Oral presentation | Numerical methods
Numerical methods-I
Tue. Jul 16, 2024 2:00 PM - 4:00 PM Room A

[5-A-02] A Fast Coupled Solver for the Steady Navier-Stokes Equations on Rectilinear Grids

*Mark Andrew George[1], Nicholas Williamson[1], Steven Armfield[1] (1. The University of Sydney)

Keywords: incompressible, coupled, solver, collocated, efficient

©Retained by Authors

Twelfth International Conference on Computational Fluid Dynamics (ICCFD12), Kobe, Japan, July 14-19, 2024

A Fast Coupled Solver for the Steady Navier-Stokes Equations on Rectilinear Grids

M. A. George*, N. Williamson* and S. W. Armfield*
Corresponding author: mark.george@sydney.edu.au
*School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney

Abstract: A fully coupled matrix-free finite volume method is developed for solving the incompressible steady-state Navier-Stokes equations on collocated grids. This is achieved by offsetting the momentum equations which are to be solved simultaneously and updating the solution by sweeping planes in 3D and lines in 2D. The method has been implemented within a FAS multigrid solver. For the 3D laminar lid-driven cavity and 3D laminar backwards facing step, up to two orders of magnitude speed-up on a single CPU core was observed, and at least an order of magnitude for all cases compared to the coupled and SIMPLE solvers in ANSYS Fluent, and the SIMPLE scheme of OpenFOAM. Linear scaling of CPU time with problem size is also observed.

Keywords: incompressible, coupled, solver, collocated, efficient.

1 Introduction

Fast running hazard prediction models for the dispersion of airborne contaminants in urban environments need an efficient means for calculating the flow field around complex city environments. A common approach to obtain fast wind fields is through diagnostic models [1] that lack much of the necessary flow physics for accurate prediction of the flow field. This work aims to bridge this gap with the development of fast CFD methods.

The most popular approach for the solution of the steady incompressible Navier-Stokes equations are segregated methods such as the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) [2], due to their simplicity and low memory requirements. However, the greater pressure velocity coupling in coupled solvers results in a more efficient and robust solution process [3, 4]. The greatest draw back of coupled methods is the larger memory requirements, which has become less of an issue due to the reduction in cost of computer hardware [5].

Coupled solvers can be implemented by first assembling the linearised set of momentum and continuity equations into a matrix system, and then solving these equations using a sparse matrix solver and iterating to solve the full non-linear equations. Examples can be found in Darwish et al. [6, 7], Deng et al. [8] and Ammara and Masson [9], all of which demonstrate significant speed up over segregated solvers on collocated and staggered grids. On the other hand, Vanka [10] introduced a matrix-free block implicit approach on staggered grids, which simultaneously updates a pressure node and its neighbouring velocity nodes. The Vanka smoother within a multigrid solver has demonstrated excellent performance compared to segregated algorithms [11, 12]. It is only applicable to staggered finite volume methods however.

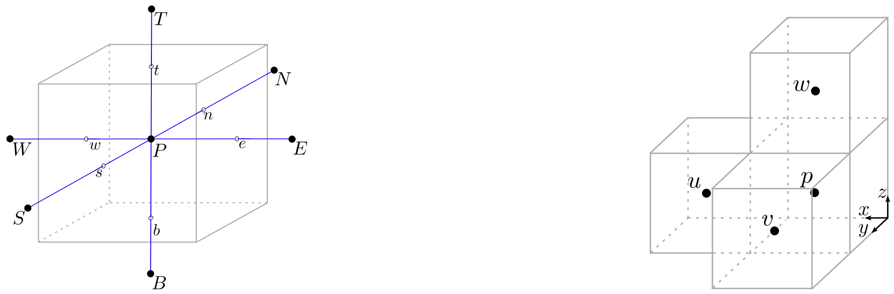
This work presents an asymmetrically coupled, collocated grid analogue to the Vanka smoother. The solution is solved by successively solving planes, which themselves are updated by solving lines. These lines and planes can be oriented to achieve an optimal convergence rate for a particular problem. The smoother is used within a Full Approximation Scheme (FAS) multigrid solver [13] and its relative performance is compared to the SIMPLE scheme and coupled solver in ANSYS Fluent version 2021 R1 [14] and the simpleFoam solver in OpenFOAM v9 [15].

2 Discretisation

The continuous equations solved are the incompressible steady Navier-Stokes equations

$$\nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$



(a) (b)

Figure 1: Finite volume cell at location P with its neighbouring node labels. Lowercase letters indicate cell faces (a) and Offset triad showing momentum equations and cells coupled together implicitly (b). Shown is the case where all momentum equations are offset in the positive coordinate direction, however the offset direction will depend on the sweeping direction.

where $\boldsymbol{u} = (u, v, w)^T$ is the velocity vector, p is the pressure, ρ is the constant fluid mass density, and ν is the kinematic viscosity. These equations are discretised on a collocated rectilinear grid using the finite volume method. Linear interpolation onto the faces, and central differences for derivatives are used for all terms except the advected velocity. The advected velocity is treated implicitly using first order upwind, and through deferred correction, QUICK, or central differencing [16]. The result is a second order accurate scheme. Momentum Weighted Interpolation (MWI) is used in the discretisation of the continuity equation to eliminate pressure oscillation that come as a result of the collocated grid discretisation [17].

With reference to Fig. 1a, the discretised momentum and continuity equations at cell location P can be written as

$$\begin{aligned} a_P^{uu} u_P + \sum_{n(P)} a_F^{uu} u_F + \\ a_W^{up} p_W + a_P^{up} p_P + a_E^{up} p_E = B_P^u, \end{aligned} \tag{3}$$

$$\begin{aligned} a_P^{vv} v_P + \sum_{n(P)} a_F^{vv} v_F + \\ a_S^{vp} p_S + a_P^{vp} p_P + a_N^{vp} p_N = B_P^v, \end{aligned} \tag{4}$$

$$\begin{aligned} a_P^{ww} w_P + \sum_{n(P)} a_F^{ww} w_F + \\ a_B^{wp} p_B + a_P^{wp} p_P + a_T^{wp} p_T = B_P^w. \end{aligned} \tag{5}$$

Where u, v, w are the x, y, and z components of velocity, p is the pressure, $n(P)$ indicates the neighbouring nodes of the cell at location P, and B_P is a source term. The first superscript in the coefficients indicates the equation the coefficient belongs to, and the second one denotes the variable the coefficient multiplies with. The discretised continuity equation is given by

$$\begin{aligned} & a_W^{cu} u_W + a_P^{cu} u_P + a_E^{cu} u_E + \\ & a_S^{cv} v_S + a_P^{cv} v_P + a_N^{cv} v_N + \\ & a_B^{cw} w_B + a_P^{cw} w_P + a_T^{cw} w_T + \\ & a_P^{cp} p_P + \sum_{n(P)} a_F^{cp} p_F = B_P^c. \end{aligned} \tag{6}$$

Due to the MWI the set of nodes $n(P)$ is extended to NN, EE, SS, WW, TT, and BB, which are the cell centre locations two points away from the central point P.

3 The Coupled Smoother

Simply performing a block Gauss-Seidel update of the momentum and continuity equations at each cell location P results in an unstable procedure. The reason for this is that when central differencing is used, the coefficients a_P for the pressure terms in the momentum equations and the velocity terms in the continuity equation are zero on a uniform grid, or very small on non-uniform grids unless the grid growth rate is excessively high. This means that each local block matrix would contain zeros (or very small values) along its diagonal.

This issue is circumvented here by offsetting the momentum equations that are coupled with the continuity equation at each cell as shown in Fig. 1b. The continuity equation is centred at location P, and the pressure is solved for that location. Each momentum equation is offset in the direction corresponding to the sweeping direction in that axis, and the velocity is solved for in that respective cell.

As an example, we consider the case where the x momentum equation is taken at the E cell, the y momentum equation is taken at the N cell, and the z momentum is taken at the T cell. The local block matrix system becomes

$$\begin{pmatrix} (a_P^{uu})_E & 0 & 0 & (a_W^{up})_E \\ 0 & (a_P^{vv})_N & 0 & (a_S^{vp})_N \\ 0 & 0 & (a_P^{ww})_T & (a_B^{wp})_T \\ a_E^{cu} & a_N^{cv} & a_T^{cw} & a_P^{cp} \end{pmatrix} \begin{pmatrix} u_E \\ v_N \\ w_T \\ p_P \end{pmatrix} = \begin{pmatrix} B_E^u - \sum\limits_{n(E)} a_F^{uu} u_F - (a_P^{up})_E p_E - (a_E^{up})_E p_{EE} \\ B_N^v - \sum\limits_{n(N)} a_F^{vv} u_F - (a_P^{vp})_N p_N - (a_N^{vp})_N p_{NN} \\ B_T^w - \sum\limits_{n(T)} a_F^{ww} u_F - (a_P^{wp})_T p_T - (a_T^{wp})_T p_{TT} \\ B_P^c \end{pmatrix}. \tag{7}$$

With this arrangement, the dominant pressure terms in the momentum equations, and velocity terms in the continuity equation appear implicitly. Through elementary row operations, this system can be written into upper triangular form and easily solved for the coupled variables p_P, u_E, v_N, w_T. Sweeping should be done in the direction that the momentum equations are offset in so that all the velocity terms in the continuity are the most recent values from the current iteration.

The Gaussian elimination process above is similar to the derivation of the discrete Poisson equation for pressure in SIMPLE like algorithms where the linearised momentum equations are substituted into the continuity equation. In this case however, only the momentum equation for the particular block being solved is substituted into the continuity equation, while the rest of the terms in the continuity equation are retained in their primitive form.

To solve all flow variables at the boundaries, the offset triad is swept in a line in both a backwards and forward directions, with the offset corresponding momentum equation being flipped. These lines are swept forwards and back to update a plane, and each plane is swept forwards and backward to update the entire domain.

4 FAS Multigrid

FAS multigrid schemes treat the nonlinearity in the equations directly by solving the non-linear problem on the coarse grid levels, as well as the finest level. Details on the FAS scheme used here are given by Henson [13]. To apply the above smoother within a multigrid solver, the discretised Navier-Stokes equations take the form

$$A^h(\boldsymbol{u}^h) = \boldsymbol{f}^h \tag{8}$$

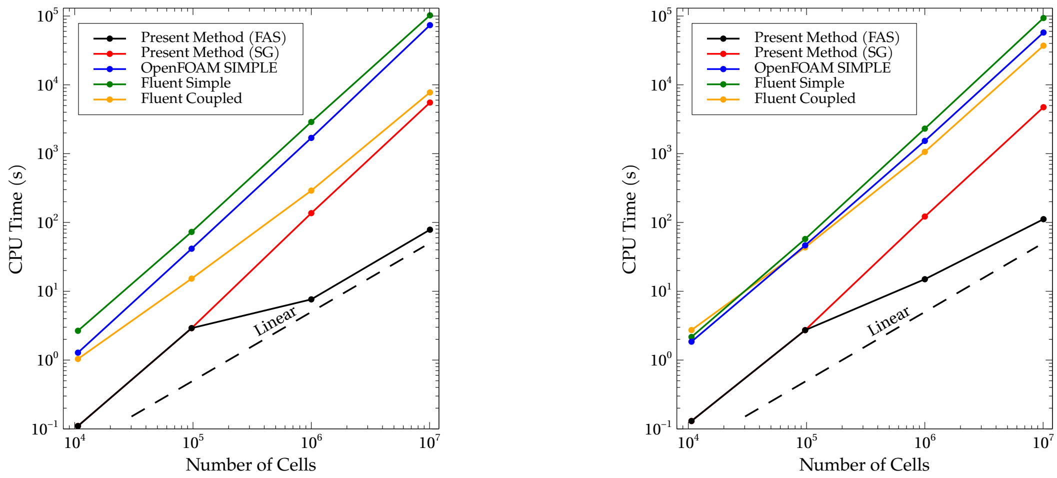
where h indicates the quanitity is on the fine grid. A^h is the non-linear operator for the discrete Navier-Stokes equations, including any constants that arise from boundary conditions, $\boldsymbol{u}^h$ is the solution to the discrete equations, and $\boldsymbol{f}^h$ is a source term which is zero on the finest grids, but in general not zero on coarse grids.

The FAS procedure is applied recursively on each grid level to this system, with the nonlinear problem being smoothed on each grid. Full Multigrid Initialisation (FMG) is used with F-cycles. Coarse grids are formed by agglomerating two cells in each dimension, so in 3D, 8 cells would merge into one. Linear interpolation is used for both prolongation and restriction. In the case where there is not an even number of cells in a particular coordinate direction, the cells on the domain boundary are not agglomerated, but simply transferred to the coarse grid. The side with the largest cells is taken to not be agglomerated to minimise large differences in sizes between adjacent cells.

5 Results and Discussion

The solver is written in C++ and uses the Eigen 3.4.0 Tensor library [18] for multidimensional arrays and storage. The code is currently completely serial. Solver performance in terms of CPU time has been compared to the SIMPLE scheme and coupled solver in ANSYS Fluent version 2021 R1 [14] and the simpleFoam solver in OpenFOAM v9 [15]. The coupled solver in Fluent solves the linearised momentum and continuity equations using an Algebraic Multigrid method.

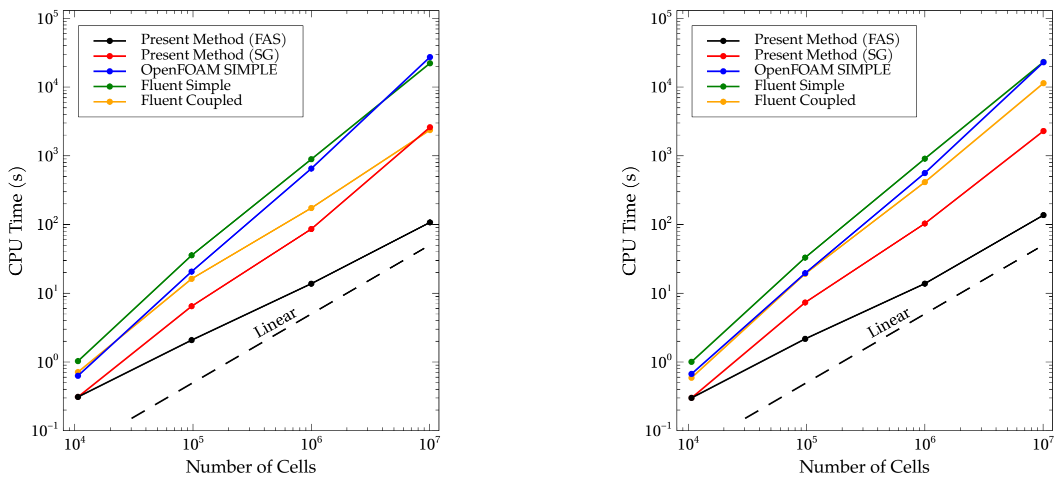
Solver time comparison for the 3D lid driven cavity using first order upwind for the advection scheme, on a non-uniform rectilinear grid for various grid sizes, is given in Fig. 2. All solvers are solved to a residual of 10^{-6} on all equations. The same results are given for the 3D backwards facing step, also on a non-uniform rectilinear grid in Fig. 3. Results are shown for both the solver on a single grid, and using a multigrid FAS scheme.



(a) (b)

Figure 2: Comparison of solver run time on single grid (SG) and multigrid (FAS) with other standard steady solvers for the 3D lid driven cavity at $Re = 200$ (a) and $Re = 1000$ (b).

For some larger grid sizes, particularly for the 3D lid driven cavity, a solution could only be obtained when using a single grid. Shown on each figure is a line indicating the slope of linear scaling of CPU time with problem size. When the mesh is fine enough to allow solution on multiple grids, CPU time scales linearly with problem size.



(a) (b)

Figure 3: Comparison of solver run time on single grid (SG) and multigrid (FAS) with other standard steady solvers for the 3D backwards facing step at $Re = 100$ (a) and $Re = 200$ (b).

The accuracy of the solver is demonstrated by performing order of accuracy tests with various advection schemes. The L_1-norm of the error between a given grid size and a reference fine grid on 320^3 nodes. The coarser grid solution is interpolated onto the finer grid using linear interpolation to calculate the error. The results are given in Fig. 4 and the expected order of accuracy is achieved for all schemes tested.

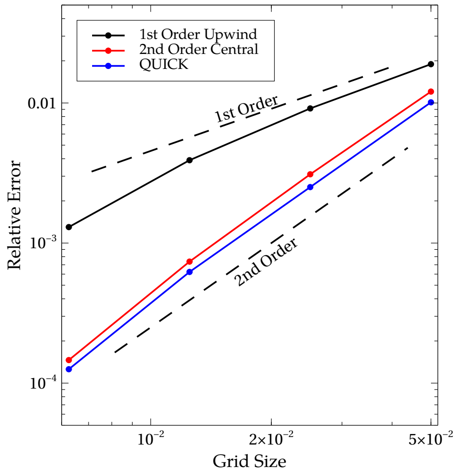
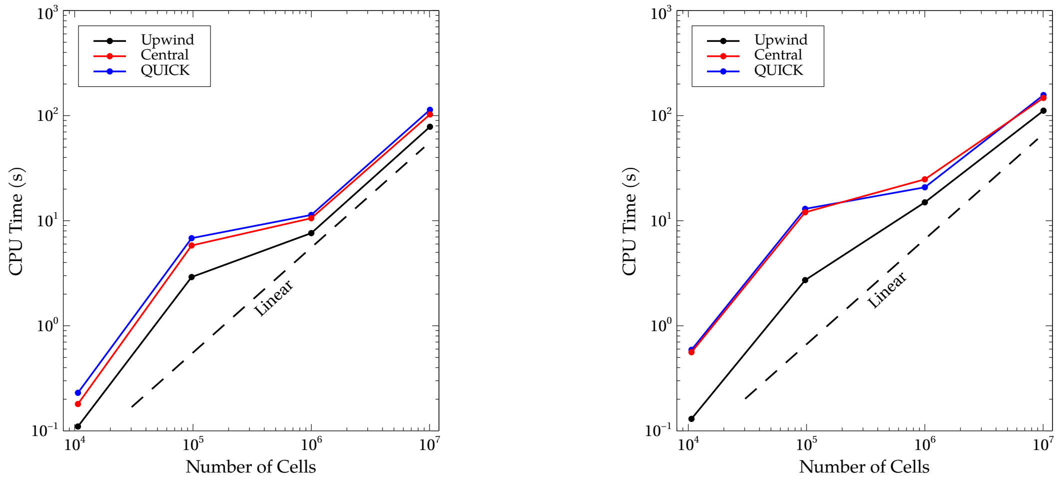


Figure 4: Grid convergence tests for 3D lid driven cavity at $Re = 200$ on a uniform grid. Shown is the L_1-norm of the relative error from a solution calculated at a grid size of 320^3.

Figures 5 and 6 show solver time comparison of the FAS solver between various advection schemes for the same grid sizes tested above. Although solver time increases when second order advection schemes are used, this increase in time is minimal compared to the speed-up achieved compared to the other standard solvers.



(a) (b)

Figure 5: Comparison of FAS solver run time of various advection schemes for the 3D lid driven cavity at $Re = 200$ (a) and $Re = 1000$ (b).

6 Conclusion and Future Work

A fully coupled matrix-free method has been developed that allows the efficient solution of the incompressible Navier-Stokes equations. The method has been used as a smoother in an FAS multigrid scheme. On a single grid, the method is shown to outperform standard implementations of coupled solvers and the SIMPLE scheme in ANSYS Fluent and OpenFoam by up to an order of magnitude for the 3D lid driven cavity and backwards facing step. When used in an FAS multigrid scheme, up to two orders of

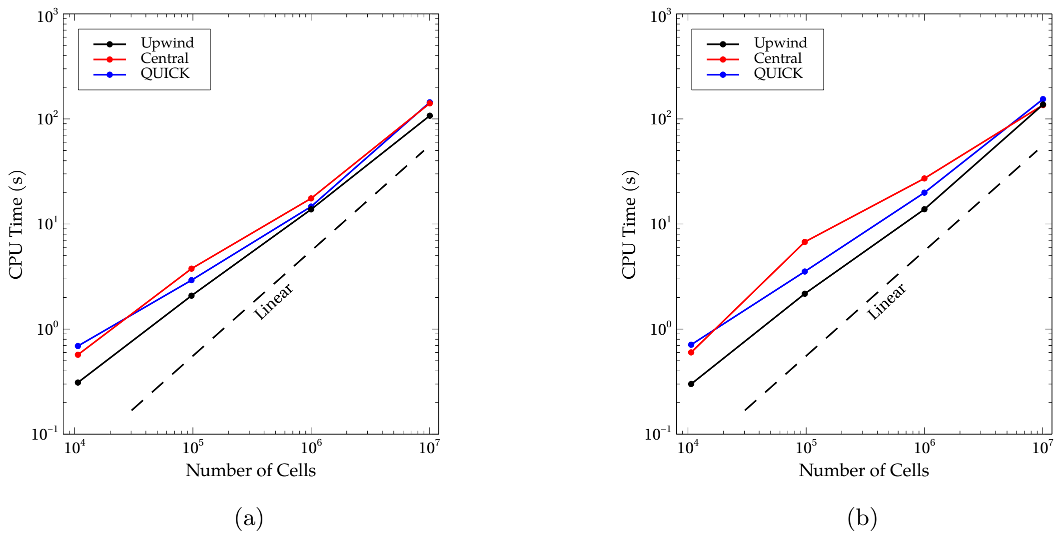


Figure 6: Comparison of FAS solver run time of various advection schemes for the 3D backwards facing step at $Re = 100$ (a) and $Re = 200$ (b).

magnitude speed-up is observed in all cases, and the CPU time of the solvers scales linearly with problem size. Currently, the authors are expanding the capability of the solver to use the immersed boundary method to allow for simulation of flows over more complex geometries.

Acknowledgements

This work was supported and funded by the the Department of Defence and was supported by DMTC Limited (Australia). The authors have prepared this paper in accordance with the intellectual property rights granted by a DMTC Project Agreement.

References

[1] Balwinder Singh, Bradley S. Hansen, Michael J. Brown, and Eric R. Pardyjak. Evaluation of the quic-urb fast response urban wind model for a cubical building array and wide building street canyon. *Environmental Fluid Mechanics*, 8(4):281–312, 2008.

[2] Suhas Patankar. *Numerical heat transfer and fluid flow*. Taylor & Francis, 2018.

[3] H van Santen, D Lathouwers, CR Kleijn, and HEA van den Akker. Influence of segregation on the efficiency of finite volume methods for the incompressible Navier-Stokes equations. *Numerical Developments in CFD. Proceedings of the Fluids Engineering Division of the Americal Society of Mechanical Engineers*, pages 151–158, 1996. Place: New York, USA

[4] Antonio Pascau, Carlos Pérez, and Francisco José Serón. A comparison of segregated and coupled methods for the solution of the incompressible Navier-Stokes equations. *Communications in Numerical Methods in Engineering*, 12(10):617–630, 1996.

[5] Haidong Wang, Hui Wang, Feng Gao, Pengzhi Zhou, and Zhiqiang John Zhai. Literature review on pressure–velocity decoupling algorithms applied to built-environment cfd simulation. *Building and Environment*, 143:671–678, 2018.

[6] M. Darwish, I. Sraj, and F. Moukalled. A Coupled Incompressible Flow Solver on Structured Grids. *Numerical Heat Transfer, Part B: Fundamentals*, 52(4):353–371, August 2007.

[7] M. Darwish, I. Sraj, and F. Moukalled. A coupled finite volume solver for the solution of incompressible flows on unstructured grids. *Journal of Computational Physics*, 228(1):180–201, January 2009.

[8] G. B. Deng, J. Piquet, X. Vasseur, and M. Visonneau. A new fully coupled method for computing turbulent flows. *Computers & Fluids*, 30(4):445–472, May 2001.

[9] Idriss Ammara and Christian Masson. Development of a fully coupled control-volume finite element method for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 44(6):621–644, 2004.

[10] S Pratap Vanka. Block-implicit multigrid solution of navier-stokes equations in primitive variables. *Journal of Computational Physics*, 65(1):138–158, 1986.

[11] Mathias Anselmann and Markus Bause. Efficiency of local vanka smoother geometric multigrid preconditioning for space-time finite element methods to the navier–stokes equations. *PAMM*, 23(1):e202200088, 2023.

[12] Petr Bauer, Vladimir Klement, Tomáš Oberhuber, and Vítězslav Žabka. Implementation of the vanka-type multigrid solver for the finite element approximation of the navier–stokes equations on gpu. *Computer Physics Communications*, 200:50–56, 2016.

[13] Van Emden Henson. Multigrid methods nonlinear problems: an overview. In Charles A. Bouman and Robert L. Stevenson, editors, *Computational Imaging*, volume 5016, pages 36 – 48. International Society for Optics and Photonics, SPIE, 2003.

[14] https://www.ansys.com/products/fluids/ansys-fluent.

[15] https://www.openfoam.org/.

[16] Joel H Ferziger, Milovan Perić, and Robert L Street. *Computational methods for fluid dynamics.* springer, 2019.

[17] Paul Bartholomew, Fabian Denner, Mohd Hazmil Abdol-Azis, Andrew Marquis, and Berend GM Van Wachem. Unified formulation of the momentum-weighted interpolation for collocated variable arrangements. *Journal of Computational Physics*, 375:177–208, 2018.

[18] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.