# A wavelet-adaptive method for multiscale simulation of turbulent flows in flying insects: Electronic supplementary material

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

## A Approximation for first order derivatives

In the present work, we use the optimized fourth order scheme proposed by Tam and Webb [6] for first derivatives. It is built on the idea of combining a fourth- and a sixth order approximation using standard finite differences, yielding

$$(\partial_x u)_i \approx \Delta x^{-1} \left( \sum_j \left( \gamma a_j^{(4\text{th})} + (1-\gamma) a_j^{(6\text{th})} \right) u_{i+j} \right)$$

where $a_j^{(4\text{th})}$ and $a_j^{(6\text{th})}$ are classical, central finite differences. The combination provides a degree of freedom $\gamma$, which is then used to optimize the modified wavenumber. The resulting stencil is

$$a_j^{(\text{TW})} = -a_{-j}^{(\text{TW})} = \{-0.02651995, 0.18941314, -0.79926643, 0\}, \quad (-3 \leq j \leq 0).$$

## B Design of the computational grid

The design of our adaptive grid is tailored to the requirements of biorthogonal wavelets (nestedness) and designed with the idea of keeping the numerical code as simple as possible. We therefore use blocks of equal size $B_s^D$, as explained in the main text. Still, the precise definition of the grid is not unique using these requirements, which is why we justify our choices in detail in this appendix.

Figure 1A illustrates the two possibilities to define a block, where type A blocks include all borders and type B only half of them. Consequently, using type A, two adjacent

*Corresponding author. *Email addresses:* thomas.engels@ens.fr (T. Engels)

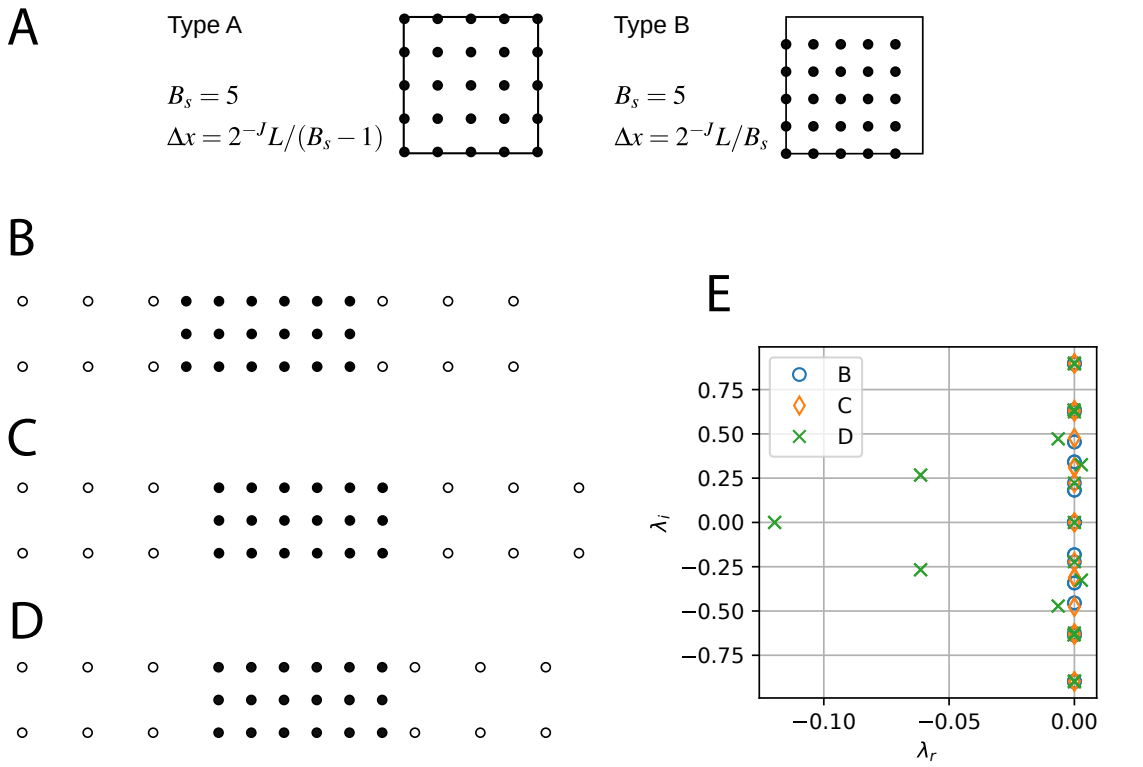


Figure 1: A: Two possible block definitions using the same block size $B_s$. Type A includes all of the block borders, which are then redundant with the neighboring block, while type B assigns each block border to one block. Present work uses type A grids exclusively. B-D: Different grid definitions for a coarse (open circles) and fine block (full circles). Ghost nodes are not shown. B: spacing between the blocks corresponds to fine block spacing on both sides. C: same as B, but using coarse block spacing. D: mixed interblock spacing E: eigenvalues of discrete first derivative (a Toeplitz matrix) for grids B-D in the complex plane.

blocks sharing a common interface both contain redundant points, which is not the case using blocks of type B. At first it may thus seem that the latter are preferable.

Different discretizations of the block interface with a jump in resolution are shown in Fig. 1B-D. The spacing between the block is equal to the fine (B) and coarse (C) block spacing, and symmetrical on both sides. In D, both are mixed. The eigenvalues of the discrete finite difference operator for the first derivative are shown in E. Clearly, grids B,C result in purely imaginary eigenvalues, while grid D also has non-zero real parts. While these are negative in the example shown, it depends on the local flow direction if the solution is excited or attenuated. From these eigenvalues, we hence conclude that grid D is unstable without an additional stabilization (e.g., filtering). We immediately note that grid D corresponds to a block-based grid using type B blocks, which is why we do not use this definition.

Blocks of type A allow stable discretizations shown in B,C, provided we attribute the border of a blocks consistently to either the fine or coarse block. This final decision was made based on the observation that using CDF 4/4 wavelets, spurious oscillations are

generated at resolution jumps if the interface is attributed to the fine block. These artifacts disappear if the interface is counted to the coarse block (grid C in Fig. 1). However, strictly speaking this grid definition is incompatible with the requirement for blocks of equal size. On the finer block, we therefore simply overwrite the solution with the coarse-block data at their common interface. This is done when synchronizing the ghost node layer.

# C CDF Wavelets

For completeness, the filter coefficients for the biorthogonal wavelets used in this work (CDF 4/0 and CDF 4/4) are given in Table 1. The resulting scaling functions $\phi$, wavelets $\psi$ and dual functions $\widetilde{\phi}$, $\widetilde{\psi}$, are illustrated in Fig. 2.

| | CDF4/0 | | | | CDF4/4 | | | |
|---|---|---|---|---|---|---|---|---|
| $i$ | $\widetilde{h}$ | $\widetilde{g}$ | $h$ | $g$ | $\widetilde{h}$ | $\widetilde{g}$ | $h$ | $g$ |
| -7 | | | | | | | | $-1/256$ |
| -6 | | | | | $-1/256$ | | | 0 |
| -5 | | | | | 0 | | | $9/128$ |
| -4 | | | | | $9/128$ | | | $1/16$ |
| -3 | | | $-1/16$ | | $-1/16$ | | $-1/16$ | $-63/256$ |
| -2 | | $1/16$ | 0 | | $-63/256$ | $1/16$ | 0 | $-9/16$ |
| -1 | | 0 | $9/16$ | | $9/16$ | 0 | $9/16$ | $87/64$ |
| 0 | 1 | $-9/16$ | 1 | | $87/64$ | $-9/16$ | 1 | $-9/16$ |
| 1 | | 1 | $9/16$ | 1 | $9/16$ | 1 | $9/16$ | $-63/256$ |
| 2 | | $-9/16$ | 0 | | $-63/256$ | $-9/16$ | 0 | $1/16$ |
| 3 | | 0 | $-1/16$ | | $-1/16$ | 0 | $-1/16$ | $9/128$ |
| 4 | | $1/16$ | | | $9/128$ | $1/16$ | | 0 |
| 5 | | | | | 0 | | | $-1/256$ |
| 6 | | | | | $-1/256$ | | | |

Table 1: Filter coefficients for the CDF4/0 and CDF4/4 interpolating biorthogonal wavelets.

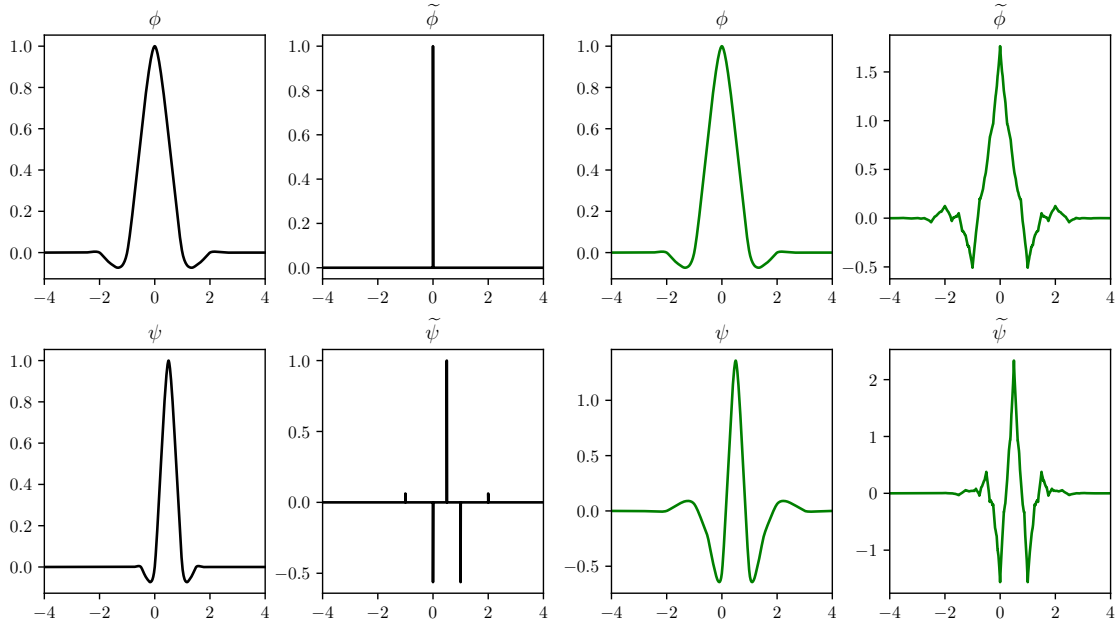


Figure 2: Scaling functions $\phi$, wavelets $\psi$ and dual functions $\widetilde{\phi}$, $\widetilde{\psi}$ for the CDF 4/0 (left, black) and CDF 4/4 wavelets (right, green).

# D Non-reflecting boundary conditions using sponges

At the outflow boundary of the computational domain, we impose non-reflecting boundary conditions using a sponge technique. The outflow removes two physically distinct phenomena: on the one hand, vortical structures not yet dissipated, and artificial sound waves on the other. The artificial sound waves transport flow divergence created at the fluid–solid interface and due to the nonlinear term. It is possible to implement such conditions based on characteristics, but for simplicity and consistency with the treatment of the obstacles, we use the volume penalization method to include non-reflecting sponges. We can conveniently discuss the choice of parameters for sponges using a simple, one-dimensional model. In the acoustic regime, we can neglect the nonlinear and diffusion terms, and we also remove the obstacle penalization term, yielding

$$\begin{aligned} \partial_t u &= -\partial_x p - \frac{\chi_{\mathrm{sp}}}{C_{\mathrm{sp}}} u \\ \partial_t p &= -\partial_x \underline{u} - \frac{\chi_{\mathrm{sp}}}{C_{\mathrm{sp}}} p, \end{aligned}$$

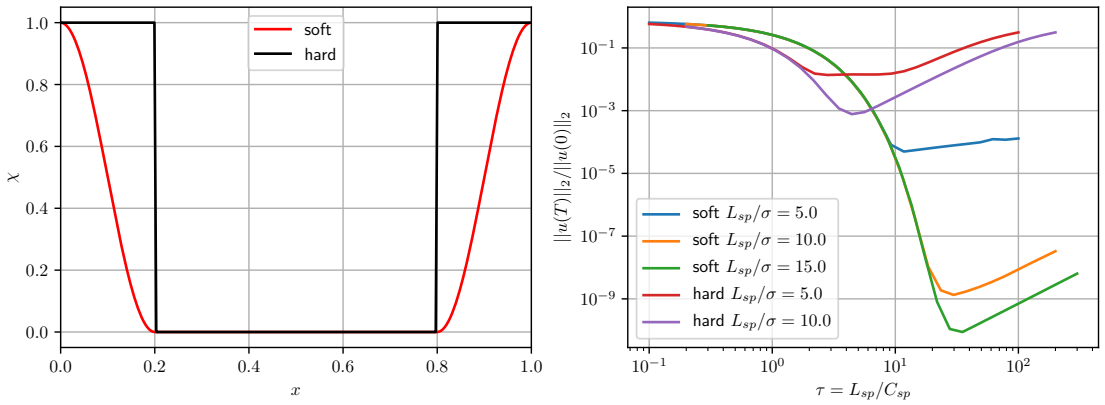


Figure 3: Left: 'Hard' and 'soft' $\chi_{sp}$ functions, here with $L_{sp} = 0.2$. The red function is cos-shaped. Right: sponge error, computed as $L^2$ norm of residual velocity at $t = T$. Because the exact solution at is zero, we use $\|u(t=0)\|_2$ as normalization.

where we further normalize $C_0 = 1$ without loss of generality. The reference state is $u_\infty = p_\infty = 0$. We consider a periodic domain of length $L$ and the initial condition

$$u(x, t=0) = \exp(-(x - L/2)^2/\sigma^2)$$
$$p(x, t=0) = 0.$$

This initial condition results in two artificial sound waves with characteristic size $\sigma$ traveling either left or right. The waves do not decay in a 1D setting, unlike in higher dimensions, making the 1D problem the most sensitive to non-reflecting boundary conditions. In 3D, the waves decay as $R^{-2}$, where $R$ is the distance to the source. Provided a sufficiently large domain, pseudoacoustic waves thus are already significantly weakened when reaching the sponge layer.

The size of the sponge layer should be larger than the waves it is supposed to absorb – the important ratio is thus $L_{sp}/\sigma$. The second relevant parameter is the time spent in the sponge layer, relative to the relaxation time $C_{sp}$. We call this parameter $\tau = L_{sp}/(C_0 C_{sp})$. The third relevant parameter is the shape $\chi_{sp}$ of the sponge layer, for which we can use either a discontinuous (hard) or smoothed (soft) function, as shown in Fig. 3 (left).

Depending of the choice of parameters, one of two things can happen: the damping is too weak, and the wave is not completely removed from the domain (it re-enters on the opposite side). The damping can also be too strong, in which case the sponge term deteriorates the regularity of the solution and spurious reflections occur.

We run the simulations until $T = 3L/4$; the artificial wave reaches the sponge around $T_0 = (L - L_{sp})/2$. The exact solution at $t = T$ is $u = p = 0$. Figure 3 (right) shows the normalized residual velocity at the final time $T$ for hard and soft mask function and different $L_{sp}/\sigma$, as a function of $\tau$. The curves coincide for small $\tau$, which indicates that $L_{sp}/\sigma$ and $\tau$ are indeed the relevant parameters. Thicker sponges allow setting larger

values of $\tau$, which reduces reflections. However, a significant improvement of the soft mask function over the discontinuous one is observed, and almost perfect cancellation of waves is observed for the largest $L_{sp}/\sigma$. From this data, we conclude that $L_{sp}/\sigma > 5$ and $\tau > 10$ minimize reflections. Sponges are usually set much larger than $L_{sp}/\sigma = 5$, because of vortical structures. We recommend setting $\tau = 20$ in the main text. Note that in practice, simulations are performed in 3D and the sensitivity to the sponge is then significantly reduced.

# E Description of parameter files required to reproduce simulations

This section summarizes and briefly explains the content of the electronic supplementary material (ESM). The intention of the ESM is to ensure reproducibility of all results presented in the main text.

### General remarks on running the code

The `WABBIT` code is started from a jobfile specific to the supercomputer. The general call to the code is `MPI_COMMAND ./wabbit PARAMETERS.ini -memory=250.0gb`. The parameter file might refer to other parameter files (for example for the wingbeat kinematics or the fractal tree).

### Three-vortices simulations

Files required to run a simulation:

- Main parameter file: `PARAMS-three-vortices.ini`
- Initial condition:
  `ux_000000000000.h5`, `uy_000000000000.h5`, `p_000000000000.h5`

As we present a total of 160 simulations for this problem, we decided to not include all required parameter files. The ESM rather contains a single simulation with CDF 4/4 wavelets, $J_{\max} = 5$, $C_0 = 10$ and $C_\varepsilon = 2.6827 \cdot 10^{-4}$

### Flapping wing simulations: Suzuki test case

Files required to run a simulation:

- Main parameter file: `PARAMS_suzuki_jmax5.ini`
  (or `*_jmax6.ini`, `*_jmax7.ini`)

Each of the coarse, medium and fine resolution simulation is included in the ESM.

### Bumblebee simulations

Files required to run a simulation:

- Main parameter file: `PARAMS_bumblebee_jmax*.ini`
- Wingbeat kinematics file: `bumblebee_new_kinematics.ini`

Each of the coarse, medium and fine resolution simulation is included in the ESM.

### Bumblebee and fractal tree

Files required to run the simulation:

- Main parameter file: `PARAMS_bumblebee_fractaltree.ini`
- Wingbeat kinematics file: `bumblebee_new_kinematics.ini`
- Text file describing the cylinders in the fractal tree: `tree_SPH_branch2.in`

## F Impulsively started cylinder

In this appendix, we consider an impulsively started cylinder at $Re = Du_\infty/\nu = 3000$ and compare with [2–4]. The volume penalization method is used to approximate the Dirichlet boundary condition, and the simulation is two-dimensional. A significant improvement with respect to spectral simulations is that the adaptive framework allows for setting very large domains (for the cylinder simulations, we use here $L = 160$), because the strongest refinement is located near the cylinder. The flow is impulsively started, *i.e.*, the initial velocity is $\underline{u}(\underline{x}, t = 0) = \underline{u}_\infty$ , where $\underline{u}_\infty = (1,0)^T$, and $p(\underline{x}, 0) = 0$. Four levels of resolution are performed, with their key parameters summarized in Fig. 4A. Because of the large computational domain, the coarsest simulation already uses $J_{\max} = 12$ scales.

Figure 4C shows the vorticity field at the final time $t = 3.0$ for the lowest and highest $J_{\max}$. The vorticity is visually very similar to results presented in the literature [2–4], and the two simulations show only little differences. Two main vortices are visible behind the cylinder, and secondary and tertiary vorticity lead to a dipole-like flow pattern near the separation point [4].

The drag force is shown in Fig. 4D together with the reference data and an incompressible spectral simulation with penalization [1] (using $L = 20$, $16384 \times 16384$ modes, $K_\eta = 0.1$). During the impulsive startup phase, present results obtained with ACM differ as expected significantly from the reference solutions. While the force is very large in the beginning, it later decreases and even becomes negative. After some time, depending on $C_0$, the drag however approaches the reference curves. In our applications to insect flight, impulsive start is used because of the lack of suitable initial conditions, rather than an interest in the short-time behavior of the solution. If one is interested in this transient

behavior, the initial speed of sound $C_0$ should be very large and can be decreased after some time. Another possibility is to follow Ohwada [5] and carefully derive appropriate initial conditions. For our focus on flapping flight, this is not required.

As discussed, the impulsive start results in pressure waves propagating in the domain, and they can contribute significantly to the number of blocks used in the simulation. The pattern is similar in all simulations: the initial wave entails an increasing $N_b$, until the wave is absorbed by the sponge (Fig. 4B). Then, the grid density rapidly decreases and then remains practically constant. In all cases, less than 0.1% of the corresponding full grid is used at most, which is also related to the large domain size. Note that pressure waves decay much faster in 3D than in 2D, as explained in the main text.

**References**


[1] T. Engels, D. Kolomenskiy, K. Schneider, and J. Sesterhenn. FluSI: A novel parallel simulation tool for flapping insect flight using a Fourier method with volume penalization. *SIAM J. Sci. Comput.*, 38(5):S3–S24, 2016.

[2] A George, LC Huang, W-P Tang, and YD Wu. Numerical simulation of unsteady incompressible flow ($Re \leq 9500$) on the curvilinear half-staggered mesh. *SIAM J. Sci. Comput.*, 21(6):2331–2351, 2000.

[3] N. Kevlahan and O. Vasilyev. An adaptive wavelet collocation method for fluid–structure interaction at high Reynolds numbers. *SIAM J. Sci. Comput.*, 26(6):1894–1915, 2005.

[4] P. Koumoutsakos and A. Leonard. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.*, 296:1–38, 1995.

[5] T. Ohwada and P. Asinari. Artificial compressibility method revisited: Asymptotic numerical method for incompressible Navier–Stokes equations. *J. Comp. Phys.*, 229:1698–1723, 2010.

[6] C. K.W. Tam and J. C. Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *J. Comput. Phys.*, 107(2):262–281, 1993.

A

| Parameter | level 1 | level 2 | level 3 | level 4 |
|---|---|---|---|---|
| $J_{\max}$ | 12 | 13 | 14 | 15 |
| $C_\varepsilon$ | $10^{-3}$ | $2.50 \cdot 10^{-4}$ | $6.25 \cdot 10^{-5}$ | $1.56 \cdot 10^{-5}$ |
| $C_0$ | 12.5 | 25 | 50 | 100 |
| $C_\eta$ | $1.01 \cdot 10^{-4}$ | $2.51 \cdot 10^{-5}$ | $6.29 \cdot 10^{-6}$ | $1.57 \cdot 10^{-6}$ |
| $C_{\mathrm{sp}}$ | $6.2 \cdot 10^{-2}$ | $3.1 \cdot 10^{-2}$ | $1.5 \cdot 10^{-2}$ | $7.7 \cdot 10^{-3}$ |

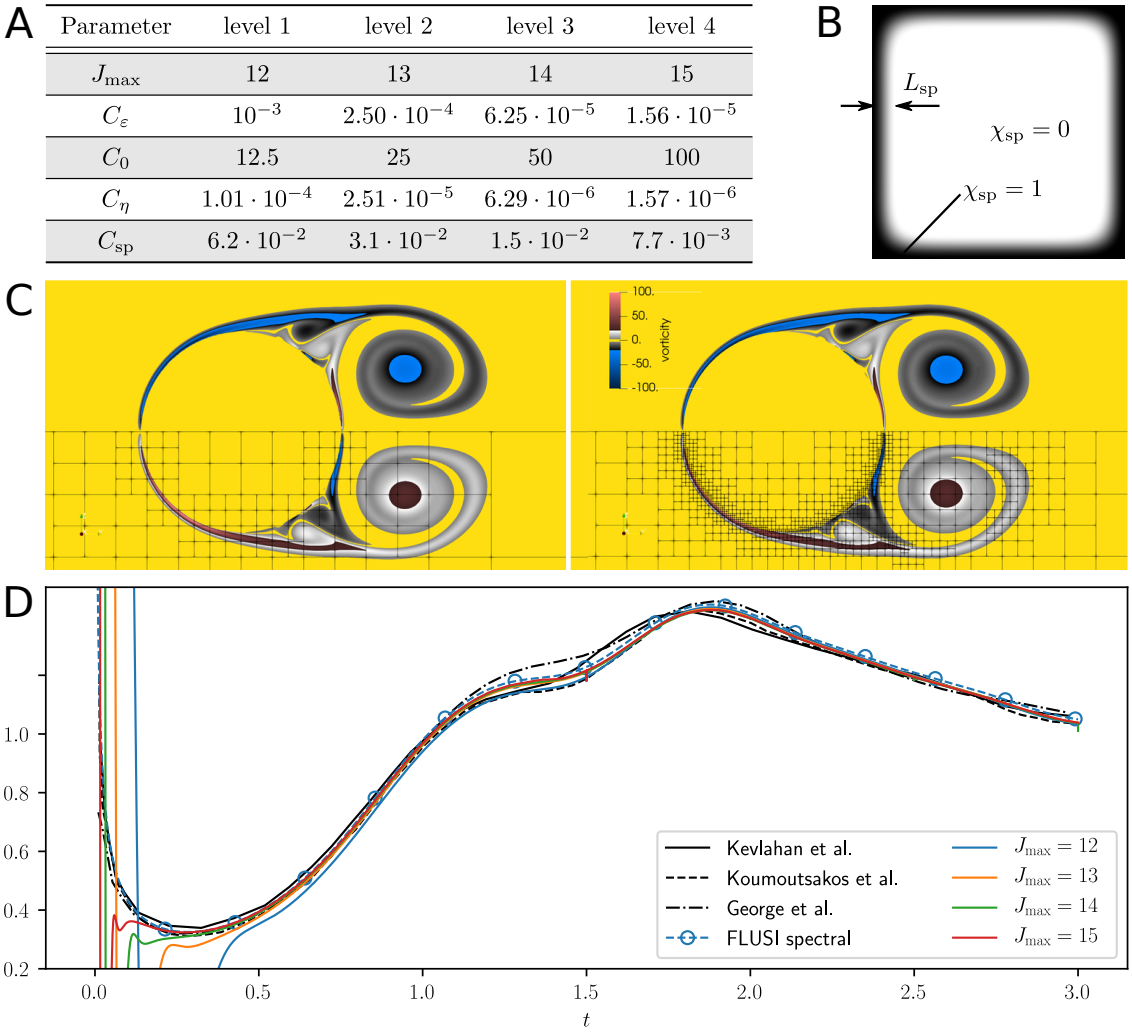


Figure 4: Impulsively started cylinder at $Re$=3000. A: computational parameters for the four levels of resolution. B: mask function $\chi_{\mathrm{sp}}$ for the sponge. The actual cylinder is not visible due to the large domain size. C: vorticity field with the block-based grid superimposed in the lower half, results from the level 1 and 4 computations, respectively. D: drag force as a function of time, compared with results from the literature. All results are obtained with CDF 4/0 wavelets.