A high-resolution, unified incompressible solver framework for turbulent flows in OpenFOAM

T.-R. Teschner[1*]

[1] School of Aerospace, Transport and Manufacturing, Cranfield University, Bedford MK43 0AL, UK.
tom.teschner@cranfield.ac.uk

Abstract

This work introduces the Factional-Step, Artificial Compressibility with Pressure Projection (FSAC-PP) method into OpenFOAM, a fast pressure-velocity coupling algorithm for incompressible flows. It is tested for the lid driven cavity problem and it is shown that the pressure Poisson solver speeds up the solution by up to 27.1% compared to the Pimple algorithm available in OpenFOAM. Comparison against the Pressure Projection method from which the FSAC-PP method is derived, are similar favourably.

Key words: *incompressible flow, FSAC-PP, OpenFOAM*

1. Introduction

The absence of a usable equation of state for incompressible flows to close the Navier-Stokes equations has seen a flourish of predictor-corrector-type algorithms to construct the pressure field through various approaches. Chorin introduced a pseudo continuity equation to predict a pressure field, akin to compressible flows, which is devoid of any physical meaning and only recovers physical meaning for steady state flows as the time derivatives vanish. This method is known as the Artificial Compressibility (**AC**) method [1]. Later, Chorin provided a mathematical framework for producing a pressure field through an exact projection of the pressure field, by requiring a divergence-free velocity field once the solution is converged, which results in the unknown velocity vector to vanish and a solution to be calculated through an iterative manner. This method is known as the Fractional-Step Pressure Projection (**FS-PP**) algorithm [2]. Both methods were later succeeded by approximate pressure projection algorithms which have their roots in the work of Patankar and Spalding [3], who introduced the Semi-Implicit Method for Pressure-Linked Equations, or SIMPLE in short. Modifications to the SIMPLE algorithm has provided derivatives such as the SIMPLEC, SIMPLER, PISO and PIMPLE algorithms. These algorithms are the default solvers for incompressible flows available in OpenFOAM. The success of these approximate pressure projection methods notwithstanding, Konozsy and Drikakis [4] proposed a unified incompressible framework, where Chorin's AC [1] and PP [2] algorithms are combined through the Fractional-Step (**FS**) procedure. Here, the pseudo continuity equation works as a preconditioner on the Pressure Poisson solver, thus reducing the computational cost to find the pressure field. The method is known as the Fractional-Step, Artificial Compressibility with Pressure Projection, or **FSAC-FS-PP**, method.

Scale-resolved, turbulent flows are typically restricted in their time-step size due to physical considerations, making higher-order multistep time integration methods a lucrative choice as they provide great temporal accuracy at low implementation effort. OpenFOAM only provides linearised, implicit time integration up to second-order and thus is best suited for time-averaged turbulent simulations. Therefore, this work provides an implementation of the FSAC-PP method of Konozsy and Drikakis into OpenFOAM using a classical fourth-order Runge-Kutta time integration method.

2. Problem description

This section will provide a quick review of the incompressible methods used in this study. Chorin's AC method can be summarised in the following set of equations:

$$\frac{1}{\beta}\frac{\partial p}{\partial \tau} + \nabla \cdot \rho \boldsymbol{u} = 0,$$

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

Here, we have replace the density ρ by the pressure p in the continuity equation's time derivative, which is based on the compressible continuity equation. As the equation of state is not valid here, we have lumped the exact relation between ρ and p into the convergence parameter β. Its exact value is not important, as the AC method is only valid for steady state flows. Once the above system reaches a steady state, the time derivatives in both equations go to zero and so the influence of β on the solution vanishes as well.

Chorin's PP method, on the other hand, removes the issue of finding the pressure by dropping it from the momentum equation and then introducing a second fractional-step where the pressure is calculated, making use of the Helmholtz-Hodge decomposition.

The momentum equation becomes (using a semi-discretised form for the time derivative)

$$\frac{\boldsymbol{u}^{*}-\boldsymbol{u}^{n}}{\Delta t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}=\nu\nabla^{2}\boldsymbol{u},$$

Were $\boldsymbol{u}^{*}$ is an intermediate velocity field and $\boldsymbol{u}^{n}$ the velocity field from the previous or initial solution. The second fraction-step, now consisting of only the semi-discretised time derivative and pressure gradient term, becomes

$$\frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^{*}}{\Delta t}=-\frac{1}{\rho}\nabla p^{n+1}.$$

Using the continuity equation, i.e. $\nabla\cdot\boldsymbol{u}=0$, we impose that for a converged solution the velocity field should be divergence-free and thus have that $\nabla\cdot\boldsymbol{u}^{n+1}=0$. With this assumption, taking the divergence of the second factional-step becomes a pressure Poisson equation of the form

$$\nabla^{2}p^{n+1}=\frac{\rho}{\Delta t}\nabla\cdot\boldsymbol{u}^{*}.$$

Since this form only depends on the intermediate velocity field $\boldsymbol{u}^{*}$, which has been compute beforehand, we can solve the second fractional-step and obtain a new pressure field. With that, we can construct a velocity correction equation using the non-divergence-free fractional-step formulation to solve for the unknown velocity field $\boldsymbol{u}^{n+1}$ as

$$\boldsymbol{u}^{n+1}=\boldsymbol{u}^{*}-\frac{\Delta t}{\rho}\nabla p^{n+1}.$$

In the FSAC-PP method, we combine the pseudo continuity equation of the AC method with the momentum and pressure Poisson equation of the PP method. This can be summarised as follows:

1. Solve for intermediate velocity $\boldsymbol{u}^{*}$
$$\frac{\boldsymbol{u}^{*}-\boldsymbol{u}^{n}}{\Delta t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}=\nu\nabla^{2}\boldsymbol{u}$$
2. Solve for an initial pressure field (preconditioning the pressure for the Poisson equation)
$$\frac{1}{\beta}\frac{\partial p}{\partial\tau}+\nabla\cdot\rho\boldsymbol{u}=0$$
3. Using the initial pressure field, obtain the pressure at time level $n+1$ through the Poisson equation
$$\nabla^{2}p^{n+1}=\frac{\rho}{\Delta t}\nabla\cdot\boldsymbol{u}^{*}$$
4. Update the velocity field based on the new pressure as
$$\boldsymbol{u}^{n+1}=\boldsymbol{u}^{*}-\frac{\Delta t}{\rho}\nabla p^{n+1}.$$

The time savings in the FSAC-PP method over the AC and FS-PP methods are twofold; first, the pressure Poisson solver requires a substantial number of iterations to converge. Providing an initial pressure field based on some form of physical considerations provides a faster solution finding procedure. Second, the AC method is hyperbolic in nature and prone to oscillations during the iterative procedure. Coupling the oscillatory pseudo continuity equation with the elliptic Poisson equation provides a mathematical mechanism of damping these oscillations and thus speeding up the iterative procedure. Konozsy and Drikakis [4] showed, in-fact, that the Poisson equation does not need to be solved exactly and further time savings are achieved by limiting the number of iterations it can use to find the pressure.

The discretisation of the time derivative is carried out with an explicit 4th-order Runge time integration method and the CFL number is limited to 0.9 during all calculations. The spatial derivatives of the non-linear term are discretised using a second-order upwind formulation.

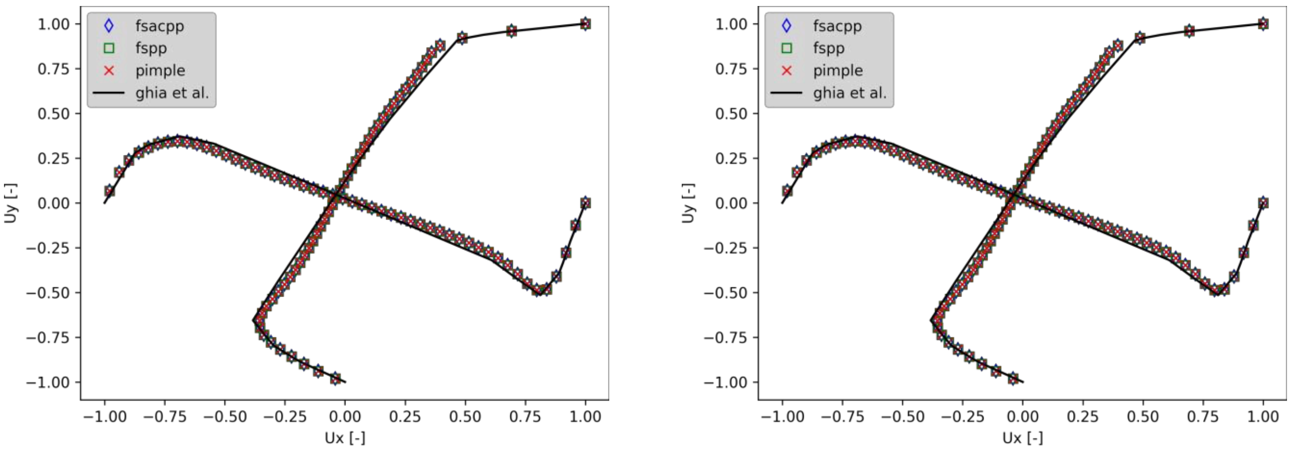

Figure 1: Velocity profiles for the y component of the velocity vector along the horizontal x direction and x component of the velocity vector along the vertical y direction, normalised between -1 and 1. Results are shown for the FSAC-PP, FS-PP and Pimple algorithm and are compared against data provided by Ghia et al. [5] at a Reynolds number of 1000. The figure on left shows the results when the implicit pressure solver is allowed to converge fully, while the plot on the right shows the case where the pressure solver is only allowed to solve for 50 iterations.

3. Numerical results

In the following, the FSAC-PP method is compared against the FS-PP method as well as the Pimple algorithm implemented in OpenFOAM, which is essentially the PISO algorithm with an additional outer corrector step so that larger CFL numbers can be achieved in the semi-implicit momentum equations. The FSAC-PP and FS-PP method do not require an additional outer iteration to stabilise the solution at higher CFL numbers as the velocity and pressure are entirely decoupled through the factional step procedure, due to the dropped pressure gradient in the momentum equation (which is retained in the pimple algorithm).

In Figure 1, the velocity profiles of the x and y component of the velocity vector are shown across the vertical and horizontal symmetry line, respectively, at a Reynolds number of 1000 for the three solver tested in this investigation. No turbulence model is used, which is in accordance with the reference data by Ghia et al. [5] and the match for all solvers is close to the reference data provided. In-fact, results between the FSAC-PP, FS-PP and Pimple algorithm are indistinguishable. In Figure 2, the same velocity profiles are shown, this time at a Reynolds number of 10000. Since no turbulence model is used, the grid size had to be increased from 128^2 (which was used for the Reynolds number of 1000 simulation) to 256^2, in order to provide a better resolved domain for which the simulation is still stable, which is also in accordance with the reference simulation.

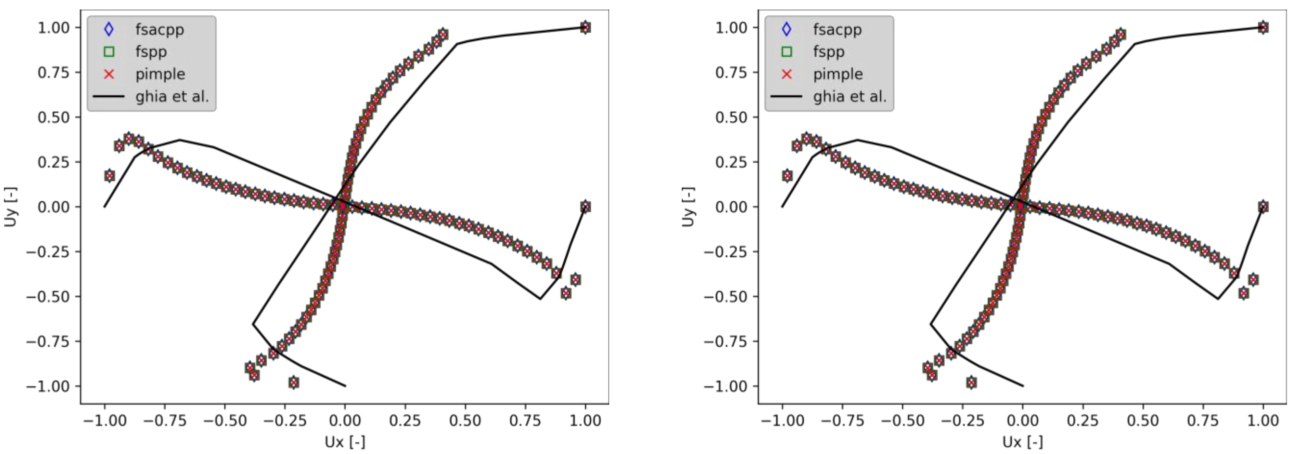

Figure 2: Velocity profiles for the y component of the velocity vector along the horizontal x direction and x component of the velocity vector along the vertical y direction, normalised between -1 and 1. Results are shown for the FSAC-PP, FS-PP and Pimple algorithm and are compared against data provided by Ghia et al. [5] at a Reynolds number of 10000. The figure on left shows the results when the implicit pressure solver is allowed to converge fully, while the plot on the right shows the case where the pressure solver is only allowed to solve for 50 iterations.

While there is good agreement for the Reynolds number of 1000 simulation, significant differences are introduced at larger Reynolds number. The main contributing factor in this case is the fact the simulations at a Reynolds number of 10000 had problems to converge, possibly due to the enhanced grid size for which smaller scales were resolved in the flow. Alternative strategies to increase dissipation would be to either use a lower-order scheme (e.g. first-order upwind) or RANS turbulence model, two strategies not explored in this preliminary investigation. However, we can see that the results are again consistent for all solvers and virtually no difference between the obtained results exists.

In Table 1, a comparison is done between the different solvers, looking at how many iterations, on average, are spend on solving the implicit pressure equation. At a Reynolds number of 1000, the FSAC-PP method uses 16.1% and 16.7% less iterations per pressure Poisson equation evaluation, compared to the FS-PP and Pimple algorithm, respectively. At the higher Reynolds number of 10000, the FSAC-PP requires 26.0% and 27.1% fewer iterations compared to the FS-PP and Pimple algorithm, respectively. Furthermore, there may be additional outer iterations the Pimple algorithm has to solve, which will introduce further computational savings at higher CFL numbers.

Table 1: Comparison of average number of iterations within the pressure solver for different solvers at Reynolds numbers of 1000 and 10000.

	FSAC-PP	FS-PP	Pimple
Re = 1000	181.8	216.7	218.2
Re = 10000	319.3	431.3	438.0

4. Conclusions

In this preliminary investigation, the Fractional-Step, Artificial Compressibility with Pressure Projection (FSAC-PP) method has been implemented into OpenFOAM using a fourth-order, explicit Runge-Kutta time integration strategy. It has been applied to the lid driven cavity test case at Reynolds numbers of 1000 and 10000. Results show, compared against the classical Pressure Projection method of Chorin (FS-PP) from which the FSAC-PP method is derived, as well as the Pimple algorithm which in turn is based on the PISO algorithm with an extended outer loop, that the FSAC-PP method produces the same results and thus for the presented test case is validated. Furthermore, the computational savings are substantial and as the physical complexity increases (i.e. turbulence becomes more important), computational savings are larger and up to 27.1%. The FSAC-PP method is a novel way to couple pressure and velocity for incompressible flows and shows potential to significantly improve performance for turbulence calculation which is in line with findings in the literature [4]. Once the method is extended for unsteady flows through a dual-time stepping integration procedure in OpenFOAM, it needs to be tested for unsteady, scale-resolved flows to test its suitability and computational savings for real and challenging flow scenarios.

References

[1] A. Chroin, (1967), A Numerical Method for Solving Incompressible Viscous Flow Problems, Journal of Computational Physics, 2, 12 – 26.

[2] A. Chorin, (1968), Numerical solution of the Navier-Stokes equations, Mathematics of Computations, 22(104), 745 – 762.

[3] S. Patankar and D. Spalding, (1972), A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, International Journal of Heat and Mass Transfer, 15(10), 1787 – 1806.

[4] L. Konozsy and D. Drikakis, (2014), A Unified Fractional-Step, Artificial Compressibility and Pressure-Projection Formulation for Solving the Incompressible Navier-Stokes Equations, Communications in Computational Physics, 16(5), 1135 – 1180.

[5] U. Ghia, K. N. Ghia andC. T. Shin, (1982), High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, Journal of computational Physics, 48, 387 – 411.

A high-resolution, unified incompressible solver framework for turbulent flows in OpenFOAM

Teschner, Tom-Robin

2023-04-21

Teschner T-R. (2022) A high-resolution, unified incompressible solver framework for turbulent flows in OpenFOAM. Presented at: UKACM 2023 Conference (UK Association for Computational Mechanics), 19-21 April 2023, Warwick University, Coventry, UK

https://dspace.lib.cranfield.ac.uk/handle/1826/19568