A Low-Pass Filter for Linear Forcing in the Open-Source Code OpenFOAM - Implementation and Numerical Performance

24th Results & Review Workshop, HLRS, 7.-8. October 2021

J.A. Denev[1], T. Zirwes[1,2], F. Zhang[2], H. Bockhorn[2]

Motivation

Direct Numerical Simulations or Large-Eddy Simulations often require that turbulence is forced and maintained throughout the simulation. Often, a technique called “linear forcing” is used for this purpose. It consists of adding a body-force term to the Navier-Stokes equations which is linearly proportional to the velocity. By applying a low-pass filter to the velocity field used for forcing, larger integral time scales in the created turbulence field can be achieved [1]. The present work proposes a new three-dimensional, explicit, low-pass Laplacian-filter which is numerically efficient, shows a good scalability and is easy to implement and parallelize.

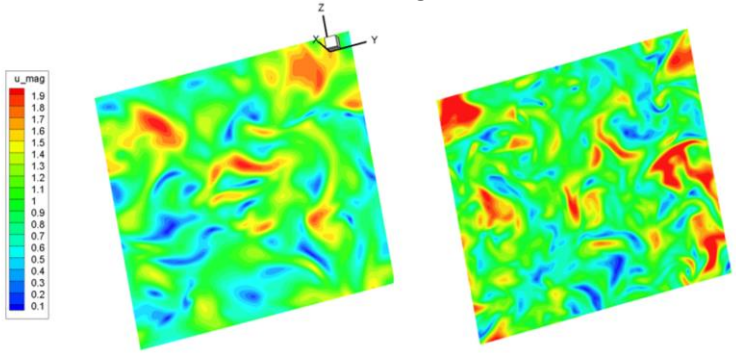
$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{1}{\rho_0}\frac{\partial p}{\partial x_i}+\frac{\partial(\nu 2S_{ij})}{\partial x_j}+f_i$$

$$f_i=C\left(\frac{u_{i,rms,0}^2}{u_{i,rms}^2}\right)\tilde{u}_i \qquad \tilde{\mathbf{u}}=\mathbf{u}+\nabla\cdot\left(B\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right)\right)$$

Example

Figure 1 shows an example of the turbulence field generated by the linear forcing approach with and without using the Laplacian filter. The three-dimensional simulation domain is a cube with 256^3 cells. The velocity field is initialized with isotropic turbulence and forced during the simulation.

***Figure 1**: Turbulent structures in the velocity magnitude at t = 300s on a cutting plane appearing with Laplacian filter (left) and without (right).*



Implementation

The filter is implemented in OpenFOAM. Its formulation follows the diffusion term of the momentum balance equation. The respective OpenFOAM C++ source code is shown on the right.

```
Foam::volVectorField laplaceFilter
(Foam::volVectorField& U, scalar width,
fvMesh& mesh)
{
    coeff    = Foam::pow(mesh.V(),
               2.0/3.0)/width;
    return U + fvc::laplacian(coeff, U)
             + fvc::div(coeff*dev(T
               (fvc::grad(U))));
}
```

Performance

The parallel performance has been measured on the HoreKa [2] supercomputer at the Steinbuch Centre for Computing (Karlsruhe Institute of Technology). The speedup and parallel efficiency are given below for a case with 256^3 cells on up to 1216 CPU cores.

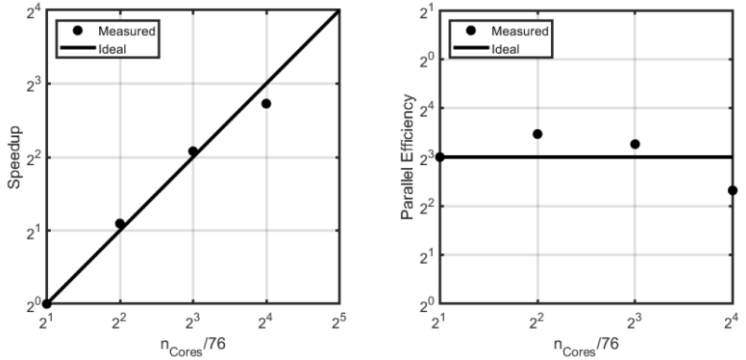


***Figure 2**: Parallel performance.*

References

[1] J. A. Palmore and O. Desjardins, “Linear forcing in numerical simulations of isotropic turbulence: Physical space implementations and convergence properties,” Physical Review Fluids

[2] https://www.scc.kit.edu/dienste/horeka.php

1 Steinbuch Centre for Computing Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1 76344, Eggenstein-Leopoldshafen, Karlsruhe

2 Engler-Bunte-Institute / Chair of Combustion Technology, Karlsruhe Institute of Technology, Engler-Bunte-Ring 7, 76131, Karlsruhe