M50, 65M55, 65M85, 65T60, 76M20, 76D05, 76Z10

Key words: Adaptive numerical method, wavelets, volume penalization, artificial compressibility, flapping flight.

*Corresponding author. *Email addresses:* `thomas.engels@ens.fr` (T. Engels), `kai.schneider@univ-amu.fr` (K. Schneider), `julius.reiss@tnt.tu-berlin.de` (J. Reiss), `marie.farge@ens.fr` (M. Farge)

1 Introduction

Computing multiscale flows in complex geometries, which may move or deform, is required in numerous applications, *e.g.*, many biological flow problems such as flying insects or beating hearts. This remains a major challenge for computational fluid dynamics, especially in the turbulent flow regime. To simulate turbulent flows in the presence of moving boundaries, numerical techniques are needed that allow the solution to be tracked in both space and scale, and the numerical grid to be dynamically adapted accordingly. If the fluid-structure interaction must also be taken into account, it is all the more difficult since the motion of the boundary is no longer known *a priori*, but depends on its nonlinear interaction with the fluid. In this regard adaptive numerical discretization methods, which can be traced back to the 1980s [6,7], are indeed attractive. In many cases they can be much more competitive than schemes on uniform fine grids, depending on the character of the solution. However, for adaptive discretizations two major challenges can be identified: their actual implementation on massively parallel supercomputers and the numerical error analysis of adaptivity.

The implementation of the code is crucial to optimize computing, and two conceptually different approaches can be distinguished: one uses point-based techniques, while the other uses block-based techniques. In the former, the error indicator determines for each grid point whether it is significant or not (*e.g.*, [45,58]), while in the latter significant grid points are clustered in patches with the drawback of including non-significant points and thus decreasing the compression rate [24]. Due to the hardware layout of modern CPU, block-based implementations are in many cases more competitive and have become increasingly popular during the last years, see, *e.g.*, [62] for a recent review on available software packages. The locally regular block data can be transferred in one contiguous chunk to the CPU cache, which greatly increases the performance despite an increase in the number of points.

The mathematical support for adaptivity needs to provide reliable error estimators of the solution and, for evolutionary problems, a prediction of the grid used to compute the next time step. For both, a variety of heuristic criteria exists, *e.g.*, gradient-based approaches [18]. Adaptive mesh refinement algorithms use these heuristic criteria and are *error-indicated* methods because of their heuristic nature. *A posteriori* error estimators [4] are mathematically rigorous but require solving expensive adjoint problems.

Wavelets and related multiresolution analysis techniques provide likewise a mathematical framework and yield reliable error estimators, coupled with high computational efficiency; thus they are well suited for developing adaptive solvers with *error control*. The idea of wavelet analysis is to decompose data into contributions in both space and scale (and possibly direction). The wavelet transform has been introduced by Grossmann and Morlet [37], and the algorithm of the fast wavelet transform by Mallat [49]. Nonlinear approximation [21] provides the conceptual support for adaptivity; indeed, it introduces a systematic way to classify functions according to the sparsity of their representation in