ParCFD2024
35th International Conference on Parallel Computational Fluid Dynamics
Sep. 02-04, 2024, Bonn, Germany

A Parallel-In-Time Spectral Deferred Corrections Method for the Incompressible Navier-Stokes Equations

Abdelouahed Ouardghi[a,*], Robert Speck[a]

[a]Forschungszentrum Jülich GmbH, Jülich Supercomputing Centre, Wilhelm-Johnen-Straße, 52428 Jülich, Germany

ARTICLE INFO[†]

Keywords:
Parallel-in-Time;
Spectral Deferred Corrections;
Incompressible Navier-Stokes Equations

ABSTRACT

In this work, spectral deferred corrections (SDC) methods are considered as parallel-in-time integrators for the solution of the unsteady incompressible Navier-Stokes equations. These temporal methods are coupled with a high-order finite element spatial approximation. The goal of this work is to illustrate and analyze the properties of the parallel-in-time method through numerical experiments, including flow past a cylinder (using the standard DFG 2D-3 benchmark) which is selected as an example of unsteady flow.

1. Introduction

The numerical simulation of the Navier-Stokes equations (NSE) in the primitive formulation for incompressible flow is an active research topic. Yet, the challenge of accurately resolving these equations in both space and time necessitates the deployment of advanced high-performance computing systems. While spatial parallelization can reduce the runtime per time step, temporal integration of time-sensitive applications often requires a large number of time steps. Therefore, for further speedup, parallel-in-time integrators are required for more parallelism in the temporal domain. One of the time integrators that can be used to enable efficient parallel-in-time integration is the spectral deferred corrections methods introduced in 2000 by Dutt et al. [1], as a more stable variant of the classical deferred corrections approach for solving ordinary differential equations (ODEs). SDC are an iterative approach for the numerical solution of ordinary differential equations. It works by refining the numerical solution for an initial value problem by performing a series of correction sweeps using a low-order time-stepping method, and can be interpreted as a preconditioned Picard iteration to solve a fully implicit collocation problem. SDC has been widely used for various initial value problems, using explicit, implicit or implicit-explicit Euler and other low-order methods as preconditioner [2, 3, 4, 5]. Moreover, two strategies to enable parallelization across the method for spectral deferred corrections are presented by R. Speck [6]. In this work the SDC methods implemented in pySDC [7] are combined with the finite element method (FEM), as facilitated by the frameworks FEniCS providing a powerful tool for computational fluid dynamics applications. FEM excels at accurately representing complex geometries and boundary conditions, making it an ideal choice for spatial discretization. This coupling results in a robust and accurate numerical solution to the Navier-Stokes equations, benefiting from the parallel computing capabilities of both methods.

2. SDC for Navier-Stokes

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with the boundary $\partial\Omega$ and $[0, T]$ is a time interval. The governing equations consist of the incompressible Navier-Stokes equations

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} &= -\mathbf{u}\cdot\nabla\mathbf{u} - \nabla p + \nu\nabla^2\mathbf{u} + \mathbf{g}, \\ \nabla\cdot\mathbf{u} &= 0, \end{cases} \tag{1}$$

where $\mathbf{u}$ is the velocity vector, p is the pressure, ν is the kinematic viscosity, and $\mathbf{g}$ represents external forces. To solve these equations, we begin by subdividing the time interval into sub-intervals $[t_n, t_{n+1}]$. Then, using Chorin's projection scheme, we first compute an intermediate velocity field $\mathbf{u}^*$ with the momentum equation

$$\begin{cases} \dfrac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} &= -\mathbf{u}^n\cdot\nabla\mathbf{u}^n + \nu\nabla^2\mathbf{u}^* + \mathbf{g}^{n+1}, \\ u^* &= 0 \qquad \text{on } \partial\Omega. \end{cases} \tag{2}$$

In the next step, the intermediate velocity is projected to the space of divergence free vector fields to get the next update of velocity and pressure

$$\mathbf{u}^{n+1} - \mathbf{u}^* = -\Delta t\nabla p^{n+1}. \tag{3}$$

[†]This paper is part of the ParCFD 2024 Proceedings. A recording of the presentation is available on YouTube. The DOI of this document is 10.34734/FZJ-2025-02513 and of the Proceedings 10.34734/FZJ-2025-02175.

*Corresponding author
a.ouardghi@fz-juelich.de (A. Ouardghi); r.speck@fz-juelich.de (R. Speck)
ORCID(s): 0000-0003-3471-7424 (A. Ouardghi); 0000-0002-3879-1210 (R. Speck)

Taking the divergence and requiring the incompressibility condition $\nabla \cdot \mathbf{u}^{n+1} = 0$, we obtain the equation

$$\Delta t \nabla^2 p^{n+1} = \nabla \cdot \mathbf{u}^*, \tag{4}$$

which is a Poisson problem for the pressure p^{n+1}. Finally, the velocity $\mathbf{u}^{n+1}$ can be corrected using Eq. (3).

2.1. Spatial discretization

To tackle the space discretization, we utilize the mixed finite element approach. We will focus on Eq. (2) as it is the only one requiring to be solved using the SDC methods. The matrix form of a weak formulation for Eq. (2) can be written as

$$[\mathtt{M}]\frac{d\mathbf{u}}{dt} = \mathbf{f}(t,\mathbf{u}) = \mathbf{f}_I(t,\mathbf{u}) + \mathbf{f}_E(t,\mathbf{u}), \tag{5}$$

where $[\mathtt{M}]$ is the finite element mass matrix, $\mathbf{f}_I(t,\mathbf{u}) = -[\mathtt{K}]\mathbf{u}+[\mathtt{M}]\mathbf{g}(t)$ is the implicit part of $\mathbf{f}$ with $[\mathtt{K}]$ is the stiffness matrix, and $\mathbf{f}_E(t,\mathbf{u}) = -[\mathtt{C}(\mathbf{u})]\mathbf{u}$ is the non-linear advection part that should be treated explicitly.

2.2. Parallel-in-time SDC

We write the Picard form of the problem in Eq. (5) as follows

$$\begin{aligned}[\mathtt{M}]\mathbf{u}(t) &= [\mathtt{M}]\mathbf{u}_0 + \int_{t_n}^{t} \mathbf{f}(s,\mathbf{u}(s))ds, \\ & t_n \le t \le t_{n+1},\end{aligned} \tag{6}$$

where $\mathbf{u}_0 = \mathbf{u}(t_n)$. The integral in Eq. (6) can be approximated using a spectral quadrature rule. For this reason we discretize the time interval $[t_n, t_{n+1}]$ using M quadrature nodes such that $t_n < \tau_1 < \cdots < \tau_M = t_{n+1}$. Thus the Eq. (6) can be written as

$$\begin{aligned}[\mathtt{M}]\mathbf{u}_m &= [\mathtt{M}]\mathbf{u}_0 + \Delta t \sum_{j=1}^{M} q_{m,j}\mathbf{f}(\tau_j,\mathbf{u}_j), \\ m &= 1,\dots,M,\end{aligned} \tag{7}$$

where $\mathbf{u}_m \approx \mathbf{u}(\tau_m)$ and $q_{m,j} = \int_{t_0}^{\tau_m} L_j(s)ds, \quad j = 1,\cdots,M$ are the set of quadrature weights with L_j is the Lagrange polynomial defined on the quadrature node $\{\tau_j\}$. Next, we combine these M equations into one system of linear or non-linear equations which is called the collocation problem

$$(\mathcal{M} - \Delta t \mathbf{Q}\mathbf{F})(\vec{\mathbf{u}}) = \mathcal{M}\vec{\mathbf{u}}_0, \tag{8}$$

where $\mathcal{M} = [\mathtt{M}] \otimes \mathbf{I}_M$, $\vec{\mathbf{u}} = (\mathbf{u}_1,\cdots,\mathbf{u}_M)^T$, $\vec{\mathbf{u}}_0 = (\mathbf{u}_0,\cdots,\mathbf{u}_0)^T$, $\mathbf{Q} = (q_{ij})_{i,j}$ is the matrix gathering the quadrature weights and the vector function $\mathbf{F}$ is given by $\mathbf{F}(\vec{\mathbf{u}}) = \big(\mathbf{f}(\mathbf{u}_1),\cdots,\mathbf{f}(\mathbf{u}_M)\big)$. Moreover, if $d \ge 2$ then $\mathbf{Q}$ and $\mathcal{M}$ must be replaced by $\mathbf{Q}\otimes\mathbf{I}_d$ and $\mathcal{M}\otimes\mathbf{I}_d$, respectively. It should be noted that this system of equation is equivalent to the fully implicit Runge-Kutta method with $\mathbf{Q}$ being the Butcher tableau. Generally, this system is dense and requires an iterative solution. SDC can be presented as precondtionated Picard iteration for the collocation problem in Eq. (8) as

$$\begin{aligned}\left(\mathcal{M} - \Delta t \mathbf{Q}_\Delta \mathbf{F}\right)(\vec{\mathbf{u}}^{k+1}) &= \mathcal{M}\vec{\mathbf{u}}_0 \\ &\quad + \Delta t\left(\mathbf{Q} - \mathbf{Q}_\Delta\right)\mathbf{F}(\vec{\mathbf{u}}^k).\end{aligned} \tag{9}$$

The ODE (5) contains both stiff and non-stiff components. Therefore, semi-implicit SDC (SISDC) is considered for the solution of Eq. (5). Consequently, the iteration in Eq. (9) becomes

$$\begin{aligned}&\left(\mathcal{M} - \Delta t \mathbf{Q}_{\Delta,I}\mathbf{F}_I - \Delta t \mathbf{Q}_{\Delta,E}\mathbf{F}_E\right)(\vec{\mathbf{u}}^{k+1}) \\ &= \mathcal{M}\vec{\mathbf{u}}_0 + \Delta t\left(\mathbf{Q} - \mathbf{Q}_{\Delta,I}\right)\mathbf{F}_I(\vec{\mathbf{u}}^k) \\ &\quad + \Delta t\left(\mathbf{Q} - \mathbf{Q}_{\Delta,E}\right)\mathbf{F}_E(\vec{\mathbf{u}}^k),\end{aligned} \tag{10}$$

where $\mathbf{Q}_{\Delta,I}$ and $\mathbf{Q}_{\Delta,E}$ are lower and strictly lower triangular matrices It has been shown in [6] that SDC can be parallelize over the quadrature nodes using diagonal preconditioners (i.e., diagonal matrix $\mathbf{Q}_\Delta$). In this work the following diagonal preconditioners are considered: $\mathbf{Q}_{\Delta,I} = \mathbf{Q}_\Delta^{IEpar}$, $\mathbf{Q}_\Delta^{MIN}$, $\mathbf{Q}_\Delta^{MIN\_SR\_S}$ and $\mathbf{Q}_\Delta^{MIN\_SR\_NS}$, (see [6, 8] for more details). Given that the matrix $\mathbf{Q}_{\Delta,E}$ must be lower triangular, the null matrix $\mathbf{Q}_{\Delta,E} = \mathbf{0}$ is considered for the explicit part.

3. Numerical results

In this section, we will consider the benchmark of flow past a cylinder. The geometry and parameters are taken from the DFG 2D-3 benchmark in FeatFlow, see Fig. 1. The inflow velocity profile is

$$\mathbf{u}_{in} = \left(\frac{4U\,y(0.41-y)}{0.41^2}, 0\right) \tag{11}$$

$$U = U(t) = 1.5\sin\left(\frac{\pi t}{8}\right) \tag{12}$$

and the kinematic viscosity is given by $\nu = 0.001$.

In Fig. 2 we show the lift coefficients in the unsteady Navier-Stokes equations subject to the time-dependent inflow profile $\mathbf{u}_{in}$. These results are computed using 2, 3, and 5 Gauß–Radau nodes with two different time steps $\Delta t = 1/1{,}600$ and $\Delta t = 1/1{,}000$. The data provided by FEATFLOW is also included in the figure as a reference solution. Based on Fig. 2, it is clear that the different SDC methods solves accurately the Navier-Stokes equations. It should be mentioned that there are slight differences between the results obtained using $\Delta t = 1/1{,}600$ and $\Delta t = 1/1{,}000$ because of the CFL constraint required by the semi-implicit scheme. Moreover, the average number of iterations for five different choices of $\mathbf{Q}_{\Delta,I}$ after 50 time steps at four different time points throughout the simulation are displayed in Fig. 3. These choices are the LU trick ($\mathbf{Q}_\Delta^{LU}$) as reference as well

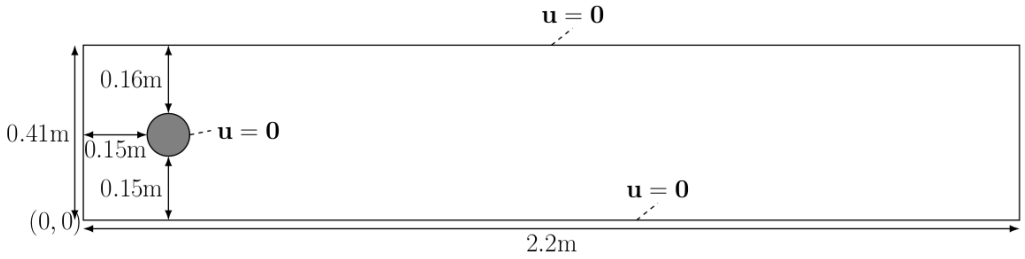

Figure 1: Computational geometry for the DFG 2D-3 benchmark.

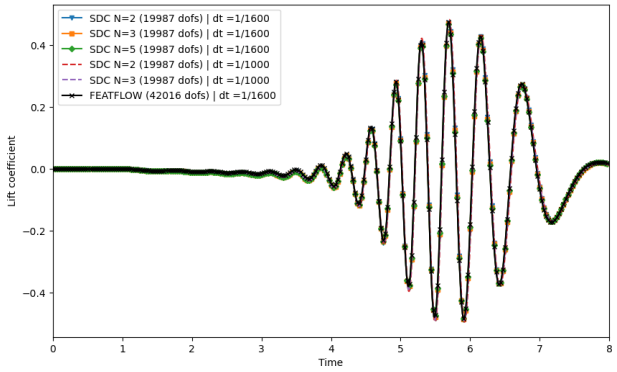

Figure 2: Lift coefficient using different SDC methods and different time steps.

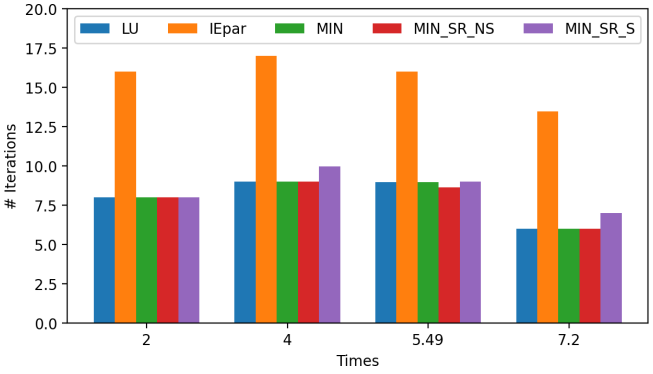

Figure 3: Average number of iterations needed by different choices of preconditioner.

as the four diagonal matrices listed in the previous section. From this figure, it can be clearly seen that the standard option is the best choice for all tests. It can be also seen that the $\mathbf{Q}_\Delta^{IEpar}$ requires at least double the number of iterations to converge, while the minimization-based preconditioners $\mathbf{Q}_\Delta^{MIN}$ and $\mathbf{Q}_\Delta^{MIN\_SR\_NS}$ seem to be reliable and converge about as fast as the standard choice $\mathbf{Q}_\Delta^{LU}$ for the considered problem. However, the choice $\mathbf{Q}_\Delta^{MIN\_SR\_S}$ requires slightly more iterations to converge.

4. Conclusion

Combining SDC with FEM accurately solves the Navier-Stokes equations. Diagonal preconditioners enable parallel SDC, and with minimization-based preconditioners, diagonal SDC converges as fast as standard SDC. The future goal is to use a fully implicit monolithic scheme for the NSE to avoid the CFL limitations of semi-implicit schemes.

References

[1] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT Numerical Mathematics 40 (2000) 241–266. doi:10.1023/A:1022338906936.

[2] D. Ruprecht, R. Speck, Spectral deferred corrections with fast-wave slow-wave splitting, SIAM Journal on Scientific Computing 38 (4) (2016) A2535–A2557. doi:10.1137/16M1060078.

[3] A. C. Hansen, J. Strain, On the order of deferred correction, Applied Numerical Mathematics 61 (8) (2011) 961–973. doi:10.1016/j.apnum.2011.04.001.

[4] I. Akramov, S. Götschel, M. Minion, D. Ruprecht, R. Speck, Spectral deferred correction methods for second-order problems, SIAM Journal on Scientific Computing 46 (3) (2024) A1690–A1713. doi:10.1137/23M1592596.

[5] L. Micalizzi, D. Torlo, A new efficient explicit deferred correction framework: Analysis and applications to hyperbolic pdes and adaptivity, Communications on Applied Mathematics and Computation (2023). doi:10.1007/s42967-023-00294-6.

[6] R. Speck, Parallelizing spectral deferred corrections across the method, Computing and Visualization in Science 19 (2018) 75–83. doi:10.1007/s00791-018-0298-x.

[7] R. Speck, Algorithm 997: pySDC - prototyping spectral deferred corrections, ACM Transactions on Mathematical Software (TOMS) 45 (3) (2019). doi:10.1145/3310410.

[8] G. Čaklović, T. Lunet, S. Götschel, D. Ruprecht, Improving efficiency of parallel across the method spectral deferred corrections (2024). arXiv:2403.18641.