SPACE–TIME PARALLELIZATION FOR CFD SIMULATIONS

December 15th, 2025 | Abdelouahed Ouardghi, Robert Speck | Jülich Supercomputing Centre, Forschungszentrum Jülich

Acknowledgment

Novel Exascale-Architectures with Heterogeneous Hardware Components for CFD Simulations

Federal Ministry of Education and Research

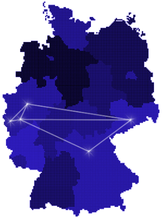
STROEMUNGSRAUM
SCALEXA HPC DEEPTECH

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



JÜLICH Forschungszentrum

Fully Implicit Monolithic Solver for the Navier–Stokes Equations

- Incompressible Navier-Stokes equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \Delta \mathbf{u} &= f \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned}$$

- Matrix-vector form of the semi-discrete form

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

- Ordinary differential equation (ODE)

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

Fully Implicit Monolithic Solver for the Navier–Stokes Equations

Incompressible Navier-Stokes Equations: Flow Around Cylinder

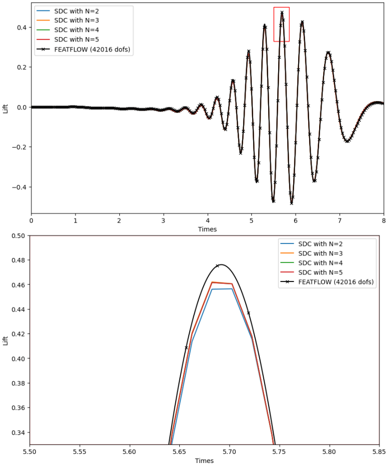


Figure: Lift coefficient using various SDC methods with $\Delta t = \frac{1}{25}$, compared to FEATFLOW reference data with $\Delta t = \frac{1}{1600}$

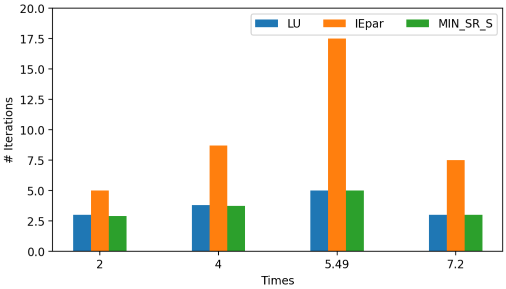


Figure: Average number of iterations needed by various preconditioners after 10 timesteps at four different time points throughout the simulation

Fully Implicit Monolithic Solver for the Navier–Stokes Equations

Incompressible Navier-Stokes Equations: Flow Around Cylinder

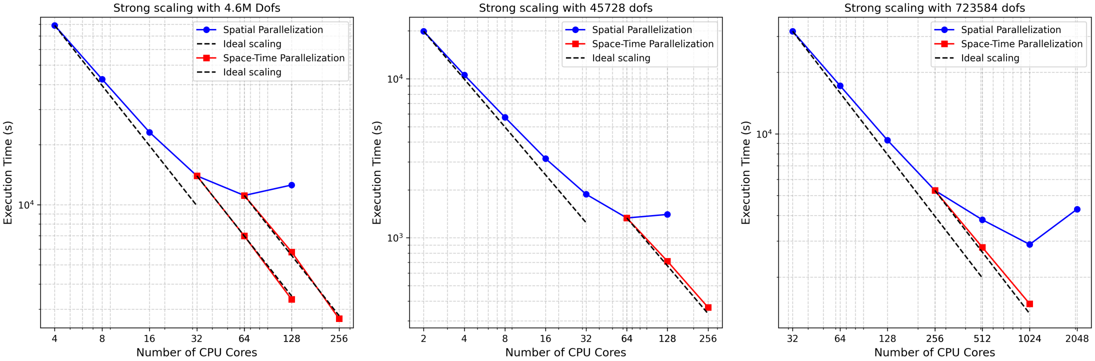


Figure: Strong scaling performance of a space-time parallel solver

Fully Implicit Monolithic Solver for the Navier–Stokes Equations

Advantages

- Strong coupling of velocity and pressure
- Naturally enforces incompressibility ($\nabla \cdot u = 0$) more accurately.
- Larger time steps: Not sensitive to CFL restrictions.
- Better convergence for strongly coupled physics

Disadvantages

- Large linear systems
- Solver complexity
- High per-step computational cost
- Implementation difficulty
- Not ideal for very large industrial meshes without strong solvers

Can we avoid solving the fully coupled nonlinear Navier–Stokes system without sacrificing accuracy or efficiency?

IMEX SDC Projection Methods for Incompressible Navier–Stokes

DFG Flow around cylinder: Chorin's projection method

1. **Step 1: Predictor Step:**

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

2. **Step 2: Corrector Step:**

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

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

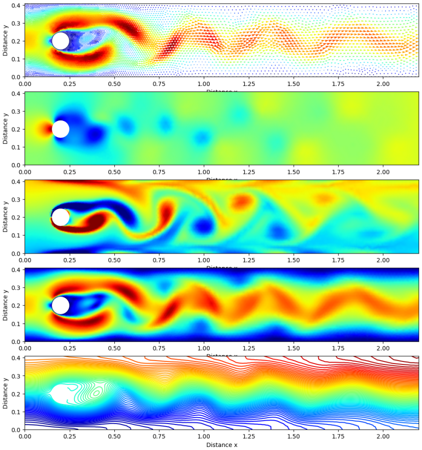


Figure: Velocity, pressure, vorticity, magnitude and streamlines at time $t = 5.5s$

IMEX SDC Projection Methods for Incompressible Navier–Stokes

Convection-diffusion equation

- Convection-diffusion equation

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \Delta \mathbf{u} + \mathbf{g}$$

- Spatial discretization

$$[\mathtt{M}]\frac{d\mathbf{u}}{dt} = -[\mathtt{C}_{\mathbf{u}}]\mathbf{u} - [\mathtt{k}]\mathbf{u} + [\mathtt{M}]\mathbf{g}$$

- ODE

$$[\mathtt{M}]\frac{d\mathbf{u}}{dt} = f(\mathbf{u}, t) = f_E(\mathbf{u}, t) + f_I(\mathbf{u}, t),$$

where $\quad f_E(\mathbf{u}, t) = -[\mathtt{C}_{\mathbf{u}}]\mathbf{u} \quad$ and $\quad f_I(\mathbf{u}, t) = -[\mathtt{k}]\mathbf{u} + [\mathtt{M}]\mathbf{g}$

IMEX SDC Projection Methods for Incompressible Navier–Stokes

Step 1 Predictor step:

$$
\begin{aligned}
[\mathtt{M}]\mathbf{u}_m^{k+1} \quad &= \quad [\mathtt{M}]\mathbf{u}_0 + \Delta t \sum_{j=1}^{M} q_{m,j} f(\mathbf{u}_j^k, \tau_j) \\
&+\Delta t \sum_{j=1}^{m-1} q_{m,j}^{\Delta,E} \left[f_E(\mathbf{u}_j^{k+1}, \tau_j) - f_E(\mathbf{u}_j^k, \tau_j) \right] + \Delta t \sum_{j=1}^{m} q_{m,j}^{\Delta,I} \left[f_I(\mathbf{u}_j^{k+1}, \tau_j) - f_I(\mathbf{u}_j^k, \tau_j) \right]
\end{aligned}
$$

Step 2 Corrector step:

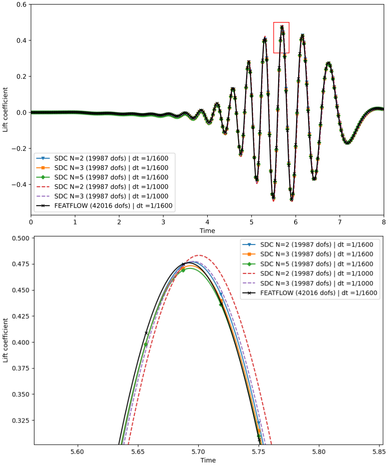
$$
\Delta p_m^{k+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}_m^{k+1}
$$

$$
\mathbf{u}_m^{k+1} = \mathbf{u}_m^{k+1} - \Delta t \nabla p_m^{k+1}
$$

IMEX SDC Projection Methods for Incompressible Navier–Stokes

Incompressible Navier-Stokes Equations: Flow Around Cylinder

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



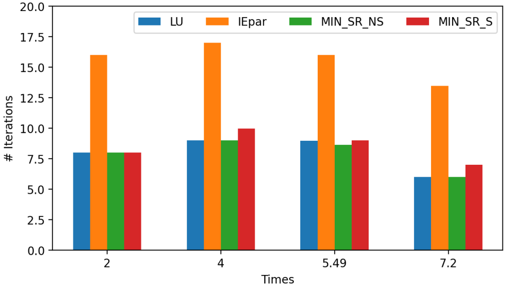


Figure: Average number of iterations needed by various preconditioners after 50 timesteps at four different time points throughout the simulation

JÜLICH Forschungszentrum

IMEX SDC Projection Methods for Incompressible Navier–Stokes

Advantages

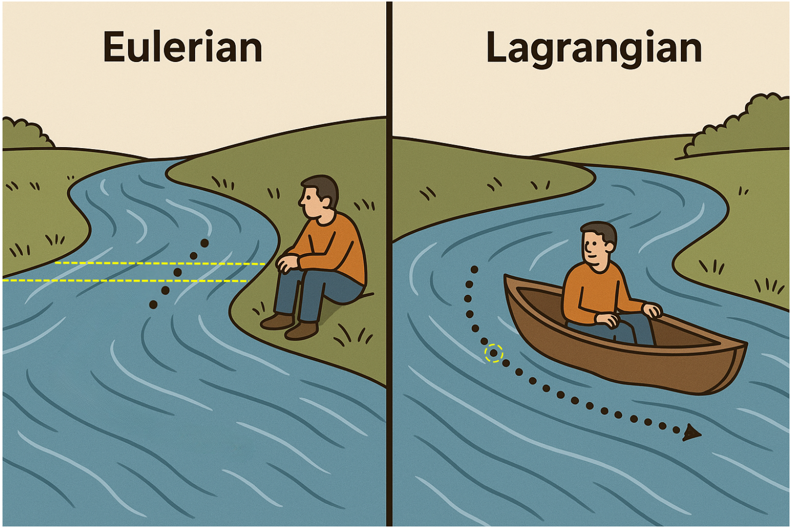
- Computational Efficiency
- Simplicity and Flexibility
- Good for Incompressible Flows
- Widely Used and Well Studied

Disadvantages

- Time Step Constraints
- Instabilities at Small Time Steps (in Some Implementations)
- Splitting Errors

Can projection methods be combined with fully implicit time integration while preserving both accuracy and computational efficiency?

Semi-Lagranian methods or Modified method of characteristics



Semi-Lagranian methods for convection-dominated problems

- Convection-diffusion equation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} = \mathbf{g},$$

- Trajectory of a virtual particle the reached the gird point **x** at time t^{n+1}

$$\begin{aligned} \frac{d\mathcal{X}(t)}{dt} &= \mathbf{u}(\mathcal{X}(t), t), \qquad t \in [t^n, t^{n+1}] \\ \mathcal{X}(t^{n+1}) &= \mathbf{x}. \end{aligned}$$

- The convection–diffusion equation along the trajectory $\mathcal{X}(t)$:

$$\frac{\partial \mathbf{u}(\mathcal{X}(t), t)}{\partial t} + \mathbf{u}(\mathcal{X}(t), t) \cdot \nabla \mathbf{u}(\mathcal{X}(t), t) - \nu \Delta \mathbf{u}(\mathcal{X}(t), t) = \mathbf{g}(\mathcal{X}(t), t).$$

Semi-Lagranian methods for convection-dominated problems

- Convection-diffusion equation

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \nu \Delta \mathbf{u} = \mathbf{g},$$

- Trajectory of a virtual particle the reached the gird point $\mathbf{x}$ at time t^{n+1}

$$\begin{aligned} \frac{d\mathcal{X}(t)}{dt} &= \mathbf{u}(\mathcal{X}(t), t), \qquad t \in [t^n, t^{n+1}] \\ \mathcal{X}(t^{n+1}) &= \mathbf{x}. \end{aligned}$$

- The convection–diffusion equation along the trajectory $\mathcal{X}(t)$:

$$\frac{\partial \mathbf{u}(\mathcal{X}(t), t)}{\partial t} + \frac{d\mathcal{X}(t)}{dt} \cdot \nabla \mathbf{u}(\mathcal{X}(t), t) - \nu \Delta \mathbf{u}(\mathcal{X}(t), t) = \mathbf{g}(\mathcal{X}(t), t).$$

Semi-Lagranian methods for convection-dominated problems

- Simpler ODE along trajectories:

$$\frac{D\mathbf{u}(\mathcal{X}(t),t)}{Dt} - \nu\Delta\mathbf{u}(\mathcal{X}(t),t) = \mathbf{g}(\mathcal{X}(t),t).$$

- Backward Euler method

$$\frac{\mathbf{u}(\mathcal{X}(t^{n+1}),t^{n+1}) - \mathbf{u}(\mathcal{X}(t^n),t^n)}{\Delta t} = \nu\Delta\mathbf{u}(\mathcal{X}(t^{n+1}),t^{n+1}) + \mathbf{g}(\mathcal{X}(t^{n+1}),t^{n+1}).$$

- At the new time t^{n+1}, the particle position is exactly the grid point $\mathbf{x}$:

$$\mathcal{X}(t^{n+1}) = \mathbf{x}.$$

- Therefore

$$\frac{\mathbf{u}(\mathbf{x},t^{n+1}) - \mathbf{u}(\mathcal{X}(t^n),t^n)}{\Delta t} = \nu\Delta\mathbf{u}(\mathbf{x},t^{n+1}) + \mathbf{g}(\mathbf{x},t^{n+1}).$$

Semi-Lagranian methods for convection-dominated problems

- Simpler ODE along trajectories:

$$\frac{D\mathbf{u}(\mathcal{X}(t),t)}{Dt} - \nu\Delta\mathbf{u}(\mathcal{X}(t),t) = \mathbf{g}(\mathcal{X}(t),t).$$

- Backward Euler method

$$\frac{\mathbf{u}(\mathcal{X}(t^{n+1}),t^{n+1}) - \mathbf{u}(\mathcal{X}(t^n),t^n)}{\Delta t} = \nu\Delta\mathbf{u}(\mathcal{X}(t^{n+1}),t^{n+1}) + \mathbf{g}(\mathcal{X}(t^{n+1}),t^{n+1}).$$

- At the new time t^{n+1}, the particle position is exactly the grid point $\mathbf{x}$:

$$\mathcal{X}(t^{n+1}) = \mathbf{x}.$$

- Therefore

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n(\mathcal{X}(t^n))}{\Delta t} = \nu\Delta\mathbf{u}^{n+1} + \mathbf{g}^{n+1}.$$

SDC + semi-Lagrangian method for convection-dominated problems

- We start from the trajectory-based equation

$$\frac{D\mathbf{u}(\mathcal{X}(t),t)}{Dt} - \nu\Delta\mathbf{u}(\mathcal{X}(t),t) = \mathbf{g}(\mathcal{X}(t),t).$$

- The Picard form

$$\mathbf{u}(\mathcal{X}(t),t) - \mathbf{u}(\mathcal{X}(t^n),t^n) = \int_{t^n}^{t} \left(\nu\Delta\mathbf{u}(\mathcal{X}(s),s) + \mathbf{g}(\mathcal{X}(s),s)\right) ds.$$

- collocation problem

$$\mathbf{u}(\mathcal{X}(\tau_m),\tau_m) = \mathbf{u}(\mathcal{X}(t^n),t^n) + \Delta t \sum_{j=1}^{M} q_{m,j}\left(\nu\Delta\mathbf{u}(\mathcal{X}(\tau_j),\tau_j) + \mathbf{g}(\mathcal{X}(\tau_j),\tau_j)\right),$$

SDC + semi-Lagrangian method for convection-dominated problems

- SDC iteration

$$
\begin{aligned}
\mathbf{u}^{k+1}(\mathcal{X}(\tau_m), \tau_m) &= \mathbf{u}(\mathcal{X}(t^n), t^n) \\
&+ \Delta t \sum_{j=1}^{M} (q_{m,j} - q_{m,j}^{\Delta}) \left(\nu \Delta \mathbf{u}^k(\mathcal{X}(\tau_j), \tau_j) + \mathbf{g}(\mathcal{X}(\tau_j), \tau_j)\right) \\
&+ \Delta t \sum_{j=1}^{m} q_{m,j}^{\Delta} \left(\nu \Delta \mathbf{u}^{k+1}(\mathcal{X}(\tau_j), \tau_j) + \mathbf{g}(\mathcal{X}(\tau_j), \tau_j)\right).
\end{aligned}
$$

- Characteristic curves

$$
\begin{aligned}
\frac{d\mathcal{X}(t)}{dt} &= \mathbf{u}(\mathcal{X}(t), t), \\
\mathcal{X}(\tau_m) &= \mathbf{x},
\end{aligned}
$$

SDC + semi-Lagrangian method for convection-dominated problems

- SDC iteration

$$
\begin{aligned}
\mathbf{u}^{k+1}(\mathbf{x}, \tau_m) \quad &= \quad \mathbf{u}(\mathcal{X}^n, t^n) \\
&+ \quad \Delta t (q_{m,m} - q^{\Delta}_{m,m}) \left(\nu \Delta \mathbf{u}^k(\mathbf{x}, \tau_m) + \mathbf{g}(\mathbf{x}, \tau_m)\right) \\
&+ \quad \Delta t \sum_{\substack{j=1 \\ j \neq m}}^{M} (q_{m,j} - q^{\Delta}_{m,j}) \left(\nu \Delta \mathbf{u}^k(\mathcal{X}_j, \tau_j) + \mathbf{g}(\mathcal{X}_j, \tau_j)\right) \\
&+ \quad \Delta t \sum_{j=1}^{m-1} q^{\Delta}_{m,j} \left(\nu \Delta \mathbf{u}^{k+1}(\mathcal{X}_j, \tau_j) + \mathbf{g}(\mathcal{X}_j, \tau_j)\right) \\
&+ \quad \Delta t q^{\Delta}_{m,m} \left(\nu \Delta \mathbf{u}^{k+1}(\mathbf{x}, \tau_m) + \mathbf{g}(\mathbf{x}, \tau_m)\right) .
\end{aligned}
$$

SDC + semi-Lagrangian method for convection-dominated problems

- SDC iteration

$$\begin{aligned}
\mathbf{u}_m^{k+1} &= \mathbf{u}^n(\mathcal{X}^n) \\
&+ \Delta t(q_{m,m} - q^{\Delta}_{m,m})(\nu\Delta\mathbf{u}_m^k + \mathbf{g}_m) \\
&+ \Delta t \sum_{\substack{j=1 \\ j\neq m}}^{M} (q_{m,j} - q^{\Delta}_{m,j})\left(\nu\Delta\mathbf{u}_j^k(\mathcal{X}_j) + \mathbf{g}_j(\mathcal{X}_j)\right) \\
&+ \Delta t \sum_{j=1}^{m-1} q^{\Delta}_{m,j}\left(\nu\Delta\mathbf{u}_j^{k+1}(\mathcal{X}_j) + \mathbf{g}_j(\mathcal{X}_j)\right) \\
&+ \Delta t q^{\Delta}_{m,m}(\nu\Delta\mathbf{u}_m^{k+1} + \mathbf{g}_m).
\end{aligned}$$

SDC + semi-Lagrangian method for convection-dominated problems

The SDC semi-Lagrangian iteration can be written as

$$
\begin{aligned}
\mathbf{u}_m^{k+1} - \Delta t q_{m,m}^{\Delta} \nu \Delta \mathbf{u}_m^{k+1} &= \mathbf{u}^n(\mathcal{X}^n) + \Delta t (q_{m,m} - q_{m,m}^{\Delta}) \nu \Delta \mathbf{u}_m^k + \Delta t q_{m,m} \mathbf{g}_m \\
&+ \Delta t \sum_{\substack{j=1 \\ j \neq m}}^{M} (q_{m,j} - q_{m,j}^{\Delta}) \left(\nu \Delta \mathbf{u}_j^k(\mathcal{X}_j) + \mathbf{g}_j(\mathcal{X}_j) \right) \\
&+ \Delta t \sum_{j=1}^{m-1} q_{m,j}^{\Delta} \left(\nu \Delta \mathbf{u}_j^{k+1}(\mathcal{X}_j) + \mathbf{g}_j(\mathcal{X}_j) \right).
\end{aligned}
$$

Conclusion

- Monolithic fully implicit formulations offer excellent stability and accuracy for incompressible Navier–Stokes equations
  $\longrightarrow$ but they require solving very large nonlinear coupled systems, which limits scalability and space-time parallel efficiency.
- Projection-based methods significantly reduce computational complexity by splitting velocity and pressure
  incompressible Navier–Stokes equations
  $\longrightarrow$ however, they introduce CFL-type time step restrictions and often require many time steps, limiting performance for long-time simulations.
- Combining semi-Lagrangian advection, implicit treatment of stiff terms, and projection-based incompressibility enforcement offers a promising compromise