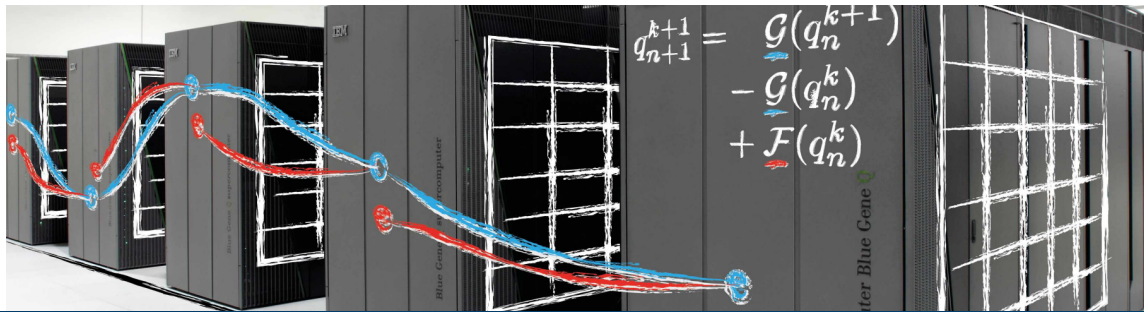




SPACE-TIME PARALLELISM USING SPECTRAL DEFERRED CORRECTIONS AND FINITE ELEMENTS FOR INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

July 8th, 2025 | [Abdelouahed Ouarghi](#), Robert Speck | Jülich Supercomputing Centre, Forschungszentrum Jülich

Outline

- 1 Collocation problem
- 2 Spectral deferred correction method (SDC)
- 3 Software: pySDC and FEniCS
- 4 IMEX SDC (SISDC) using projection-based splitting method
- 5 Fully implicit SDC using the monolithic approach
- 6 Summary

The Collocation Problem

Consider the classical initial value problem

$$\frac{du}{dt} = f(t, u), \quad u \in \mathbb{R}^d, \quad t \in [t_0, t_0 + \Delta t].$$

Write the Picard form of the problem.

$$u(t) = u_0 + \int_{t_0}^t f(s, u(s)) ds, \quad t_0 \leq t \leq t_0 + \Delta t$$

Discretize $[t_0, t_0 + \Delta t]$ using M quadrature nodes, then approximate f using Lagrange polynomials $l_j(t)$:

$$f(t, u) \approx \sum_{j=1}^M f(\tau_j, u(\tau_j)) l_j(t), \quad t_0 \leq \tau_1 \leq \dots \leq \tau_m \leq \dots \leq \tau_M \leq t_0 + \Delta t$$

Approximate the integral on given quadrature nodes

$$u(\tau_m) \approx u_m = u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_j, u_j), \quad q_{m,j} = \int_{t_0}^{\tau_m} l_j(s) ds.$$

The Collocation Problem

Consider the classical initial value problem

$$\frac{du}{dt} = f(t, u), \quad u \in \mathbb{R}^d, \quad t \in [t_0, t_0 + \Delta t].$$

Write the Picard form of the problem.

$$u(t) = u_0 + \int_{t_0}^t f(s, u(s)) ds, \quad t_0 \leq t \leq t_0 + \Delta t$$

Discretize $[t_0, t_0 + \Delta t]$ using M quadrature nodes, then approximate f using Lagrange polynomials $l_j(t)$:

$$f(t, u) \approx \sum_{j=1}^M f(\tau_j, u(\tau_j)) l_j(t), \quad t_0 \leq \tau_1 \leq \dots \leq \tau_m \leq \dots \leq \tau_M \leq t_0 + \Delta t$$

Approximate the integral on given quadrature nodes

$$\vec{u} = \vec{u}_0 + \Delta t QF(\vec{u})$$

Spectral Deferred Corrections

- Consider the fully implicit collocation method

$$(I - \Delta t Q F)(\vec{u}) = \vec{u}_0$$

- Standard Picard iteration can be written as

$$\vec{u}^{k+1} = \vec{u}^k + \underbrace{(\vec{u}_0 - (I - \Delta t Q F)(\vec{u}^k))}_{\text{residual } \vec{r}^k}$$

- Improve it with a preconditioned iteration

$$P\vec{u}^{k+1} = P\vec{u}^k + [\vec{u}_0 - (I - \Delta t Q F)(\vec{u}^k)]$$

- Use the preconditioner $P = (I - \Delta t Q_\Delta F)$ with $Q_\Delta \approx Q$ is a lower triangular approximation of Q

$$(I - \Delta t Q_\Delta F)(\vec{u}^{k+1}) = (I - \Delta t Q_\Delta F)(\vec{u}^k) + (\vec{u}_0 - (I - \Delta t Q F)(\vec{u}^k))$$

Spectral Deferred Corrections

- Consider the fully implicit collocation method

$$(I - \Delta t Q F)(\vec{u}) = \vec{u}_0$$

- Standard Picard iteration can be written as

$$\vec{u}^{k+1} = \vec{u}^k + \underbrace{(\vec{u}_0 - (I - \Delta t Q F)(\vec{u}^k))}_{\text{residual } \vec{r}^k}$$

- Improve it with a preconditioned iteration

$$P\vec{u}^{k+1} = P\vec{u}^k + [\vec{u}_0 - (I - \Delta t Q F)(\vec{u}^k)]$$

- Use the preconditioner $P = (I - \Delta t Q_\Delta F)$ with $Q_\Delta \approx Q$ is a lower triangular approximation of Q

$$(I - \Delta t Q_\Delta F)(\vec{u}^{k+1}) = \vec{u}_0 + \Delta t (Q - Q_\Delta) F(\vec{u}^k)$$

Spectral Deferred correction Sweep

Implicit SDC: $Q_{\Delta}^{LU} = \begin{pmatrix} q_{1,1}^{\Delta} & & & \\ q_{1,2}^{\Delta} & q_{2,2}^{\Delta} & & \\ \vdots & & \ddots & \\ q_{1,M}^{\Delta} & \cdots & \cdots & q_{M,M}^{\Delta} \end{pmatrix}$ with $Q_{\Delta}^{LU} = U^T$ for $Q^T = LU$.

$$u_m^{k+1} = u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_j, u_j^k) + \Delta t \sum_{j=1}^m q_{m,j}^{\Delta} \left[f(\tau_j, u_j^{k+1}) - f(\tau_j, u_j^k) \right]$$

Spectral Deferred correction Sweep

Implicit SDC: $Q_{\Delta}^{LU} = \begin{pmatrix} q_{1,1}^{\Delta} & & & \\ q_{1,2}^{\Delta} & q_{2,2}^{\Delta} & & \\ \vdots & & \ddots & \\ q_{1,M}^{\Delta} & \cdots & \cdots & q_{M,M}^{\Delta} \end{pmatrix}$ with $Q_{\Delta}^{LU} = U^T$ for $Q^T = LU$.

$$u_m^{k+1} = u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_j, u_j^k) + \Delta t \sum_{j=1}^m q_{m,j}^{\Delta} \left[f(\tau_j, u_j^{k+1}) - f(\tau_j, u_j^k) \right]$$

Explicit SDC: $Q_{\Delta}^{EE} = \begin{pmatrix} 0 & & & \\ q_{2,1}^{\Delta} & 0 & & \\ \vdots & & \ddots & \\ q_{M,1}^{\Delta} & \cdots & q_{M-1,M-1}^{\Delta} & 0 \end{pmatrix}$

$$u_m^{k+1} = u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_j, u_j^k) + \Delta t \sum_{j=1}^{m-1} q_{m,j}^{\Delta} \left[f(\tau_j, u_j^{k+1}) - f(\tau_j, u_j^k) \right]$$

Semi-implicit spectral deferred correction method (SISDC)

Consider the case when the initial value problem given previously can be cast in the form

$$\frac{du}{dt} = f(t, u(t)) = f_E(t, u(t)) + f_I(t, u(t)), \quad u \in \mathbb{R}^d, \quad t \in [t_0, t_0 + \Delta t].$$

Semi-implicit spectral deferred correction method (SISDC)

Consider the case when the initial value problem given previously can be cast in the form

$$\frac{du}{dt} = f(t, u(t)) = f_E(t, u(t)) + f_I(t, u(t)), \quad u \in \mathbb{R}^d, \quad t \in [t_0, t_0 + \Delta t].$$

Replace $Q_\Delta F$ with $Q_{\Delta,E}F_E + Q_{\Delta,I}F_I$ in SDC iteration

$$(I - \Delta t Q_{\Delta,E}F_E - \Delta t Q_{\Delta,I}F_I)(\vec{u}^{k+1}) = (I - \Delta t Q_{\Delta,E}F_E - \Delta t Q_{\Delta,I}F_I)(\vec{u}^k) + (\vec{u}_0 - (I - \Delta t QF)(\vec{u}^k))$$

Semi-implicit spectral deferred correction method (SISDC)

Consider the case when the initial value problem given previously can be cast in the form

$$\frac{du}{dt} = f(t, u(t)) = f_E(t, u(t)) + f_I(t, u(t)), \quad u \in \mathbb{R}^d, \quad t \in [t_0, t_0 + \Delta t].$$

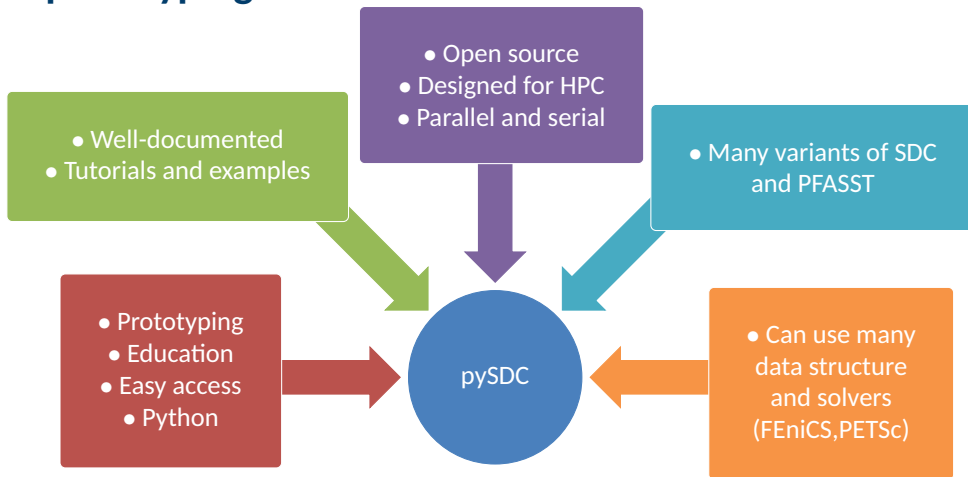
Replace $Q_\Delta F$ with $Q_{\Delta,E}F_E + Q_{\Delta,I}F_I$ in SDC iteration

$$(I - \Delta t Q_{\Delta,E}F_E - \Delta t Q_{\Delta,I}F_I)(\vec{u}^{k+1}) = (I - \Delta t Q_{\Delta,E}F_E - \Delta t Q_{\Delta,I}F_I)(\vec{u}^k) + (\vec{u}_0 - (I - \Delta t QF)(\vec{u}^k))$$

Choose $Q_{\Delta,I}$ lower triangular and $Q_{\Delta,E}$ strictly lower triangular for explicit integration

$$\begin{aligned} u_m^{k+1} &= u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_j, u_j^k) \\ &\quad + \Delta t \sum_{j=1}^{m-1} q_{m,j}^{\Delta,E} \left[f_E(\tau_j, u_j^{k+1}) - f_E(\tau_j, u_j^k) \right] + \Delta t \sum_{j=1}^m q_{m,j}^{\Delta,I} \left[f_I(\tau_j, u_j^{k+1}) - f_I(\tau_j, u_j^k) \right] \end{aligned}$$

pySDC prototyping



<https://parallel-in-time.org/pySDC>

IMEX SDC (SISDC) for advection-diffusion equations

Advection-diffusion of a Gaussian hill problem

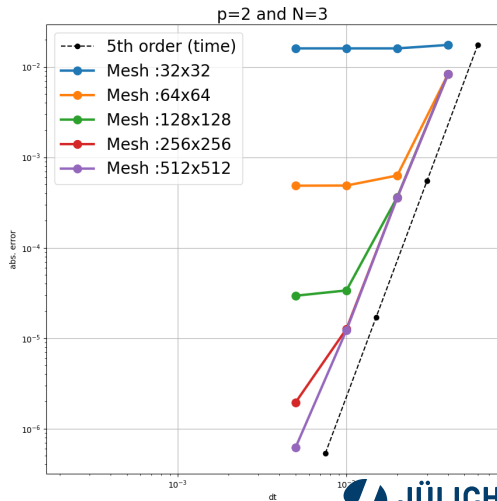
1 Mathematical model

$$\begin{cases} \frac{\partial c}{\partial t} + \mathbf{v} \cdot \nabla c - \nu \Delta c = f & \text{in } \Omega \times [0, T] \\ c = g & \text{on } \partial\Omega \times [0, T] \\ c(x, y, 0) = c_0(x, y) & \text{in } \Omega \end{cases}$$

2 Problem setup

$$c(x, y, t) = \frac{5}{7\sqrt{1 + \frac{4\nu t}{l^2}}} \exp \left\{ - \left(\frac{x - x_0 - t}{l\sqrt{1 + \frac{4\nu t}{l^2}}} \right)^2 \right\}$$

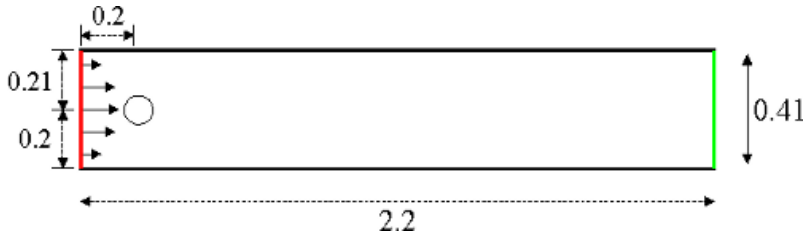
$$\text{with } l = 7\frac{\sqrt{2}}{300}, x_0 = \frac{2}{15} \text{ and } \Omega = [0, 1] \times [0, 1]$$



IMEX SDC (SISDC) using projection-based splitting scheme

DFG flow around cylinder: Benchmark 2D-3

$$\left\{ \begin{array}{ll} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \Delta \mathbf{u} = f & \text{in } \Omega \times [0, T] \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times [0, T] \\ \mathbf{u} = \mathbf{u}_{in} & \text{on } \Gamma_1 \times [0, T] \\ \mathbf{u} = 0 & \text{on } \Gamma_2 \times [0, T] \\ \nu \partial_n \mathbf{u} - p \mathbf{n} = 0 & \text{on } \Gamma_3 \times [0, T] \end{array} \right. \quad \text{with} \quad \begin{array}{l} \mathbf{u}_{in} = \frac{(4Uy(0.41 - y))}{0.41^2}, 0), \\ U = U(t) = 1.5 \sin(\pi t/8) \end{array}$$



IMEX SDC (SISDC) using projection-based splitting scheme

DFG Flow around cylinder: Chorin's projection method

1 Step 1: Predictor Step:

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

2 Step 2: Corrector Step:

$$\Delta p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*$$

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

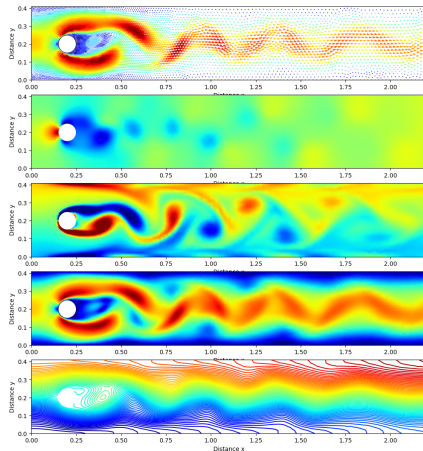


Figure: Velocity, pressure, vorticity, magnitude and streamlines at time $t = 5.5s$

IMEX SDC (SISDC) using projection-based splitting scheme

Lift coefficient using different SDC

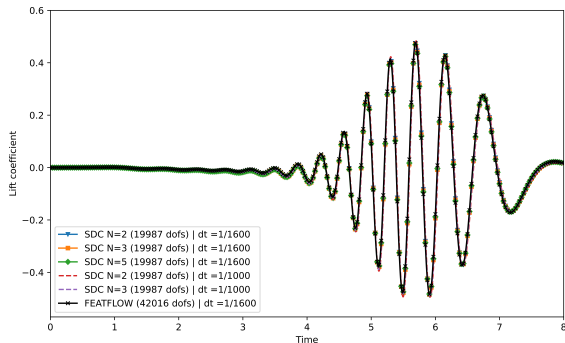


Figure: Lift coefficient using different SDC setups and time steps

IMEX SDC (SISDC) using projection-based splitting scheme

Lift coefficient using different SDC

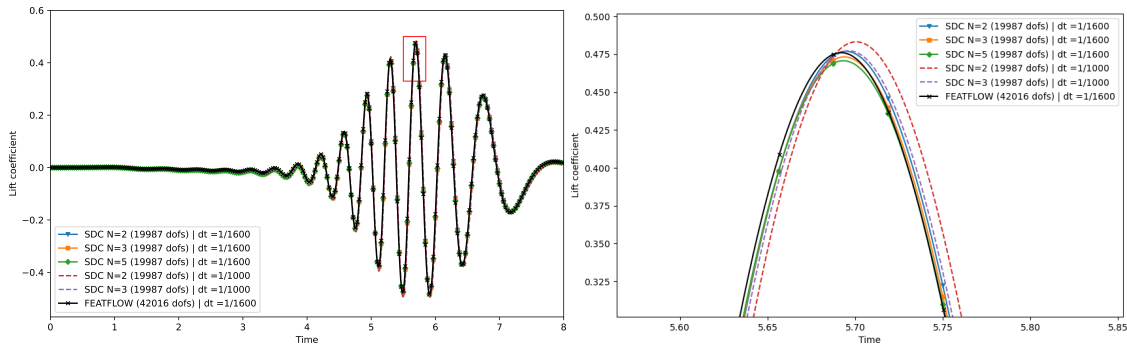


Figure: Lift coefficient using different SDC setups and time steps

Parallel SDC: Diagonal Preconditioners

G. Čaklović et al., Improving Efficiency of Parallel Across the Method Spectral Deferred Corrections. *Preprint in arXiv*

Diagonal SDC: $Q_{\Delta}^{diag} = \text{diag}(d_1, \dots, d_M)$

$$u_m^{k+1} = u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_m, u_j^k) + \Delta t d_m [f(\tau_m, u_m^{k+1}) - f(\tau_m, u_m^k)]$$

Parallel SDC: Diagonal Preconditioners

G. Čaklović et al., Improving Efficiency of Parallel Across the Method Spectral Deferred Corrections. *Preprint in arXiv*

Diagonal SDC: $Q_{\Delta}^{diag} = \text{diag}(d_1, \dots, d_M)$

$$u_m^{k+1} = u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_m, u_j^k) + \Delta t d_m [f(\tau_m, u_m^{k+1}) - f(\tau_m, u_m^k)]$$

1 Diagonal implicit Euler (IEpar)

$$Q_{\Delta}^{IEpar} = \text{diag}(\tau_i - t_0),$$

with τ_i the nodes of the quadrature rule.

Parallel SDC: Diagonal Preconditioners

G. Čaklović et al., Improving Efficiency of Parallel Across the Method Spectral Deferred Corrections. *Preprint in arXiv*

Diagonal SDC: $Q_{\Delta}^{diag} = \text{diag}(d_1, \dots, d_M)$

$$u_m^{k+1} = u_0 + \Delta t \sum_{j=1}^M q_{m,j} f(\tau_m, u_j^k) + \Delta t d_m [f(\tau_m, u_m^{k+1}) - f(\tau_m, u_m^k)]$$

1 Diagonal implicit Euler (IEpar)

$$Q_{\Delta}^{IEpar} = \text{diag}(\tau_i - t_0),$$

with τ_i the nodes of the quadrature rule.

2 MIN-SR-NS diagonal coefficients

For any collocation method on M distinct nodes, then $\rho(Q - Q_{\Delta}) = 0$ for

$$Q_{\Delta} = \text{diag}\left(\frac{\tau_1}{M}, \dots, \frac{\tau_M}{M}\right)$$

3 MIN-SR-S diagonal coefficients

We minimize $\rho(I - Q_{\Delta}^{-1}Q)$ by choosing diagonal coefficients such that

$$\det((1-z)I + zQ_{\Delta}^{-1}Q) = 1, \quad z \in \{\tau_1, \dots, \tau_M\}$$

IMEX SDC (SISDC) using projection-based splitting scheme

Lift coefficient using different SDC

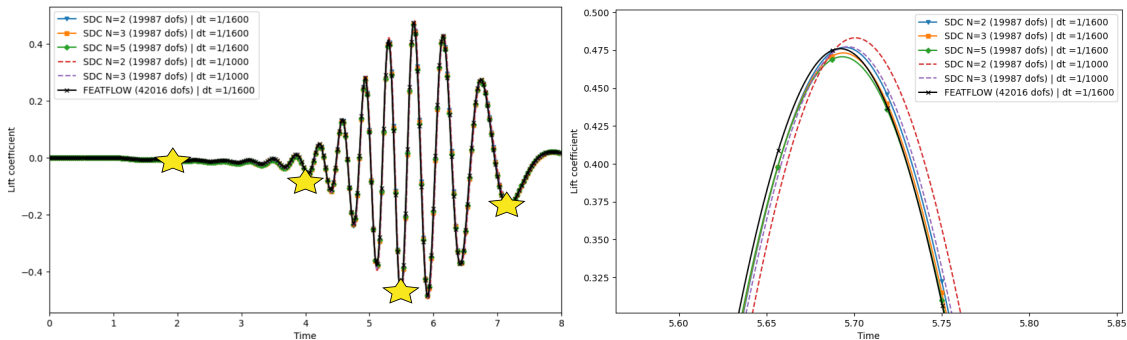


Figure: Lift coefficient using different SDC setups and time steps

IMEX SDC (SISDC) using projection-based splitting scheme

Comparison of different preconditioners

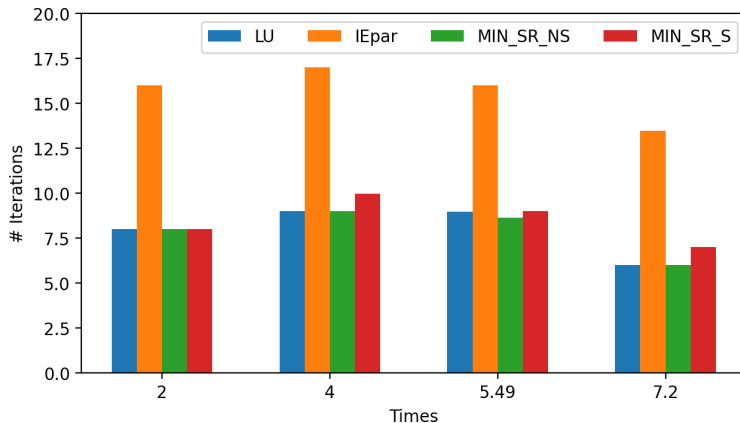


Figure: Average number of iterations needed by various preconditioners after 50 timesteps at four different time points throughout the simulation

Fully implicit SDC using the monolithic approach

- Semi-discrete form of the Navier-Stokes equations.

$$\begin{aligned} [\mathbf{M}] \frac{d\mathbf{u}}{dt} &= -[\mathbf{C}_u]\mathbf{u} - [\mathbf{k}]\mathbf{u} + [\mathbf{B}]\mathbf{p} + [\mathbf{M}]\mathbf{g}, \\ \mathbf{0} &= [\mathbf{B}^T]\mathbf{u} \end{aligned}$$

- Differential algebraic equation (DAEs)

$$\begin{aligned} [\mathbf{M}] \frac{d\mathbf{u}}{dt} &= \mathcal{F}(\mathbf{u}, \mathbf{p}, t), \\ \mathbf{0} &= \mathcal{G}(\mathbf{u}, t). \end{aligned}$$

Fully implicit SDC using the monolithic approach

- Semi-discrete form of the Navier-Stokes equations.

$$\begin{aligned} [M] \frac{d\mathbf{u}}{dt} &= -[C_u]\mathbf{u} - [k]\mathbf{u} + [B]p + [M]\mathbf{g}, \\ \mathbf{0} &= [B^T]\mathbf{u} \end{aligned}$$

- Matrix-vector form of the semi-discrete form

$$\begin{pmatrix} [M] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \frac{d\mathbf{u}}{dt} \\ \frac{dp}{dt} \end{pmatrix} = \begin{pmatrix} -[C_u] - [k] & [B] \\ [B^T] & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} + \begin{pmatrix} [M]\mathbf{g} \\ \mathbf{0} \end{pmatrix}.$$

- Ordinary differential equation (ODE)

$$[\alpha] \frac{d\mathbf{w}}{dt} = [\beta]\mathbf{w} + \gamma = f(\mathbf{w}, t),$$

Fully implicit SDC using the monolithic approach

Lift coefficient using different SDC

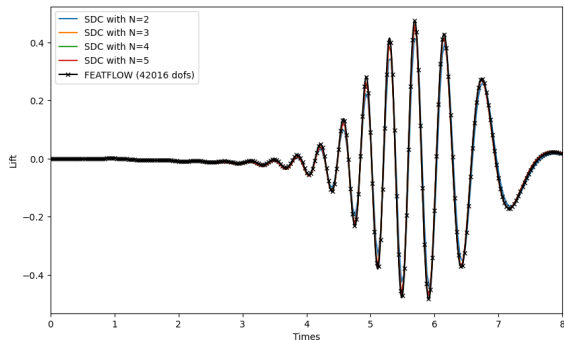


Figure: Lift coefficient using various SDC methods with $\Delta t = \frac{1}{25}$, compared to FEATFLOW reference data with $\Delta t = \frac{1}{1600}$

Fully implicit SDC using the monolithic approach

Lift coefficient using different SDC

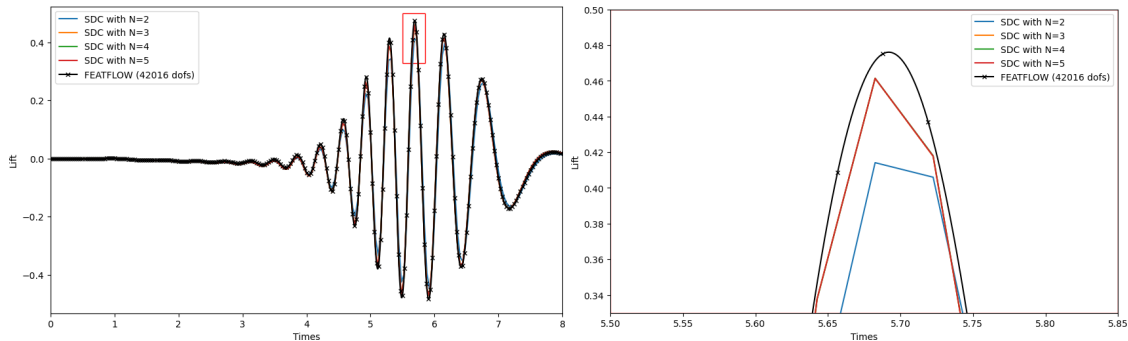


Figure: Lift coefficient using various SDC methods with $\Delta t = \frac{1}{25}$, compared to FEATFLOW reference data with $\Delta t = \frac{1}{1600}$

Fully implicit SDC using the monolithic approach

Comparison of different preconditioners

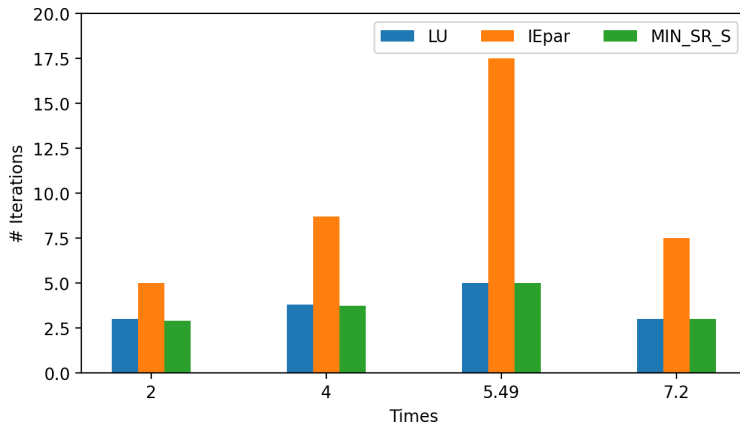
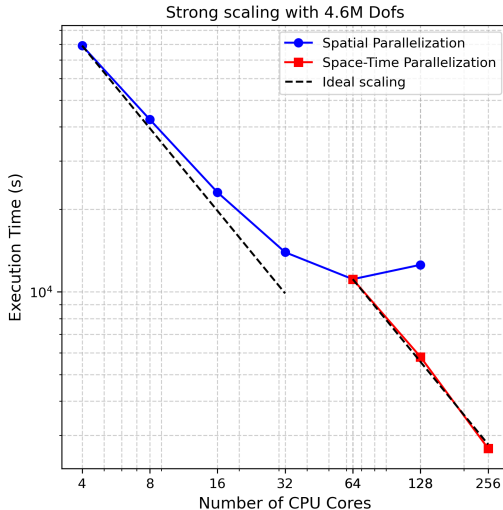


Figure: Average number of iterations needed by various preconditioners after 10 timesteps at four different time points throughout the simulation

Fully implicit SDC using the monolithic approach

Strong scaling tests

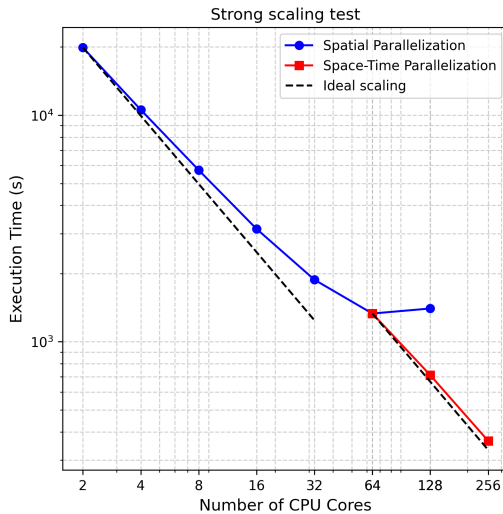
Figure: Strong scaling performance of a space-time parallel solver using pySDC and FEniCSx for the flow around cylinder benchmark with 4.6M dofs



Fully implicit SDC using the monolithic approach

Strong scaling tests

Figure: Strong scaling performance of a space-time parallel solver using SDC and FEATFLOW for the flow around cylinder benchmark with 50k dofs



Summary

- SDC method can use simple numerical method (even a first-order method) to compute a solution with higher-order accuracy.
- SDC can be used for various initial value problems, using explicit, implicit or implicit- explicit schemes.
- Parallelization can be done across the method (i.e. using diagonal preconditioners).

what's next?

- Assess code performance with advanced numerical test cases to ensure robustness.
- Use the method to solve more complex problems.
- Use semi-Lagrangian method to deal with the non-linear convective term.

Acknowledgment

Novel Exascale-Architectures with Heterogeneous Hardware Components for CFD Simulations



Federal Ministry
of Education
and Research



STROEMUNGSRAUM
SCALEXA HPC DEEPTech

Project goals

- Enhance scalability and efficiency of FEATFLOW
- Develop exascale-ready methods
- Provide IANUS customers with access to methods, codes, and knowledge.

Project partners

- IANUS Simulation GmbH
- TU Dortmund University - Coordinator
- Jülich Supercomputing Centre
- University of Cologne
- Friedrich Alexander University Erlangen -Nuremberg
- Freiberg University of Technology

