No. 686 **May 2026**

A Parallel-in-Time Navier-Stokes Solver Using Augmented Lagrangian Acceleration and Space-Time Multigrid Methods

J. Dünnebacke, C. Lohmann, S. Turek

ISSN: 2190-1767

A Parallel-in-Time Navier-Stokes Solver Using Augmented Lagrangian Acceleration and Space-Time Multigrid Methods

Jonas Dünnebacke[1], Christoph Lohmann[1], and Stefan Turek[1]

[1]Institute of Applied Mathematics (LS III), TU Dortmund University, Vogelpothsweg 87, D-44227 Dortmund, Germany

Abstract

In this work we present a parallel-in-time solver of the Navier-Stokes equation for low Reynolds numbers. The solver uses an all-at-once Newton-Krylov method and the combination of Augmented Lagrangian and LSC preconditioning leading to embarrassingly parallel pressure solves. For the solution of the unsteady velocity system we compare multigrid waveform relaxation and space-time multigrid. For low and moderate Reynolds numbers this solver can improve the strong scalability compared to a time stepping solver. This method is also applied to a Carreau fluid showing the possible application in the area of transient simulation of polymer melts.

1 Introduction

Modern high performance computing (HPC) systems rely more on parallelization than ever before. While clock frequencies barely increase anymore, higher performance is achieved by increasing processor core counts, utilizing wider SIMD vectorization, and using more processors and nodes. Furthermore, driven by AI workloads, there is a shift towards GPU based hardware accelerators employing SIMT parallelism that need multiple thousands of threads running at once to achieve high performance.

Traditionally, transient PDE simulations rely on time stepping schemes and parallelize in space by either domain decomposition, data-parallel multigrid methods or combinations thereof. However, in many cases the parallelization in space does not provide enough parallelism to make use of all compute resources of current HPC systems. This may be due to small spatial problems or dominating coarse grid solves in multigrid methods. Even though, solving a single time step is usually fast in these cases, the overall time to simulate the problem can get large when many time steps need to be solved for accuracy reasons.

To overcome these challenge parallel-in-time (PinT) methods gain popularity. For a broad overview on PinT methods, we refer to the review papers by Gander [17] and

Ong and Schroder [26]. The most prominent PinT method, Parareal [22], was applied to solve the Navier-Stokes equation by Trindade [32] and Fischer et al. [14]. The performance of this application [5] and the influence of the Reynolds number [31] have been extensively studied. Falgout et al. [12] applied multigrid reduction in time to fluid flow problems. Other viable strategies for the parallel in time solution of flow problems are the usage of stage parallel time integrators or preconditioners [27] or exponential time integrators [21]. Benedusi et al. [1] have applied a multigrid method to a global-in-time discretization of a time-periodic, incompressible Navier-Stokes problem.

We follow the approach of Lohmann and Turek [23] and use an augmented Lagrangian (AL) preconditioner applied to an all-at-one discretization. We briefly discuss, how the time discretization can be modified, so that the pressure problems of the Schur complement preconditioner have a Kronecker product structure allowing for embarrassingly parallel pressure solves at each time step. For the velocity problem we evaluate the use of space-time multigrid (STMG) [16] and multigrid waveform relaxation (WRMG) [25, 20] based methods and analyze the scaling behavior of the overall solver. As application, we consider Newtonian and non-Newtonian fluid models in the regime of low Reynold numbers.

Similar methods were analyzed by Danieli et al. [6], where a PCD preconditioner was applied to global-in-time system without the augmented Lagrangian modification and by Jackaman and MacLachlan [19] that solves the global saddle point system by a Vanka based multigrid method.

2 Equation and Discretization

For a given kinematic viscosity parameter $\nu = \frac{\mu}{\rho}$, we solve the unsteady incompressible Navier-Stokes equations

$$\begin{aligned} \partial_t u + (u \cdot \nabla)\, u - \frac{\mu}{\rho}\Delta u + \nabla p &= f, \\ \nabla \cdot u &= 0 \end{aligned}$$

for the velocity field $u \in [H^1(0,T;H^2(\Omega))]^2$ and the kinematic pressure field $p \in L^2(0,T;H^1(\Omega))$. The derivation assumes that the deviatoric stress $\tau = 2\mu\varepsilon$ depends linearly upon the shear rate tensor $\varepsilon(\nabla u) = \frac{1}{2}\left(\nabla u + (\nabla u)^\top\right)$ by a constant viscosity μ. This assumption is not valid for more complex fluids like polymer melts. We also consider Carreau fluids [3] where the effective viscosity μ_{eff} depends on the shear rate $\dot\gamma = \sqrt{2\varepsilon(u) : \varepsilon(u)}$ according to

$$\mu_{\text{eff}}(\dot\gamma) = \mu_\infty + (\mu_0 - \mu_\infty)\left(1 + (\lambda\dot\gamma)^2\right)^{\frac{n-1}{2}}$$

and a set of material parameters $\mu_0, \mu_\infty, \lambda$ and n. Due to the shear-rate dependent viscosity the divergence of the rate tensor does not simplify to a Laplacian and the system of equation reads

$$\begin{aligned} \partial_t u + (u \cdot \nabla)\, u - \frac{1}{\rho}\nabla \cdot (2\mu_{\text{eff}}(\dot\gamma)\varepsilon(u)) + \nabla p &= f, \\ \nabla \cdot u &= 0. \end{aligned}$$

For the remainder of this article we set $\rho = 1$ and always consider the viscosity parameters μ_0, μ_∞ and the pressure p to be kinematic quantities.

The discretization is done separately in time and space by a tensor product approach. Therefore, both discretizations can be chosen independently, but the spatial mesh has to be fixed in time. For the spatial discretization we use the LBB-stable $Q_2 - P_1$ finite element pair on a quadrilateral triangulation $\mathcal{T}_h$. The velocity field is approximated by the continuous and piecewise biquadratic finite element space

$$V_h = \{v_h \in C(\Omega) : v_h|_K \in \mathbb{Q}_2(K) \; \forall K \in \mathcal{T}_h\}$$

while discontinuous, linear elements

$$W_h = \{w_h \in L^2(\Omega) : w_h|_K \in \mathbb{P}_1(K) \; \forall K \in \mathcal{T}_h\}$$

are used for the pressure variable. Other discretizations may be used, but special smoothers are required for the Augmented Lagrangian approach [2, 13, 34].

This leads to the semi-discrete problem

$$\begin{aligned} \mathbf{M}_u \partial_t \mathbf{u}_h(t) + \mathbf{S}_h(\mathbf{u}_h(t))\mathbf{u}_h(t) + \mathbf{B}_h \mathbf{p}_h(t) &= \mathbf{f}_h(t) \\ \mathbf{B}_h^\top \mathbf{u}_h(t) &= 0 \end{aligned} \tag{1}$$

with the velocity and pressure vectors $\mathbf{u}_h \in \mathbb{R}^{n_v}$ and $\mathbf{p}_h \in \mathbb{R}^{n_p}$. $n_v = \dim V_h$ and $n_p = \dim W_h$ are the number of degrees of freedom (DoF) of the velocity and pressure space respectively. $\mathbf{M}_u$ and $\mathbf{S}_h \in \mathbb{R}^{n_{v,h} \times n_{v,h}}$ are the velocity mass matrix and the stiffness matrix. $\mathbf{B}_h \in \mathbb{R}^{n_{v,h} \times n_{p,h}}$ is the discrete pressure gradient. Here, we ignore the boundary conditions, so that the discrete divergence operator is given by $\mathbf{B}^\top$. As there is no time derivative of the pressure in the continuity equation, this is a differential-algebraic system of equations.

2.1 Time discretization

Similar to [16], we consider a discontinuous Galerkin (DG) time discretization of the initial value problem

$$\begin{aligned} \partial_t u(t) + a u(t) &= f(t) \\ u(0) &= u_0 \end{aligned} \tag{2}$$

involving a linear scalar ordinary differential equation (ODE).

We choose a time mesh $0 = t_0 < t_1 < \cdots < t_K$ with $K \in \mathbb{N}$ time steps and an associated mesh size τ which is assumed to be constant throughout this article for a more comprehensible notation. The ansatz space of the DG(p) method is defined by

$$V_{\tau,p} = \{v \in L^2(0,T) : v|_{T_i} \in \mathbb{P}_p(T_i) \; \forall i = 1, \ldots, K\}$$

where the functions are polynomials of degree p on each time interval $T_i = (t_{i-1}, t_i)$. Using the notation of the jump $[\![u_\tau]\!]_i = u_\tau|_{T_{i+1}}(t_i) - u_\tau|_{T_i}(t_i)$ and $u_\tau|_{T_0}(t_0) = u(0)$, we can define the DG(p) approximation of equation (2).

Definition 1. The DG(p) approximation $u_\tau \in V_{\tau,p}$ to the solution of the ODE (2) satisfies

$$\sum_{i=1}^{K} \int_{T_i} \partial_t u_\tau(t) \partial_t v(t) + a u_\tau(t) v_\tau(t) \mathrm{d}t + [\![u_\tau]\!]_{i-1} \, v_\tau|_{T_i}(t_{i-1}) \\ = \sum_{i=1}^{K} \int_{T_i} f(t) v_\tau(t) \mathrm{d}t \qquad \forall v_\tau \in V_{\tau,p}.$$

Choosing a basis $\{\Psi_0, \dots, \Psi_p\}$ of $\mathbb{P}_p([0,1])$ and using linear transformations $F_i : [0,1] \to T_i$ leads to the coefficient matrices $\mathbf{M}_{DG}, \mathbf{A}_{DG}, \mathbf{C}_{DG} \in \mathbb{R}^{(p+1)\times(p+1)}$ defined by

$$(\mathbf{M}_{DG})_{k,l} = \int_0^1 \Psi_k \Psi_l, \quad (\mathbf{C}_{DG})_{k,l} = -\Psi_k(0)\Psi_l(1) \quad \text{and}$$
$$(\mathbf{A}_{DG})_{k,l} = \int_0^1 \Psi_k \partial_t \Psi_l + \Psi_k(0)\Psi_l(0).$$

The right-hand side coefficients on each time interval T_i are given by $(\tilde{\mathbf{f}}_i)_k = \int_{T_i} f(F_i \circ \Psi_k)$.

Choosing an L^2-orthonormal basis and exact quadrature yields $\mathbf{M}_{DG} = \mathbf{I}_p$, however, this does not hold for time dependent data and nonlinear problems. Instead, we choose the right Gauß-Radau quadrature that includes the right boundary of the reference interval $(0,1)$ as quadrature node. Under this quadrature with nodes q_i and weights ω_i the nodal basis satisfying $\Psi_i(q_j) = \sqrt{1/\omega_i}\delta_{ij}$ for $i,j = 0,\dots,p$ is L^2-orthonormal. The Gauß-Radau quadrature is of order $2p+1$ which is enough to maintain the expected convergence order $p+1$ of the L^2-error. Because this discretization is equivalent to Radau-IIA Runge-Kutta methods [18], we can expect super-convergence in the time points t_i of order $2p+1$.

We define the temporal mass and stiffness matrices $\mathbf{M}_\tau, \mathbf{A}_\tau \in \mathbb{R}^{sK\times sK}$ by

$$\mathbf{M}_\tau = \mathbf{M}_{DG} \otimes \mathbf{I}_K \quad \text{and} \quad \mathbf{A}_\tau = \begin{bmatrix} \mathbf{A}_{DG} & & & \\ \mathbf{C}_{DG} & \mathbf{A}_{DG} & & \\ & \ddots & \ddots & \\ & & \mathbf{C}_{DG} & \mathbf{A}_{DG} \end{bmatrix}$$

with the number of stages per time step $s = p+1$.

Applying this discretization to the semi-discrete problem (1) yields the global-in-time block system

$$\begin{bmatrix} \mathbf{A}_K & \mathbf{B}_K \\ \mathbf{B}_K^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} = \begin{bmatrix} \mathbf{f}_K \\ 0 \end{bmatrix} \tag{3}$$

using the right-hand side vector $\mathbf{f}_K = [\tilde{\mathbf{f}}_1^\top, \dots, \tilde{\mathbf{f}}_K^\top]^\top$, the velocity vector $\mathbf{u}_K$ and the pressure vector $\mathbf{p}_K$ that contain the DoFs of all time steps. In case of a constant spatial stiffness operator the block matrices read

$$\mathbf{A}_K = \frac{1}{\tau}\mathbf{A}_\tau \otimes \mathbf{M}_u + \mathbf{M}_\tau \otimes \mathbf{S}_h \qquad \text{and} \qquad \mathbf{B}_K = \mathbf{I}_{n_K} \otimes \mathbf{B}_h$$

where $n_K = sK$ is the total number of temporal DoFs. Note, that we use the identity matrix $\mathbf{I}_{n_K} \in \mathbb{R}^{K\times K}$ in the definition of $\mathbf{B}_K$. When the temporal mass matrix $\mathbf{M}_\tau$ of a given discretization is not an identity matrix, the required structure of $\mathbf{B}_K$ can often be achieved by a transformation of variables.

Example (Crank-Nicolson discretization). A Crank-Nicolson (CN) discretization results in the temporal mass and system matrices given by

$$\mathbf{A}_\tau = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{M}_\tau = \frac{1}{2}\begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix} \in \mathbb{R}^{K\times K}$$

and the corresponding global vectors are

$$\mathbf{u}_K = \begin{bmatrix} \mathbf{u}_h(t_1) \\ \mathbf{u}_h(t_2) \\ \vdots \\ \mathbf{u}_h(t_K) \end{bmatrix}, \quad \mathbf{p}_K = \begin{bmatrix} \tilde{\mathbf{p}}_{h,1} \\ \tilde{\mathbf{p}}_{h,2} \\ \vdots \\ \tilde{\mathbf{p}}_{h,K} \end{bmatrix} \quad \text{and} \quad \mathbf{f}_K = \frac{1}{2}\begin{bmatrix} \mathbf{f}_h(t_0)+\mathbf{f}_h(t_1)+2\tau^{-1}\mathbf{u}_h(t_0) \\ \mathbf{f}_h(t_1)+\mathbf{f}_h(t_2) \\ \vdots \\ \mathbf{f}_h(t_{K-1})+\mathbf{f}_h(t_K) \end{bmatrix}$$

using the auxiliary pressure variables $\tilde{\mathbf{p}}_{h,i} = 0.5(\mathbf{p}_h(t_{i-1}) + \mathbf{p}_h(t_i))$. This substitution leads to the desired structure of equation (3) with $\mathbf{B}_K = \mathbf{I}_K \otimes \mathbf{B}_h$. The new pressure can also be used as second order approximation in the midpoint of the time step $\tilde{\mathbf{p}}_{h,i} \approx \mathbf{p}_h\,(0.5(t_{i-1} + t_i))$.

3 Solver description

To solve the global nonlinear system (3) simultaneously for all time steps by employing a Schur-complement approach, we linearize the system by means of a global-in-time Newton's method. If the Newton's method does not converge due to bad initial guesses, the alternating Picard-Newton iteration [28] often performs better. This method switches between a Picard and Newton linearization in each nonlinear step.

3.1 Augmented Lagrangian Block-Preconditioning

For the linear solver we adopt Lohmann's approach [23]. Let us consider the linearized problem

$$\begin{bmatrix} \mathbf{A}_K & \mathbf{B}_K \\ \mathbf{B}_K^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} = \begin{bmatrix} \mathbf{f}_K \\ \mathbf{g}_K \end{bmatrix}$$

which has a non-vanishing right-hand side in the continuity equation, because intermediate solutions might not satisfy the continuity equation. A common approach for solving the (Navier-)Stokes equation is the use of augmented Lagrangian preconditioning where the residual of the continuity equation is added as a penalty term to the momentum equation [15, 2]. We apply this idea directly to the global block system, so that the modified system reads

$$\begin{bmatrix} \mathbf{A}_K + \gamma\mathbf{B}_K\mathbf{W}^{-1}\mathbf{B}_K^\top & \mathbf{B}_K \\ \mathbf{B}_K^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_K \\ \mathbf{p}_K \end{bmatrix} = \begin{bmatrix} \mathbf{f}_K + \gamma\mathbf{B}_K\mathbf{W}^{-1}\mathbf{g}_K \\ \mathbf{g}_K \end{bmatrix} \tag{4}$$

with a symmetric positive (semi-)definite matrix $\mathbf{W}$. Here, we choose a block-diagonal extension of the pressure mass matrix $\mathbf{W} = \mathbf{I}_{n_K} \otimes \mathbf{M}_{p,h}$. Due to the use of discontinuous pressure spaces and the Kronecker structure of $\mathbf{W}$, the inverse $\mathbf{W}^{-1}$ is easy to compute and the modified top left block

$$\mathbf{A}_{K,\gamma} := \mathbf{A}_K + \gamma \mathbf{B}_K \mathbf{W}^{-1} \mathbf{B}_K^\top = \mathbf{A}_K + \gamma \mathbf{I}_{n_K} \otimes \left(\mathbf{B}_h \mathbf{M}_{p,h}^{-1} \mathbf{B}_h^\top \right)$$

of equation (4) retains its original sparsity pattern.

Using a suitable approximation of the Schur complement $\mathbf{S}_{K,\gamma} = \mathbf{B}_K^\top \mathbf{A}_{K,\gamma}^{-1} \mathbf{B}_K$ and the decomposition

$$\begin{bmatrix} \mathbf{A}_{K,\gamma} & \mathbf{B}_K \\ \mathbf{B}_K^\top & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_{v,K}} & 0 \\ \mathbf{B}_K^\top \mathbf{A}_{K,\gamma}^{-1} & \mathbf{I}_{n_{p,K}} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{A}_{K,\gamma} & \mathbf{B}_K \\ 0 & -\mathbf{S}_{K,\gamma} \end{bmatrix}}_{P}$$

we can solve the system (4) by applying an FGRMES method with the triangular block-preconditioner P^{-1}. The Shermann-Morrison-Woodburry identity leads to

$$\mathbf{S}_{K,\gamma}^{-1} = \left(\mathbf{B}_K^\top \mathbf{A}_{K,\gamma}^{-1} \mathbf{B}_K \right)^{-1} = \left(\mathbf{B}_K^\top \mathbf{A}_K^{-1} \mathbf{B}_K \right)^{-1} + \gamma \mathbf{W}^{-1} = \mathbf{S}_K^{-1} + \gamma \mathbf{W}^{-1} \tag{5}$$

where $\mathbf{S}_K$ is the Schur complement matrix of $\mathbf{A}_K$ [2, 23]. Using the inverse pressure mass matrix as an approximation to the inverse of the original Schur complement $\mathbf{S}_K^{-1} \approx \nu \mathbf{I}_{n_K} \otimes \mathbf{M}_{p,h}$, we define the block matrix

$$P = \begin{bmatrix} \mathbf{A}_{K,\gamma} & \mathbf{B}_K \\ 0 & -\mathbf{S}_{UZ} \end{bmatrix} \text{ with } \mathbf{S}_{UZ}^{-1} = (\nu + \gamma) \mathbf{I}_{n_K} \otimes \mathbf{M}_{p,h} \tag{6}$$

that we use as preconditioner in an FGMRES iteration to solve the modified system (4). For the remainder we will call $\mathbf{S}_{UZ}$ Uzawa preconditioner to make the distinction to the least squares commutator preconditioner more clear, even though in the literature the AL method often indicates the usage of this block-preconditioner.

In the application of this preconditioner the inverse of the pressure mass matrix has to be applied to every time stage. Afterwards, a velocity problem in space-time has to be solved via an approximate multigrid solver, typically we execute a single V-cycle.

3.1.1 Least squares commutator

Another frequently used approximation of the Schur complement is given by the least squares commutator (LSC) preconditioner [11], also known under the name BFBT [10]. The application of this preconditioner to the global-in-time system (3) was considered in [24]. Following [23], applying this preconditioner directly to the modified Schur complement $\mathbf{S}_{K,\gamma}$ yields

$$\begin{aligned} \mathbf{S}_{LSC,1}^{-1} &= (\mathbf{B}_K^\top \mathbf{V}^{-1} \mathbf{B}_K)^{-1} \mathbf{B}_K^\top \mathbf{V}^{-1} \mathbf{A}_{K,\gamma} \mathbf{V}^{-1} \mathbf{B}_K (\mathbf{B}_K^\top \mathbf{V}^{-1} \mathbf{B}_K)^{-1} \\ &= \mathbf{D}_{K,p}^{-1} \left(\mathbf{I}_{n_K} \otimes \left(\mathbf{B}_h^\top \mathbf{M}_{u,h}^{-1} \right) \right) \mathbf{A}_{K,\gamma} \left(\mathbf{I}_{n_K} \otimes \left(\mathbf{M}_{u,h}^{-1} \mathbf{B}_h \right) \right) \mathbf{D}_{K,p}^{-1} \end{aligned}$$

with a block diagonal extension of the velocity mass matrix $\mathbf{V} = \mathbf{I}_{n_K} \otimes \mathbf{M}_{u,h}$. This, together with the block-diagonal structure of matrices $\mathbf{B}_K$ and $\mathbf{B}_K^\top$, implies that the pressure Laplace matrix reads $\mathbf{D}_{K,p} = \mathbf{I}_{n_K} \otimes (\mathbf{B}_h^\top \mathbf{M}_{u,h}^{-1} \mathbf{B}_h)$. The linear systems associated with $\mathbf{D}_{K,p}$ decompose into independent subproblems for each time stage. Alternatively, we can interpret this as a single spatial system with multiple right-hand sides and apply block-Krylov methods and use batched linear algebra kernel.

Analytically, by applying the LSC preconditioner only onto $\mathbf{S}_K$ and using the decomposition (5) the LSC preconditioner can also be written as

$$\mathbf{S}_{LSC,2}^{-1} = \mathbf{D}_{K,p}^{-1} \left(\mathbf{I}_{n_K} \otimes \left(\mathbf{B}_h^\top \mathbf{M}_{u,h}^{-1} \right) \right) \mathbf{A}_K \left(\mathbf{I}_{n_K} \otimes \left(\mathbf{M}_{u,h}^{-1} \mathbf{B}_h \right) \right) \mathbf{D}_{K,p}^{-1} + \gamma \mathbf{W}^{-1}.$$

Although the identity $\mathbf{S}_{LSC,1}^{-1} = \mathbf{S}_{LSC,2}^{-1}$ holds, we distinguish between both versions, because they behave differently when the action of $\mathbf{D}_{K,p}^{-1}$ is approximated by inexact solves.

3.2 Multigrid methods for the velocity problems

In this section we introduce the iterative solvers for the pressure and velocity subproblems. The pressure solver is a standard spatial multigrid method with a Jacobi preconditioned FGMRES smoother. The velocity systems containing $\mathbf{A}_{K,\gamma}$ are solved by the space-time multigrid method or multigrid waveform relaxation. However, we need to use specialized smoothers and transfer operators to handle the modification by the augmented Lagrangian, as the penalty matrix $\mathbf{B}_K \mathbf{W}^{-1} \mathbf{B}_K^\top$ is singular and hinders the rapid convergence of the multigrid when γ is large.

We briefly reiterate how the modified system may be solved in a stationary or time stepping simulation [13, 34]. The multigrid method uses an additive Schwarz smoother using a star-patch decomposition [30]. For a given triangulation $\mathcal{T}_h$ we define the decomposition $V_h = \sum V_{h,j}$ into local FE-spaces $V_{h,j} = \{\mathbf{v}_h \in V_h : \mathrm{supp}(\mathbf{v}_h) \subset \mathrm{star}(x_j)\}$. Each local space $V_{h,j}$ consists of FE-functions whose support is a subset of the patch $\mathrm{star}(x_j) = \bigcup_{K \in \mathcal{T}_h, x_j \in \bar{K}} \bar{K}$. So for every mesh node x_j the union of all adjacent mesh elements defines a star-patch and the subspaces $V_{h,j}$ are the span of basis functions whose support is a subset of each patch.

Using the restriction operator $\mathcal{R}_{h,i} : V_h \to V_{h,i}$ that only selects the interior DoFs of a patch $\mathrm{star}(x_i)$, we define the additive Schwarz preconditioner

$$\mathcal{S} = \sum_i \mathcal{R}_{h,i}^\top \left(\mathcal{R}_{h,i} \mathbf{A}_{h,\gamma} \mathcal{R}_{h,i}^\top \right)^{-1} \mathcal{R}_{h,i}.$$

This means, that we do a parallel correction of each patch by solving a local system. The operator $\mathcal{S}$ is then used as preconditioner of an FGMRES relaxation.

A similar modification is done to the transfer operator where the prolongated solution is corrected on each coarse grid cell. We define subspaces $\tilde{V}_{h,j} = \{\mathbf{v}_h \in V_h : \mathrm{supp}(\mathbf{v}_h) \subset K_j\} \subset V_h$ for each coarse mesh element $K_j \in \mathcal{T}_H$ and the corresponding restriction operators $\tilde{\mathcal{R}}_{h,j} : V_h \to \tilde{V}_{h,j}$ onto those patches. The AL-Prolongation $\tilde{\mathbf{P}}_H^h : V_H \to V_h$ uses the natural prolongation $\mathbf{P}_H^h$ of the FE-space and then, executes a

correction on each subspace by solving a restricted system. Algebraically, this operation is defined by

$$\tilde{\mathbf{P}}_H^h = \mathbf{P}_H^h - \gamma \sum_j \left(\tilde{\mathcal{R}}_{h,j}^\top \left(\tilde{\mathcal{R}}_{h,j} A_{h,\gamma} \tilde{\mathcal{R}}_{h,j}^\top \right)^{-1} \tilde{\mathcal{R}}_{h,j} \right) \mathbf{B}_h \mathbf{M}_{h,p}^{-1} \mathbf{B}_h^\top \mathbf{P}_H^h.$$

As usual, the restriction $\tilde{\mathbf{R}}_h^H = (\tilde{\mathbf{P}}_H^h)^\top$ is the transposed prolongation.

3.2.1 Space-time multigrid

The STMG method [16] uses semi-coarsening in time and line smoothing in space. The primary level hierarchy is constructed by refinement in time only. For DG discretizations, the prolongation operators are given by the natural injection $\mathbf{P}_{2\tau}^\tau : V_{2\tau,q} \to V_{\tau,q}$ between the nested spaces $V_{2\tau,q} \subset V_{\tau,q}$. The restriction is the adjoint operator $\mathbf{R}_\tau^{2\tau} = (\mathbf{P}_{2\tau}^\tau)^\top$. As the same spatial grids are used on each level, the global transfers are given by

$$\mathbf{P}_{2\tau,h}^{\tau,h} = \mathbf{P}_{2\tau}^\tau \otimes \mathbf{I}_{n_v} \text{ and } \mathbf{R}_{\tau,h}^{2\tau,h} = \mathbf{R}_\tau^{2\tau} \otimes \mathbf{I}_{n_v}.$$

The smoother is a damped block Jacobi iteration

$$\mathbf{u}_K \mapsto \mathbf{u}_K + \omega \mathcal{S}(\mathbf{A}_K)^{-1} (\mathbf{f}_K - \mathbf{A}_K \mathbf{u}_K) \text{ with } \mathcal{S}(\mathbf{A}_K) = (\mathbf{I}_K \otimes \mathbb{1}_s \otimes \mathbb{1}_{n_v}) \circ \mathbf{A}_K$$

using the Hadamard product $(A \circ B)_{ij} = a_{ij} b_{ij}$ and the matrix $\mathbb{1}_N \in \mathbb{R}^{N \times N}$ consisting only of ones. $\mathcal{S}(\mathbf{A}_K)$ is a block-diagonal approximation of $\mathbf{A}_K$ where each block corresponds to one time step and the coupling between time steps is ignored. Each block of $\mathbf{A}_K$ is approximated by a single multigrid V-cycle in space that uses the specialized smoothing and transfer operators. The damping parameter ω can be chosen as 0.5, but for small K a choice of ω closer to 1 can lead to faster convergence [4]. We determine the damping parameter by the means of an FGMRES(1) solver using $\mathcal{S}(\mathbf{A}_K)$ as preconditioner. When using a diagonal temporal mass matrix $\mathbf{M}_\tau$, it is also possible to use $\mathcal{S}(\mathbf{A}_K) = (\mathbf{I}_{n_K} \otimes \mathbb{1}_{n_v}) \circ \mathbf{A}_K$ where each block corresponds a single stage of a multistage time discretization instead of a complete time step. However, the STMG method does not converge for a Crank-Nicolson discretization, because the matrix $\mathbf{M}_\tau$ is not (block-)diagonal and the smoother does not capture all the contributions of the spatial stiffness matrices:

$$\mathcal{S}(\mathbf{M}_\tau \otimes \mathbf{S}_h) = (\mathbf{I}_{n_K} \circ \mathbf{M}_\tau) \otimes \mathbf{S}_h \neq \mathbf{M}_\tau \otimes \mathbf{S}_h.$$

3.2.2 Multigrid waveform relaxation

The multigrid waveform relaxation method [20], initially called Multigrid dynamic iteration in [25], uses semi-coarsening in space and line smoothing in time. Accordingly, the transfer operators are given by

$$\mathbf{P}_{\tau,H}^{\tau,h} = \mathbf{I}_{n_K} \otimes \tilde{\mathbf{P}}_H^h \text{ and } \mathbf{R}_{\tau,h}^{\tau,H} = \mathbf{I}_{n_K} \otimes \tilde{\mathbf{R}}_h^H$$

using the modified spatial transfers. The Jacobi waveform relaxation smoother is defined by

$$\mathbf{u}_K \mapsto \mathbf{u}_K + \omega \mathcal{S}(\mathbf{A}_K)^{-1} (\mathbf{f}_K - A_K \mathbf{u}_K) \text{ with } \mathcal{S}(\mathbf{A}_K) = (\mathbb{1}_{n_K} \otimes \mathbf{I}_{n_v}) \circ \mathbf{A}_K$$

where for each spatial DoF a forward sweep in time is done. This corresponds to solving an ODE for each spatial DoF. Instead of a Jacobi waveform smoother, we perform a Schwarz waveform relaxation using the star-patch decomposition. So for each patch a small ODE system is solved [23]. Algebraically this can be written as

$$\mathcal{S}_{AL}(\mathbf{A}_K) = \sum \mathcal{R}_{K,i}^{\top} \left(\mathcal{R}_{K,i} \mathbf{A}_{K,\gamma} \mathcal{R}_{K,i}^{\top} \right)^{-1} \mathcal{R}_{K,i}$$

using the time extended restriction operators $\mathcal{R}_{K,i} = \mathbf{I}_{n_K} \otimes \mathcal{R}_{h,i}$. While the forward solve in time is not easily parallelizable, the subproblems associated with the local patches can be solved independently. Overall, this results in less frequent inter-process communication and improved parallel scalability, even when parallelization is done only in space.

4 Numerical experiments

In the numerical tests, we use an inexact Newton's method and Eisenstat-Walker forcing [9] with $\alpha = 2$ and $\gamma = 0.9$ as the adaptive stopping criterion for the linear Oseen solver. The Newton method aborts successfully, when the unmodified non-linear residual $\mathbf{r}_K$ with $\gamma = 0$ is improved by a factor $tol_{rel} = 1e-6$ or until $\tau \mathbf{r}_K < 1e-9$. The inner solver is an FGMRES(4) iteration using the block-preconditioner P and the previously described Schur complement approximations. For a given test case, time discretization, and time step size, we always simulate the same number of time step, and only vary the number of simultaneously calculated time steps K. To solve the same time interval for each K we execute an outer macro time stepping loop. The reported number of iterations are always averaged across all macro time windows.

4.1 Flow around a cylinder benchmark

To evaluate the presented solver, we first consider the unsteady flow around a cylinder benchmark [29]. In this test case, the domain is a rectangular channel with a cylindrical obstacle while a parabolic inflow is prescribed at the left boundary (see Figure 1). The domain is triangulated with 64 quadrilateral elements on mesh level 0. Each finer mesh level $i + 1$ is generated by a uniform refinement of the previous mesh level i. The simulated time interval is $(0, 8)$ and a uniform time step size τ is used. The spatial multigrid uses the mesh on level 0 as the coarse level whereas in the time multigrid methods the time mesh is always coarsened until a single time step is left. All multigrid methods use a V-cycle.

The inflow is defined as

$$u(t,x) = \sin\left(\frac{1}{8}\pi t\right) u_{\max} \frac{x_1(0.41 - x_1)}{4 \cdot 0.41^2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

with $u_{\max} = 1.5$ The viscosity $\nu = 0.001$ results in the maximum Reynolds number of $Re = 100$. We also use the modified problem with $u_{max} = 0.3$ and $Re = 20$. While the case with $Re = 100$ exhibits vortex shedding, the case with $Re = 20$ remains laminar, but still incorporates transient effects.

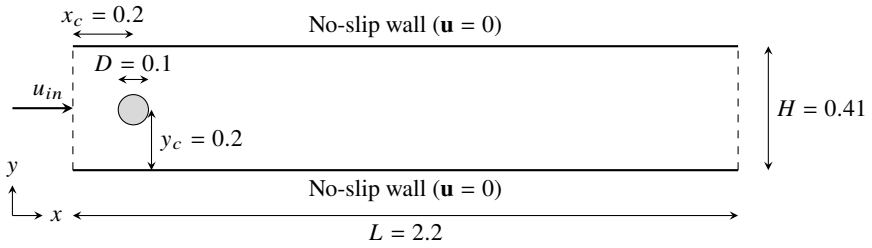
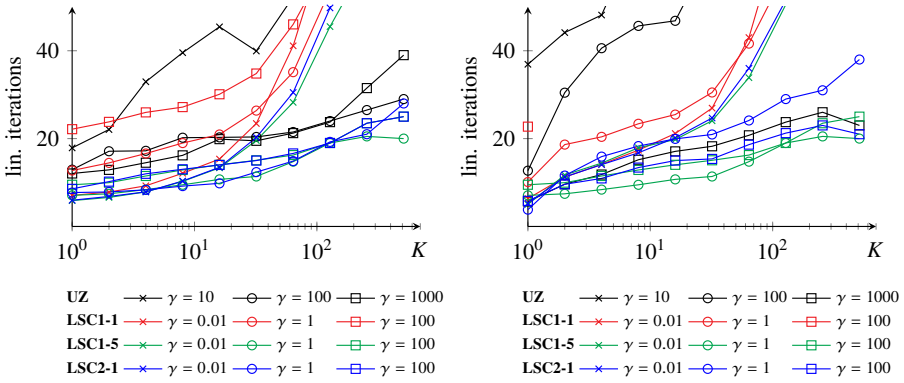


Figure 1: Domain: flow around a cylinder



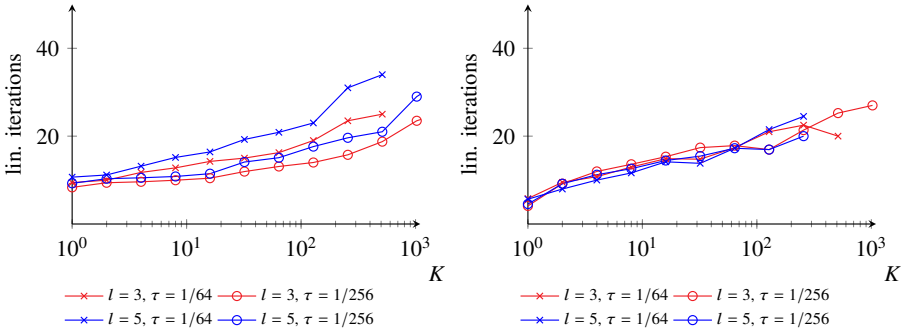
(a) WRMG, CN

(b) STMG, DG(1)

Figure 2: $Re = 20$, Number of linear iterations: average over the time domain, Level 3, $\tau = 1/64$

Figure 2 shows the number of linear iterations for different Schur complement preconditioning strategies and AL parameters γ and a varying number of simultaneously computed time steps K for a Reynolds number of $Re = 20$. The different variants of the LSC1 preconditioner, denoted by LSC1-1 and LSC1-5, use one or five multigrid steps to solve the pressure Laplacian, illustrating the influence of the pressure solver accuracy. For a sequential solver and $K = 1$ all preconditioning strategies are viable. However, as K increases, the methods with a small values of γ require more iterations and fail to converge on long time frames. This shows the necessity of AL stabilization even when the LSC preconditioner is used. There is only a minor difference between the waveform relaxation and the space-time multigrid method. The Uzawa preconditioner works for $\gamma = 1000$, but the solver fails to converge for even larger γ due to ill-conditioning of the system. Thus, γ must be chosen carefully, but it is unclear how to predict the optimal choice of γ and whether a viable choice even exists for different discretizations and problems.

The LSC2 preconditioning variant exhibits a stable convergence behavior with respect to K for $\gamma = 1$ and $\gamma = 100$. In the case of $\gamma = 1$, the number of Newton iterations increases from 1.88 ($K = 1$) to 4 ($K = 512$). Since each nonlinear iteration



(a) WRMG, CN

(b) STMG, DG(1)

Figure 3: Number of linear iterations: avg. over time domain, LSC-Al preconditioner, $\gamma = 100$

takes an average of ≈ 4 inner linear iterations to reach the desired tolerance, the total number of linear iteration increases from 7.1 to 20.5. The increase of nonlinear iterations can be explained by the less accurate initial guesses because a constant extrapolation of the last calculated time step is used. This is a good initial approximation for the next time step but not for subsequent ones. Using a coarse grid approximation in a full multigrid algorithm may improve this behavior [8].

The LSC1 formulation, which uses one pressure multigrid iteration, fails to converge for large K. However, it performs as well as the LSC2 variant when the subproblems are solved to a higher accuracy using multiple multigrid steps. Nevertheless, the LSC2 variant is preferable as it only needs one multigrid iteration. Figure 3 shows that different time step and mesh sizes only slightly affect the number of iterations.

Table 1 shows the number of iterations for the benchmark for $u_{\max} = 1.5$ using the LSC2-1 preconditioner. The alternating Picard-Newton method requires only 2 to 2.5 iteration in a time-stepping scheme. However, this number increases to ≈ 8 when time frame of length 1 is solved simultaneously. Furthermore, the number of inner linear iterations increases more than tenfold showing that also the convergence of the linear solver worsens. Increasing the number of time steps K further leads to the failure of all methods.

The failure is caused by the velocity systems that cannot be solved with a global-in-time multigrid, because the reactive term is not compensated by the mass matrix from the time derivative. For instance, the space-time multigrid method fails on the coarse meshes, as the inner spatial multigrid method diverges due to the large time step and the strong convective and reactive terms. Although WRMG does not solve on a coarse time mesh, convection dominated problems cannot be efficiently solved by this method either (c.f. [7]).

τ^{-1}	K	1	4	16	64	128	256	512
256	ST	2.00	2.44	3.13	4.94	6.00	8.50	-
256	WR	2.15	2.28	3.24	4.84	5.88	8.00	-
64	ST	2.42	3.13	4.97	8.38	-	-	-
64	WR	2.46	3.38	5.00	8.13	-	-	-

(a) nonlinear iterations

τ^{-1}	K	1	4	16	64	128	256	512
256	ST	5.1	12.5	16.6	27.2	41.1	103.4	-
256	WR	8.1	9.0	15.5	30.7	53.4	144.6	-
64	ST	7.0	12.4	23.5	90.3	-	-	-
64	WR	10.7	16.8	31.5	125.4	-	-	-

(b) linear iterations

Table 1: Average number of iterations: Re=100, Alternating Picard Newton, LSC2-1, level 3

4.2 Carreau test case

The second test case uses the same spatial domain, the time interval $(0, 4)$, the inflow condition

$$u(t,x) = (0.2 + 0.1\cos(\pi t))\frac{x_1(0.41 - x_1)}{4 \cdot 0.41^2}\begin{pmatrix}1\\0\end{pmatrix} \tag{7}$$

and a Carreau fluid with $\mu_0 = 0.01$, $\mu_\infty = 0$, $\lambda = 1$ and $n = 0.31$. This test case is diffusion dominated, yet it exhibits nonlinear behavior due to the rheology and strong transient effects. We start with the solution of the stationary problem at $t = 0$ and simulate two periods. For long time frames, Newton's method fails to converge, so globalization strategies are necessary. Here, we choose the alternating Picard-Newton iteration which has been shown to converge globally and quadratically for the Navier-Stokes equations [28].

The Uzawa preconditioner (6) uses the scaling factor $\nu = \mu_0$. For the choice of $\gamma = 1$, the results are slightly worse than in the Newtonian test case, but for $\gamma = 100$ the overall solver behavior resembles that of the Newtonian fluid (see Table 2). Again, the LSC1 preconditioner needs a more exact solver for the pressure Laplacian and the Uzawa preconditioner needs higher AL parameters γ. The increase of the total number of linear iterations is caused by the increase of nonlinear iterations. For example the LSC2-1 preconditioner shows an increase from 8.5 to 26 linear iterations per macro time step and the number of nonlinear iterations increases from 2.125 to 6. The number of linear iterations per nonlinear iteration remains between 3.9 and 4.5 for all considered values of K.

K	1	4	16	64	256	512
LSC1-1, $\gamma = 100$	24.2	56.8	–	–	–	–
LSC1-5, $\gamma = 100$	8.0	14.5	21.8	30.0	32.0	32.0
LSC2-1, $\gamma = 100$	8.0	16.7	22.5	31.5	32.0	32.0
Uzawa, $\gamma = 1000$	1.3	17.7	25.8	36.0	38.0	36.0

(a) STMG

K	1	4	16	64	256	512
LSC1-1, $\gamma = 100$	30.3	44.5	–	–	–	–
LSC1-5, $\gamma = 100$	8.7	10.8	16.9	27.1	26.0	26.0
LSC2-1, $\gamma = 100$	8.4	10.9	16.9	27.3	26.0	26.0
Uzawa, $\gamma = 1000$	11.8	14.0	19.9	30.3	31.5	32.0

(b) WRMG

Table 2: Carreau test case: Number of linear iterations, $\tau = 1/128$, level 3

4.3 Strong scaling

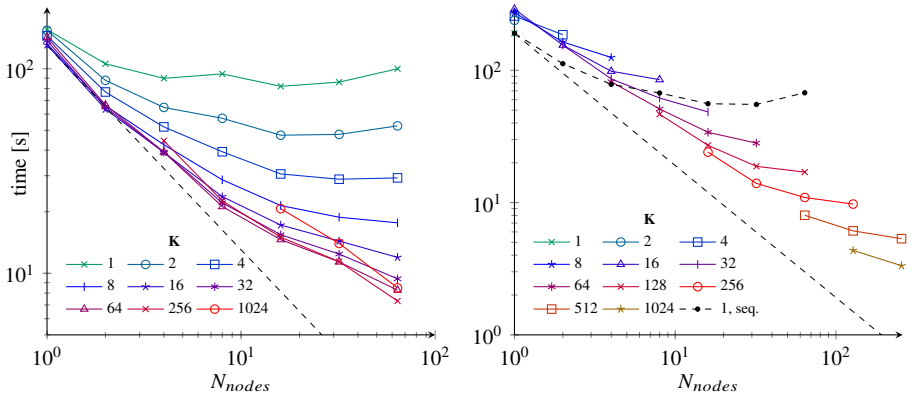
To demonstrate the potential speedup of these methods we now consider a strong scaling test executed on the LiDO3 cluster. The cluster has 316 nodes, each equipped with two Intel Xeon E5-2640v4 CPUs and 64GB of DDR4 memory, interconnected by a QDR-40 Infiniband network. The test case uses a Newtonian fluid and $Re = 20$ on the spatial refinement level 4 consisting of 16384 mesh elements.

Figure 4 presents the solver time normalized by the average number of linear iterations. The solver time per iteration is similar for Carreau fluids, because both test cases lead to structurally identical matrices. Multiplying these times by the average number of linear iterations yields a good estimate of the total solver time.

Due to the increased efficiency of the batched pressure solves, multigrid waveform relaxation profits from solving multiple time steps at once, even on a single node. However, the WRMG method with $K = 1$ should not be taken as a benchmark for a sequential solver, because the star-patch smoother for a single time step can be implemented more efficiently by computing the inverse of each local patch matrix and assembling a single preconditioning matrix. A comparison against a reasonable sequential solver is shown in Figure 5.

Solving more time steps at once initially improves the speedup, as it decreases the latency induced communication overhead. Nonetheless, solving more than 64 time steps has less significant effects, as the overhead is mostly hidden. Comparing the lowest solver times, we see a speedup factor of 17.85 when comparing $K = 1$ on 16 nodes with $K = 1024$ on 64 nodes. Both of these data points have a low parallel efficiency, but the WRMG shows nearly ideal strong scaling much longer than a sequential method with $K = 1$. However, for the parallel efficiency we have to consider a better suited time stepping implementation, and we have to take into account the increase of the number of iterations.

The space-time multigrid method has a higher initial overhead compared to a true



(a) WRMG, CN

(b) STMG, DG(1)

Figure 4: Solver time [s] per iteration

time stepping implementation, but shows good scaling behavior (see Figure 4b). In this test we use spatial parallelization on each node and time parallelization between nodes. When K is increased we also increase the level depth of the time multigrid scheme. Each time level uses as many resources as possible, so that every process on each level handles 2 time steps if enough processes are available. Here, we need at least two time steps per process, because the coarsening is done on the fine level.

The scaling is not perfect due to the increasing level depth and the global reduction operations that do not scale perfectly. Both factors lead to a logarithmic factor, resulting in the expected strong scaling behavior of $O(\frac{\log_2(K)}{K})$ when using one process per time step. Therefore, the parallel efficiency decreases as more time steps are used. Nevertheless, the method achieves a 16.6x speedup per iteration. The total wall time decreases from 1550 seconds for the sequential method on one node to 124.6 seconds on 128 nodes. These times could be equally improved by using more spatial parallelization.

To compare the two methods with a sequential time stepping, we use the backward Euler discretization. Figure 5 shows the wall time of these tests. The WRMG has less overhead than STMG. On the other hand the overhead compared to sequential time stepping is significant. The graph shows that, as long as it scales, spatial parallelization is the fastest method. In summary, multigrid waveform relaxation reduces the overall wall time by solving multiple time steps simultaneously and utilizing more parallel resources. Space-time multigrid methods achieve even better scaling but suffer from a higher overhead.

5 Conclusion

We have presented a global-in-time solution strategy for the Navier-Stokes equation for low Reynolds numbers. This strategy uses a global-in-time linearization and augmented Lagrangian preconditioning which can be combined with an LSC preconditioner. With

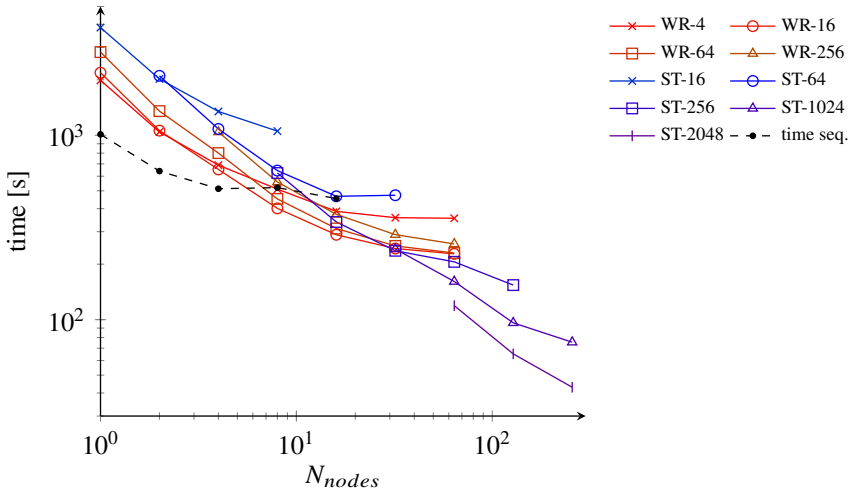


Figure 5: Total wall time, Backward Euler, $\tau = 1/256$

a carefully chosen time discretization, the resulting pressure problems are embarrassingly parallel and can utilize batched linear algebra and block-Krylov methods. The velocity problems need to be solved in space and time. For doings so, multigrid waveform relaxation and space-time multigrid methods are viable strategies as long as the convection and reaction terms are not dominant.

This method is also well-suited to Carreau fluids. Parallel scaling tests show a wall time reduction by a factor of 12.5, but the parallel efficiency is only 9.7% when comparing these data points. Due to the need for specialized smoothing strategies for the augmented Lagrangian problem, the multigrid waveform relaxation can only achieve a small improvement compared to the time stepping method and is outperformed by the space-time multigrid method, when more resources are used. Therefore, it is only viable in a very small regime. A time parallel implementation of the Schwarz waveform relaxation smoother using cyclic reduction [33] can improve this behavior. Currently it is unclear, how this would compare to the space-time multigrid method.

Acknowledgements This work was supported by the Federal Ministry of Research, Technology and Space (BMFTR) through the project "StroemungsRaum" 16ME0706K, which is part of the initiative "Neue Methoden und Technologien für das Exascale-Höchstleistungsrechnen" (SCALEXA).

The authors gratefully acknowledge the computing time provided on the Linux HPC cluster at Technical University Dortmund (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359.

References

[1] P. Benedusi et al. “A Parallel Multigrid Solver for Time-Periodic Incompressible Navier–Stokes Equations in 3D”. In: *Numerical Mathematics and Advanced Applications ENUMATH 2015*. Ed. by B. Karasözen et al. Cham: Springer International Publishing, 2016, pp. 265–273.

[2] M. Benzi and M. A. Olshanskii. “An Augmented Lagrangian-Based Approach to the Oseen Problem”. In: *SIAM Journal on Scientific Computing* 28.6 (2006), pp. 2095–2113.

[3] P. J. Carreau. “Rheological Equations from Molecular Network Theories”. In: *Transactions of the Society of Rheology* 16.1 (1972), pp. 99–127.

[4] B. Chaudet-Dumas, M. Gander, and A. Pogozelskyte. *An optimized Space-Time Multigrid algorithm for parabolic PDEs*. Feb. 2023.

[5] R. Croce, D. Ruprecht, and R. Krause. “Parallel-in-Space-and-Time Simulation of the Three-Dimensional, Unsteady Navier-Stokes Equations for Incompressible Flow”. In: *Modeling, Simulation and Optimization of Complex Processes – HPSC 2012*. Ed. by H. G. Bock et al. Springer International Publishing, 2014, pp. 13–23.

[6] F. Danieli, B. S. Southworth, and A. J. Wathen. “Space-Time Block Preconditioning for Incompressible Flow”. In: *SIAM Journal on Scientific Computing* 44.1 (2022), A337–A363.

[7] W. Drews, S. Turek, and C. Lohmann. “Improving Convergence of Time–Simultaneous Multigrid Methods for Convection–Dominated Problems Using VMS Stabilization Techniques”. In: *Numerical Mathematics and Advanced Applications*. Ed. by M. Griebel et al. Springer, 2025, pp. 278–286.

[8] J. Dünnebacke et al. “Increased space–parallelism via time–simultaneous Newton–multigrid methods for nonstationary nonlinear PDE problems”. In: *International Journal of High Performance Computing Applications* online (2021).

[9] S. C. Eisenstat and H. F. Walker. “Choosing the Forcing Terms in an Inexact Newton Method”. In: *SIAM Journal on Scientific Computing* 17.1 (1996), pp. 16–32.

[10] H. C. Elman. “Preconditioning for the Steady-State Navier–Stokes Equations with Low Viscosity”. In: *SIAM Journal on Scientific Computing* 20.4 (1999), pp. 1299–1316.

[11] H. C. Elman et al. “Block Preconditioners Based on Approximate Commutators”. In: *SIAM Journal on Scientific Computing* 27.5 (2006), pp. 1651–1668.

[12] R. D. Falgout et al. *Parallel Time Integration with Multigrid Reduction for a Compressible Fluid Dynamics Application*. Tech. rep. Lawrence Livermore National Laboratory, 2014.

[13] P. E. Farrell, L. Mitchell, and F. Wechsung. “An Augmented Lagrangian Preconditioner for the 3D Stationary Incompressible Navier–Stokes Equations at High Reynolds Number”. In: *SIAM Journal on Scientific Computing* 41.5 (2019), A3073–A3096.

[14] P. F. Fischer, F. Hecht, and Y. Maday. “A parareal in time semi-implicit approximation of the Navier-Stokes equations”. In: *Domain Decomposition Methods in Science and Engineering*. Ed. by R. Kornhuber and et al. Vol. 40. Lecture Notes in Computational Science and Engineering. Berlin: Springer, 2005, pp. 433–440.

[15] M. Fortin and R. Glowinski. *Augmented Lagrangian methods : applications to the numerical solution of boundary-value problems*. Amsterdam, New York, New York, N.Y: Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co, 1983.

[16] M. Gander and M. Neumüller. “Analysis of a New Space-Time Parallel Multigrid Algorithm for Parabolic Problems”. In: *SIAM Journal on Scientific Computing* 38 (July 2016), A2173–A2208.

[17] M. J. Gander. “50 Years of Time Parallel Time Integration”. In: *Multiple Shooting and Time Domain Decomposition Methods*. Ed. by T. Carraro et al. Cham: Springer International Publishing, 2015, pp. 69–113.

[18] H. T. Huynh. “Discontinuous Galerkin and Related Methods for ODE”. In: *Journal of Scientific Computing* 96.51 (2023).

[19] J. Jackaman and S. MacLachlan. *Space-time waveform relaxation multigrid for Navier-Stokes*. 2026.

[20] J. Janssen and S. Vandewalle. “Multigrid Waveform Relaxation on Spatial Finite Element Meshes: The Discrete-Time Case”. In: *SIAM Journal on Scientific Computing* 17.1 (1996), pp. 133–155.

[21] G. L. Kooij, M. A. Botchev, and B. J. Geurts. “An Exponential Time Integrator for the Incompressible Navier–Stokes Equation”. In: *SIAM Journal on Scientific Computing* 40.3 (2018), B684–B705.

[22] J.-L. Lions, Y. Maday, and G. Turinici. “Résolution d’EDP par un schéma en temps «pararéel»”. In: *Comptes Rendus de l’Académie des Sciences - Series I - Mathematics* 332.7 (2001), pp. 661–668.

[23] C. Lohmann and S. Turek. “Augmented Lagrangian Acceleration of Global–in–Time Pressure Schur Complement Solvers for Incompressible Oseen Equations”. In: *Journal of Mathematical Fluid Mechanics* 26.27 (2024).

[24] C. Lohmann and S. Turek. “On the Design of Global–in–Time Newton–Multigrid–Pressure Schur Complement Solvers for Incompressible Flow Problems”. In: *Journal of Mathematical Fluid Mechanics* 25.64 (2023).

[25] C. Lubich and A. Ostermann. “Multi-grid dynamic iteration for parabolic equations”. In: *BIT Numerical Mathematics* 27 (1987), pp. 216–234.

[26] B. W. Ong and J. B. Schroder. “Applications of time parallelization”. In: *Computing and Visualization in Science* 23.11 (Sept. 2020).

[27] W. Pazner and P.-O. Persson. “Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations”. In: *Journal of Computational Physics* 335 (2017), pp. 700–717.

[28] S. Pollock et al. “Analysis of the Picard-Newton Iteration for the Navier-Stokes Equations: Global Stability and Quadratic Convergence”. In: *Journal of Scientific Computing* 104.25 (2025).

[29] M. Schäfer et al. “Benchmark Computations of Laminar Flow Around a Cylinder”. In: *Flow Simulation with High-Performance Computers II: DFG Priority Research Programme Results 1993–1995*. Ed. by E. H. Hirschel. Wiesbaden: Vieweg+Teubner Verlag, 1996, pp. 547–566.

[30] Y.-h. Shih, G. Stadler, and F. Wechsung. “Robust Multigrid Techniques for Augmented Lagrangian Preconditioning of Incompressible Stokes Equations with Extreme Viscosity Variations”. In: *SIAM Journal on Scientific Computing* 45.3 (2023), pp. 27–53.

[31] J. Steiner et al. “Convergence of Parareal for the Navier-Stokes equations depending on the Reynolds number”. In: *Numerical Mathematics and Advanced Applications - ENUMATH 2013*. Ed. by A. Abdulle et al. Vol. 103. Lecture Notes in Computational Science and Engineering. Springer International Publishing, 2015, pp. 195–202.

[32] J. M. F. Trindade and J. C. F. Pereira. “Parallel-in-time simulation of the unsteady Navier-Stokes equations for incompressible flow”. In: *International Journal for Numerical Methods in Fluids* 45.10 (2004), pp. 1123–1136.

[33] S. Vandewalle and E. Velde. “Space-time Concurrent Multigrid Waveform Relaxation”. In: *Annals of Numerical Mathematics* 1.1-4 (1994), pp. 347–363.

[34] F. Wechsung. “Shape optimisation and robust solvers for incompressible flow”. PhD thesis. University of Oxford, 2019.