acrt AI, Computer Science and Robotics Technology

IntechOpen Journals

RESEARCH PAPER

Numerical Analysis of a Time-Simultaneous Multigrid Solver for Stabilized Convection-Dominated Transport Problems in 1D

Wiebke Drews*, Stefan Turek and Christoph Lohmann

Institute of Applied Mathematics (LSIII), TU Dortmund University, Vogelpothsweg 87, Germany
*Corresponding author. E-mail: wiebke.drews@math.tu-dortmund.de

Abstract

This work focuses on the solution of the convection–diffusion equation, especially for small diffusion coefficients, employing a time-simultaneous multigrid algorithm, which is closely related to multigrid waveform relaxation. For discretization purposes, linear finite elements are used while the Crank–Nicolson scheme acts as the time integrator. By combining all time steps into a global linear system of equations and rearranging the degrees of freedom, a space-only problem is formed with vector-valued unknowns for each spatial node. The generalized minimal residual method with block Jacobi preconditioning can be used to numerically solve the (spatial) problem, allowing a higher degree of parallelization in space. A time-simultaneous multigrid approach is applied, utilizing space-only coarsening and the aforementioned solution techniques for smoothing purposes. Numerical studies analyze the iterative solution technique for 1D test problems. For the heat equation, the number of iterations stays bounded independently of the number of time steps, the time increment, and the spatial resolution. However, convergence issues arise in situations where the diffusion coefficient is small compared to the grid size and the magnitude of the velocity field. Therefore, a higher-order variational multiscale stabilization is used to improve the convergence behavior and solution smoothness without compromising its accuracy in convection-dominated scenarios.

Citation
Wiebke Drews, Stefan Turek and Christoph Lohmann (2024), Numerical Analysis of a Time-Simultaneous Multigrid Solver for Stabilized Convection-Dominated Transport Problems in 1D. *AI, Computer Science and Robotics Technology* 3(1), 1–28.

DOI
https://doi.org/10.5772/acrt.37

Received: 17 October 2023
Accepted: 10 July 2024
Published: 25 July 2024

Keywords: Convection-diffusion equations, multigrid waveform relaxation, parallel-in-time methods, stabilization techniques, variational multiscale methods

1. Introduction

The interest in massively parallel computing has grown rapidly due to the ever-increasing number of processors, or cores, in modern hardware architecture. To fully exploit the potential of these computers and actually reduce the runtimes of applications, it is essential to develop algorithms whose computational tasks can be performed in parallel on the different processors. More than 50 years ago, the first investigations on parallel-in-time methods for the numerical solution of time-dependent partial differential equations were published. While initial value problems are typically solved numerically using methods that operate sequentially in time, the new algorithms solve the problem for all time steps simultaneously, providing increased parallelization capabilities that are otherwise limited by spatial resolution. A general introduction and an overview of parallel-in-time methods can be found in [1, 2]. These methods can be broadly categorized into different approaches. One approach is to parallelize in time by decomposing the time interval into independent subintervals or time steps. Related methods include parareal [3], multigrid reduction-in-time (MGRIT) [4], and the space–time parallel multigrid algorithm published in [5], which is a multigrid method in time. Another approach involves algorithms for global space–time systems that do not necessarily parallelize in time but consider the aspect of data communication for performance benefits. An example is the multigrid waveform relaxation (WRMG) method developed by Lubich and Ostermann [6], which treats time steps simultaneously and parallelizes in space. This solution strategy is characterized by the fact that it is applied to the evolution equation before it is discretized in time. Recently, this approach was implemented in a different way in [7] and referred to as a time-simultaneous multigrid method. Starting from a sequential problem already discretized in space and time, a global system of equations is set up in which all time steps are blocked so that it can be interpreted as a space-only problem for vector-valued unknowns. A geometric multigrid method with block Jacobi smoothing is applied to this linear system of equations and is designed to be highly parallelizable. Since all time steps are treated simultaneously, the problem size is increased for each spatial node, and the iterative multigrid solver allows parallelization in space to fully exploit the potential of multiple cores. As also reported in [7], the runtimes of this approach were compared with those of the sequential time-stepping method for the 2D heat equation. Strong scaling tests show that the runtime can be significantly reduced by the time-simultaneous multigrid method, achieving a substantial speedup over the sequential approach. Further analysis of the parallel performance of the WRMG method can be found in [8, 9]. Additionally, there are some theoretical convergence results for the WRMG method. For example, Janssen and Vandewalle [10] proved that the asymptotic rate of convergence of the solver is the same as in the time-stepping approach if the problem is discretized in time using linear multistep methods. In another theoretical investigation, bounds on the spectral norm were derived using related circular matrices [11]. Furthermore, a Fourier analysis was exploited to analyze the time-simultaneous two-grid algorithm using a damped Jacobi (waveform relaxation) smoother. In the case of the one-dimensional heat equation on a uniform grid, it was shown that the spectral norm of the iteration matrix is uniformly bounded if the one-step theta scheme is used for time integration [12].

The application of many parallel-in-time methods to convection-dominated transport problems presents difficulties with respect to the parallel efficiency of the solver. Studies on higher-order hyperbolic problems and corresponding limitations were published in [13] for time-decomposed parallel time integrators and for parareal, e.g., in [14]. Recent studies on the MGRIT algorithm applied to constant-wave-speed linear advection problems with an alternative coarse-grid operator show fast solver convergence for various method-of-lines discretizations and a speedup compared to the sequential time-stepping method [15, 16]. Moreover, optimized transmission conditions for the Schwarz waveform relaxation have been studied for the scope of convection–diffusion problems in, e.g., [17, 18].

Considering these convection-dominated transport problems, it is additionally well known from the discretization side that the Galerkin finite element (FE) solution is polluted by spurious artifacts. To reduce these oscillations, various stabilization techniques are proposed in the literature; e.g., see [19, Section 12.8]. These include strongly consistent methods (such as GLS, SUPG), the introduction of artificial diffusion, or a decentralized discretization of the convection term based on a Petrov–Galerkin approach.

The stabilization to be considered in this work is a (fully implicit) variational multiscale (VMS) type method, which was originally proposed by [20] and is adapted in [21] and [22]. Modification of the variational form of the underlying problem by adding a diffusive term and removing low-frequency diffusion within the VMS context can improve the accuracy of the numerical solution compared to the one based on an artificial diffusion stabilization. In our context, however, the stabilization technique is used not only for accuracy reasons but also for the improvement of the time-simultaneous multigrid solver under consideration. The main idea for its use is to perturb the system by higher-order diffusion for preservation of high accuracy, but better convergence behavior.

In this paper, the mentioned time-simultaneous multigrid algorithm is first investigated for the d-dimensional convection–diffusion equation in Section 2.

In addition to the special case of the heat equation, our numerical studies focus on convection-dominated problems in 1D. As the second part of this work, we introduce a higher-order VMS-type stabilization technique and numerically study this strategy in 1D in combination with the introduced time-simultaneous method in Section 3. Finally, Section 4 summarizes the results and offers considerations for future research.

2. *Time-simultaneous multigrid method*

The algorithm to be presented in this chapter is a geometric multigrid method in space and aims to be a highly parallelizable solution strategy to numerically solve the unsteady convection–diffusion equation. For this purpose, all time steps are considered simultaneously for each spatial grid point, allowing parallelization in space in a straightforward manner. We first introduce the spatial discretization of the d-dimensional problem under consideration and then formulate the linear system of equations to be solved by the time-simultaneous method. Afterwards, the time-simultaneous multigrid algorithm, which is closely related to the WRMG [6, 10], is introduced. In numerical studies, the performance of the solver is analyzed in one spatial dimension to quickly obtain representative results.

2.1. Discretization

We consider the d-dimensional convection–diffusion problem: Find $u : \Omega \times (0, T) \to \mathbb{R}$ such that

$$\begin{aligned} \partial_t u(x,t) - \varepsilon \Delta u(x,t) + \boldsymbol{v}(x,t) \cdot \nabla u(x,t) &= f(x,t) && (x,t) \in \Omega \times (0,T) \\ u(x,t) &= 0 && (x,t) \in \partial\Omega \times (0,T) \\ u(x,0) &= u^0(x) && x \in \Omega, \end{aligned} \tag{1}$$

where $T > 0$ denotes the final time and $\Omega \subset \mathbb{R}^d$, $d \in \{1,2,3\}$, is the spatial domain on whose boundary $\partial\Omega$ homogeneous Dirichlet boundary values are imposed for simplicity. The velocity field and the right-hand side are given by $\boldsymbol{v} : \Omega \times (0,T) \to \mathbb{R}^d$ and $f : \Omega \times (0,T) \to \mathbb{R}$, while $\varepsilon \geq 0$ is a constant diffusion coefficient.

Let $(\cdot, \cdot)$ denote the $L^2(\Omega)$-inner product. The solution u: $(0, T) \to V$ of the variational formulation for (1) satisfies the initial condition $u\ (0,\ \boldsymbol{x}) = u^0(\ \boldsymbol{x}) \in V = H^1(\Omega)$ and

$$(\partial_t u, \varphi) + \varepsilon(\nabla u, \nabla\varphi) + (\boldsymbol{v} \cdot \nabla u, \varphi) = (f, \varphi) \quad \forall \varphi \in V. \tag{2}$$

Problem (2) is discretized using the subspace $V_h \subset V$ of linear FEs defined on the triangulation $\mathcal{T}_h$. Then $u_h\colon (0, T) \to V_h$ satisfies

$$(\partial_t u_h, \varphi_h) + \varepsilon(\nabla u_h, \nabla \varphi_h) + (\boldsymbol{v} \cdot \nabla u_h, \varphi_h) = (f, \varphi_h) \quad \forall \varphi_h \in V_h, \tag{3}$$

leading to the semi-discrete formulation in matrix form

$$\boldsymbol{M}_h \partial_t \boldsymbol{u}_h(t) + \varepsilon \boldsymbol{L}_h \boldsymbol{u}_h(t) + \boldsymbol{K}_h \boldsymbol{u}_h(t) = \boldsymbol{f}_h(t), \tag{4}$$

where $\boldsymbol{M}_h, \boldsymbol{L}_h, \boldsymbol{K}_h \in \mathbb{R}^{N\times N}$ are the mass, diffusion, and convection matrices, respectively. The discretized right-hand side is given by $\boldsymbol{f}_h \in \mathbb{R}^N$ for $N \in \mathbb{N}$ spatial degrees of freedom. All occurring integrals are approximated using the trapezoidal rule. This leads to a diagonal mass matrix $\boldsymbol{M}_h$, and problem (4) is equivalent to the well-known second-order finite difference (FD) discretization of (1) in the case of an equidistant triangulation in one dimension and a constant velocity field.

Discretization in time using the Crank–Nicolson (CN) scheme results in the discrete sequential form

$$\begin{gathered} \boldsymbol{A}_I \boldsymbol{u}_h^m + \boldsymbol{A}_E \boldsymbol{u}_h^{m-1} = \boldsymbol{f}^m, \quad m = 1, \dots, K \\ \text{for } \boldsymbol{A}_I := \boldsymbol{M}_h + \tfrac{1}{2}\Delta t(\varepsilon \boldsymbol{L}_h + \boldsymbol{K}_h), \quad \boldsymbol{A}_E := -\boldsymbol{M}_h + \tfrac{1}{2}\Delta t(\varepsilon \boldsymbol{L}_h + \boldsymbol{K}_h), \quad \boldsymbol{f}^m := \tfrac{1}{2}\Delta t\left(\boldsymbol{f}_h^m + \boldsymbol{f}_h^{m-1}\right), \end{gathered} \tag{5}$$

where $\boldsymbol{u}_h^0 \in \mathbb{R}^N$ is a suitable approximation of u^0, Δt is the time step size, and $K \in \mathbb{N}$ denotes the number of time steps. The solution technique described in the following can be applied similarly to non-equidistant grids and to other time-stepping methods like the implicit Euler method or Runge–Kutta methods.

So far, common discretization techniques have been presented. We now use algebraic transformations to obtain a global system matrix with a specific structure and construct the time-simultaneous multigrid method. Considering equation (5), using $\boldsymbol{u}^m$ as a shorthand notation for $\boldsymbol{u}_h^m$, and combining all K time steps in a global linear system of equations result in

$$\underbrace{\begin{pmatrix} \boldsymbol{A}_I & & & \\ \boldsymbol{A}_E & \boldsymbol{A}_I & & \\ & \ddots & \ddots & \\ & & \boldsymbol{A}_E & \boldsymbol{A}_I \end{pmatrix}}_{\in \mathbb{R}^{NK\times NK}} \underbrace{\begin{pmatrix} \boldsymbol{u}^1 \\ \boldsymbol{u}^2 \\ \vdots \\ \boldsymbol{u}^K \end{pmatrix}}_{\in \mathbb{R}^{NK}} = \underbrace{\begin{pmatrix} \boldsymbol{f}^1 - \boldsymbol{A}_E \boldsymbol{u}^0 \\ \boldsymbol{f}^2 \\ \vdots \\ \boldsymbol{f}^K \end{pmatrix}}_{\in \mathbb{R}^{NK}}. \tag{6}$$

Here, all degrees of freedom are sorted in a time-major ordering. The degrees of freedom are then rearranged so that all unknowns associated with one spatial node can be blocked into a macro degree of freedom as follows:

$$\begin{array}{rlr} & (u_1^1, u_2^1, \dots u_N^1, u_1^2, u_2^2, \dots, u_N^2, \dots, u_1^K, u_2^K, \dots, u_N^K)^\top & \text{time-major ordering} \\ & & \downarrow \\ \boldsymbol{u} := & (\underbrace{u_1^1, u_1^2 \dots, u_1^K}_{=:\boldsymbol{u}_1^\top}, \underbrace{u_2^1, u_2^2, \dots, u_2^K}_{=:\boldsymbol{u}_2^\top}, \dots, \underbrace{u_N^1, u_N^2, \dots, u_N^K}_{=:\boldsymbol{u}_N^\top})^\top & \text{space-major ordering} \end{array} \tag{7}$$

This results in the space-major ordering, where all K time steps are blocked so that the global vector of unknowns $\boldsymbol{u}$ can be interpreted as a block vector with vector-valued unknowns $\boldsymbol{u}_n$ for each spatial node $n = 1, \dots, N$. By doing the same for the system matrix as well as for the right-hand side, the new global system matrix $\boldsymbol{S}$ has the following block structure:

$$\begin{pmatrix} \square & \square & & & \\ \square & \square & \square & & \\ & \ddots & \ddots & \ddots & \\ & & \square & \square & \square \\ & & & \square & \square \end{pmatrix} \text{ with block matrix entries } \boxed{\begin{matrix} * & & & \\ * & * & & \\ & \ddots & \ddots & \\ & & * & * \end{matrix}} \in \mathbb{R}^{K \times K}. \tag{8}$$

The system matrix $\boldsymbol{S} := \boldsymbol{A}_I \otimes \boldsymbol{I}_K + \boldsymbol{A}_E \otimes \boldsymbol{I}_{K-1}$ can be written as a Kronecker product, where $\boldsymbol{I}_K, \boldsymbol{I}_{K-1} \in \mathbb{R}^{K \times K}$ are the identity matrix and a shifted matrix with only ones on the lower subdiagonal. The tridiagonal structure of $\boldsymbol{S}$ stems from the sparsity pattern of matrices $\boldsymbol{A}_I$ and $\boldsymbol{A}_E$ for the one-dimensional problem considered here for simplicity, while each block entry is a lower bidiagonal matrix due to the use of the Crank–Nicolson scheme for time integration. Therefore, it can be interpreted as a global space–time system with space-major ordering. Obviously, this new system matrix still has the same dimension $NK \times NK$ as the system matrix before. In what follows, we present a time-simultaneous solution algorithm for $\boldsymbol{Su} = \boldsymbol{f}$, which highly exploits the special structure of the system matrix.

2.2. Solution strategy

In this section, we apply a geometric multigrid method in space to the constructed system (8), which is designed to be highly parallelizable on modern hardware architectures. The iterative solver allows parallelization in space while all time steps are treated simultaneously for each spatial grid point.

In general, a multigrid solver is based on a hierarchy of mesh levels, which are used to reduce different modes of the error. More precisely, the idea of a two-grid algorithm is to start with a fine grid, where a smoother performs a number of smoothing steps to dampen highly oscillating error components. The remaining smooth part of the error is then approximated on a coarser grid and used to update the solution on the fine mesh. Performing this procedure iteratively results in a very efficient solution strategy if the coarse-grid problem is recursively approximated using the same technique [23, Section 5.1].

The time-simultaneous multigrid algorithm to be presented makes use of the same multigrid components, i.e., smoothing and coarse-grid correction, but is now applied to the space-only system (8) with vector-valued unknowns. For a more detailed description, the following sketch of the algorithm gives a brief overview of the ith iteration of the two-grid case, which can be extended to the multigrid case.

Time-simultaneous two-grid algorithm (ith iteration)

Let $\boldsymbol{x}^{(i)} \in \mathbb{R}^{NK}$ be a given initial guess.

1. **Pre-Smoothing:** $\boldsymbol{x}^{(i,1)} = \text{smoother}(\boldsymbol{x}^{(i)}, \boldsymbol{D}, \nu_1)$
   - ν_1 iterations
   - preconditioner $\boldsymbol{D}$
2. **Coarse-Grid Correction:**
   - compute residual $\boldsymbol{d} = \boldsymbol{f} - \boldsymbol{S}\,\boldsymbol{x}^{(i,1)}$
   - restrict residual $\bar{\boldsymbol{f}} = \boldsymbol{R}\boldsymbol{d}$
   - solve coarse-grid problem $\bar{\boldsymbol{S}}\bar{\boldsymbol{x}}^{(i)} = \bar{\boldsymbol{f}}$
   - prolongation and solution update $\boldsymbol{x}^{(i,2)} = \boldsymbol{x}^{(i,1)} + \boldsymbol{P}\bar{\boldsymbol{x}}^{(i)}$
3. **Post-Smoothing:** $\boldsymbol{x}^{(i+1)} = \text{smoother}(\boldsymbol{x}^{(i,2)}, \boldsymbol{D}, \nu_2)$
   - ν_2 iterations
   - preconditioner $\boldsymbol{D}$

On the fine grid with N spatial unknowns in each time step, a smoother is applied to a given initial guess to smooth high-frequency error components in the first place. Then the coarse-grid correction is performed. It consists of computing the residual on the fine grid, which is transferred to the coarse grid using the restriction operator $\boldsymbol{R}$. With $\bar{N}$ spatial nodes on this coarse level, the system matrix $\bar{\boldsymbol{S}} \in \mathbb{R}^{\bar{N}K \times \bar{N}K}$ is discretized in the same way as $\boldsymbol{S}$ was constructed before. The coarse-grid solution $\bar{\boldsymbol{x}}^{(i)} \in \mathbb{R}^{\bar{N}K}$ is then computed exactly. By prolonging this solution to the fine grid using the prolongation operator $\boldsymbol{P}$, the pre-smoothed solution can be corrected. Finally, some post-smoothing can be performed on the fine mesh to again dampen highly oscillating error components. The preconditioner $\boldsymbol{D}$ involved in the definition of the smoother should be designed to be efficiently applicable and will be discussed in what follows.

Smoothing We consider the iterative (damped) block Jacobi method

$$\boldsymbol{x} \mapsto \boldsymbol{x} + \omega \boldsymbol{D}^{-1}(\boldsymbol{f} - \boldsymbol{S}\boldsymbol{x}) \tag{9}$$

applied to the linear system of equations (8) using a damping parameter $\omega \in (0,1]$, where the block Jacobi preconditioner is the block diagonal of the system matrix $\boldsymbol{S}$:

$$\boldsymbol{D} := \begin{pmatrix} \square & & \\ & \ddots & \\ & & \square \end{pmatrix} \quad \text{with block matrix entries} \quad \underbrace{\boxed{\begin{matrix} * & & & \\ * & * & & \\ & \ddots & \ddots & \\ & & * & * \end{matrix}}}_{\in \mathbb{R}^{K\times K}}. \tag{10}$$

The diagonal entries of this matrix correspond to the diagonal entries of $\boldsymbol{A}_I = \boldsymbol{M}_h + \frac{1}{2}\Delta t(\varepsilon \boldsymbol{L}_h + \boldsymbol{K}_h)$ due to the construction of $\boldsymbol{S}$ using a Kronecker product. They do not vanish due to the diagonal entries of $\boldsymbol{M}_h$, $\boldsymbol{L}_h$, and $\boldsymbol{K}_h$. Therefore, the lower triangular preconditioner $\boldsymbol{D}$ is invertible. The individual blocks of $\boldsymbol{D}$ provide a high degree of parallelization because they can be considered independently of each other and only couple the (temporal) degrees of freedom associated with a single spatial node. Its lower bidiagonal structure makes it easy to solve the resulting linear systems of equations by other appropriate direct or iterative approaches.

For smoothing purposes, we employ the generalized minimal residual (GMRES) method without restart as introduced in [24], where the linear system of equations is preconditioned by the block Jacobi matrix $\boldsymbol{D}$. A total number of ν_1 and $\nu_2 \geq 0$ iterations are performed to damp high-frequency error components in the pre- and post-smoothing steps of the multigrid algorithm, respectively.

Transfer between space–time grids Intergrid transfer operators are necessary to exploit the coarse-grid correction in the multigrid method. The idea is to use common coarsening techniques in space while the temporal mesh stays fixed throughout the whole algorithm. The two-grid case is considered again, where h and $H = 2h$ are the mesh sizes of the fine level and the coarse level, respectively. To restrict the residual from the fine grid $\mathcal{T}_h$ to the coarse grid $\mathcal{T}_H$, the restriction operator

$$\boldsymbol{R} := \boldsymbol{R}_h^H \otimes \boldsymbol{I}_K = \begin{pmatrix} \frac{1}{2} & 1 & \frac{1}{2} & & & & & \\ & & \frac{1}{2} & 1 & \frac{1}{2} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \frac{1}{2} & 1 & \frac{1}{2} \end{pmatrix} \otimes \boldsymbol{I}_K \tag{11}$$

has to be applied, which is defined as the Kronecker product of the canonical restriction operator $\boldsymbol{R}_h^H \in \mathbb{R}^{\bar{N}\times N}$ of the FE space and the identity matrix $\boldsymbol{I}_K \in \mathbb{R}^{K\times K}$. Similarly, the prolongation operator $\boldsymbol{P} = \boldsymbol{R}^\top$ projects an FE solution from the coarse grid to the fine grid by means of an interpolation for piecewise linear functions [23, Section 5.1].

In general, the two-grid algorithm described above can be easily transferred to a multigrid method by recursively applying the two-grid idea to approximately solve the coarse-grid problem. In this case, only the discrete problem on the coarsest mesh has to be solved exactly. In the following numerical multigrid examples, we mostly focus on the well-known V- and F-cycle as described in [23, Section 5.1, Section 5.4].

Overall, the presented time-simultaneous multigrid algorithm can be interpreted as a variant of WRMG as introduced in [6] when using the Jacobi smoother and the same time-stepping method. This interpretation has already been discussed in [25]. In this context, we therefore refer to the existing literature on convergence analysis of the WRMG method in [11, 12, 26, 27].

2.3. Numerical studies

We now numerically investigate the solution behavior of the time-simultaneous multigrid algorithm for the convection–diffusion equation (1) in 1D as a representative of similar results in higher dimensions. For this purpose, various combinations of the velocity field and the diffusion coefficient are considered, including the special case of the heat equation for $v = 0$ and convection-dominated transport problems for $v = 1$ and small ε. Unless otherwise specified, we consider the manufactured solution

$$u(x,t) = \exp(-\eta(\tfrac{1}{2} - x + \tfrac{1}{4}\sin(\tfrac{\pi}{2}t))^2)\sin(\pi x), \quad (x,t) \in (0,1) \times (0,T), \qquad (12)$$

where the parameter $\eta = 100$ characterizes the steepness of u and is chosen to keep the spatial and temporal error in balance. The peak of the function oscillates periodically in time and follows the shape of a sine curve, as shown for certain time instances in Figure 1.

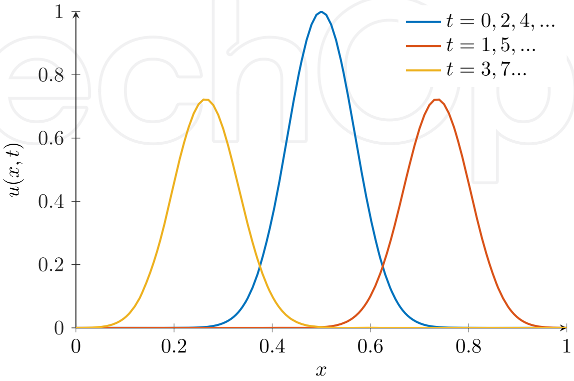


Figure 1. Smooth solution (12) on $\Omega = (0, 1)$ for certain time steps.

This exact solution satisfies homogeneous Dirichlet boundary conditions while the initial data is given by

$$u(x, 0) = \exp(-\eta(\tfrac{1}{2} - x)^2) \sin(\pi x), \tag{13}$$

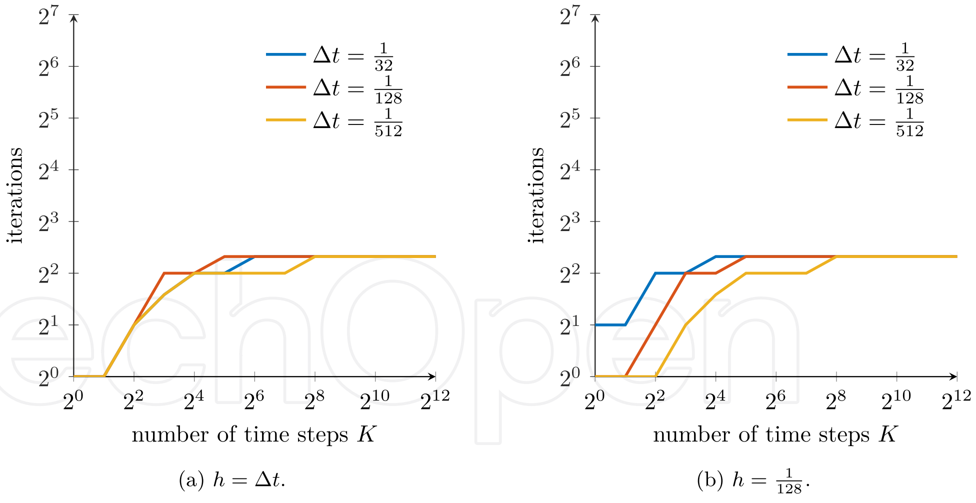
coinciding with $u\ (x, t)$ for $t = 0,\ 2,\ 4, \ldots$. As mentioned in Section 2.1, we discretize the problem in space using an equidistant mesh and linear FEs with quadrature-based mass lumping, while the Crank–Nicolson scheme with a fixed time step size Δt is employed for time integration. This results in second-order accuracy in space and time. In addition, a first-order upwind discretization is considered to further illustrate the impact of the findings. This discretization technique is a special Petrov–Galerkin method and can be interpreted in the FE context as the Galerkin method supplemented by first-order artificial diffusion [28, Section 8.2.2].

In the following studies, parameters like the mesh size h, the number of global time steps K as well as the time increment Δt are varied, implying the final time $T = \Delta t \cdot K$. In the context of the multigrid algorithm, the mesh size $h = 2^{-l}$ describes the resolution on the fine level l. In addition to the multigrid case, where the coarse level is always chosen to be level 1, we also investigate the two-grid case, using the coarse level $l - 1$. The smoother is given by the GMRES method without restart, which uses block Jacobi preconditioning and performs $\nu_1 = \nu_2 = 4$ pre- and post-smoothing steps. We report the number of iterations required to reduce the norm of the initial residual $||f||_2$ for a zero initial guess by a factor of 10^{-8}, while the maximum number of iterations is set to 100.

2.3.1. Heat equation

First, we take a brief look at some time-simultaneous multigrid results for the special case of the one-dimensional heat equation, which is equivalent to the partial differential equation (1) with the fixed velocity field $v = 0$. As known from the theory for the time-simultaneous two-grid algorithm with a damped block Jacobi smoother, the spectral norm of the iteration matrix is uniformly bounded above by a value smaller than 1, which is independent of the mesh size, the time step size, and the number of time steps [12]. In Figure 2, we investigate the V-cycle with the preconditioned GMRES smoother, where the number of iterations is plotted for different values of K. For a fixed ratio between the spatial and temporal resolutions, the number of iterations is indeed bounded above for the different numbers of time steps and does not depend on the fine mesh size. The upper bound is even independent of the time step size Δt for sufficiently large K. In this special case, the number of iterations is bounded above by a value of 5.

Therefore, we can treat many time steps simultaneously without increasing the number of iterations. The resulting linear system of equations can then be solved efficiently in parallel due to the fact that the application of the preconditioner provides a decomposition into N independent local systems, which are each large enough to be solved on a single processor.



(a) $h = \Delta t$.

(b) $h = \frac{1}{128}$.

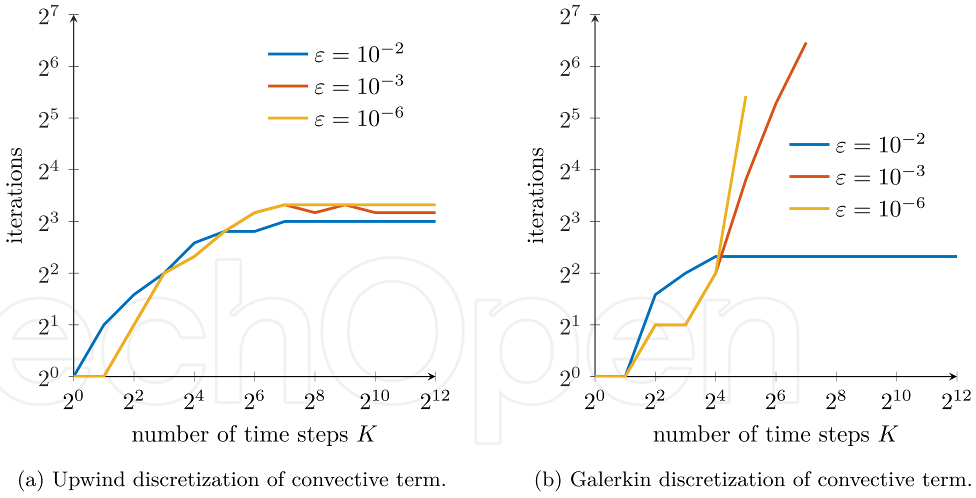
Figure 2. Number of iterations for multigrid method using V-cycle in the case of the smooth solution, $v = 0$, and $\varepsilon = 10^{-2}$.

2.3.2. Convection-dominated transport problems

In what follows, we focus on the convection–diffusion equation by setting the velocity field to $v = 1$ and, thus, use the Galerkin discretization for the convective part as well. Due to the one-dimensional space and the use of mass lumping, the discretized convective term corresponds to the second-order central difference operator for the first derivative in the context of FDs. Furthermore, we also consider a lower-order discretization at this point. The following tests are additionally investigated for the convection term discretized by the first-order upwinding technique as mentioned in [28, Section 8.2.2].

Figure 3 summarizes the multigrid results for a fixed fine level and time step size while different values of the diffusion coefficient are investigated. For the upwind approach and $\varepsilon = 10^{-2}$, it can be observed that the number of iterations stays bounded above even for a large time interval under consideration. Since we study the convection-dominated case in particular, smaller values of ε are also examined. In this case, we still find similar upper bounds for the number of iterations. Overall, these observations show a similar behavior to the studies of the heat equation; see Section 2.3.1. However, the number of iterations immediately reaches its maximum of 100 if we decrease the value of ε and treat more time steps simultaneously for the Galerkin discretization of the convective term as presented in Figure 3b.

More precisely, the time-simultaneous multigrid algorithm converges fast if the diffusion coefficient is chosen sufficiently large. This also holds true for convection-dominated problems using $\varepsilon = 10^{-3}, 10^{-6}$ and a maximum number of blocked time steps of 16. However, in this convection-dominated regime, the number of iterations required to solve the global linear system of equations increases



(a) Upwind discretization of convective term.

(b) Galerkin discretization of convective term.

Figure 3. Number of iterations for multigrid method using V-cycle in the case of the smooth solution, $v = 1$, and fixed $h = \Delta t = \frac{1}{128}$.

significantly when more time steps K are treated simultaneously. The same results can be observed for simultaneous two-grid approaches.

Next, we highlight the quality of the solution by focusing on the Heaviside step function as the exact solution with periodic boundary conditions, velocity field $v = 1$, and diffusion coefficient $\varepsilon = 0$. Figure 4 compares the exact solution at final time $T = 1$ with the numerical solutions obtained using either the upwind scheme or the Galerkin discretization for the convective term. Although strong oscillations can be observed in the solution of the second-order approach, the upwind scheme provides a highly diffusive approximation.

The instabilities and oscillations occurring in the case of unstabilized Galerkin FEs are well known and can obviously be avoided using the upwind scheme. However, this approach is very diffusive and only leads to first-order accuracy $\mathcal{O}(h)$ [28, 29, Section 8.3]. To preserve the higher order of the Galerkin discretization and especially improve the convergence of the solver even for a large number of time steps, we next introduce some stabilization based on the VMS method. Although accelerating the convergence of the solver is the main focus of this modification, we also observe from Figure 4 that oscillations in the numerical solution are damped for a specific stabilization parameter α as a positive side effect in this work.

3. Variational multiscale stabilization

When using the Galerkin approximation of the convection–diffusion equation (1), the numerical solution might be inaccurate and polluted by unphysical oscillations if the diffusion coefficient ε is small compared to the mesh size h and the magnitude of

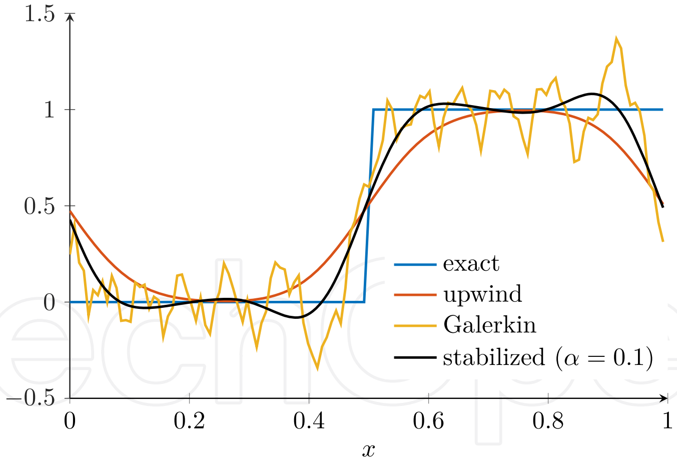


Figure 4. Heaviside step function at final time $T = 1$ in the case of $h = \Delta t = \frac{1}{128}$ and corresponding numerical solutions compared to stabilized discretization with stabilization parameter α.

the velocity field $\mathbf{v}$ [28, Section 8.3]. This problem was also observed in the numerical examples above and calls for some stabilization that attempts to reduce instabilities and artifacts in the numerical solution. In this paper, we focus on the VMS method, first introduced for this purpose in [20], and in particular presented as a projection-based extension in [22] and [21]. The stabilization technique is presented below based on the d-dimensional problem under consideration. This is followed by an interpretation in the context of FDs and numerical examples to discuss the influence of the method on the accuracy of the solution and the performance of the multigrid solver in 1D.

3.1. Definition

We again consider the d-dimensional convection–diffusion equation as presented in (1), where the velocity field $\boldsymbol{v}(x,t) = \boldsymbol{v} \in \mathbb{R}^d$ and the diffusion coefficient $\varepsilon > 0$ are given. The VMS method under investigation introduces an additional diffusive term and removes low-frequency diffusion by means of the divergence of a recovered gradient approximation to the variational form introduced in (3). Let $(V_h)^d$ denote a d-dimensional vector-valued FE subspace of $(L^2(\Omega))^d$ and $\alpha_{add} \geq 0$ be the constant stabilization parameter to be defined later. Then the solution $(u_h, \boldsymbol{g}_h)$ is sought so that

$$
\begin{aligned}
(\partial_t u_h, \varphi_h) + \varepsilon(\nabla u_h, \nabla \varphi_h) + (\boldsymbol{v} \cdot \nabla u_h, \varphi_h) & & & \\
+ \alpha_{add}(\nabla u_h, \nabla \varphi_h) - \alpha_{add}(\boldsymbol{g}_h, \nabla \varphi_h) &= (f, \varphi_h) & \forall \varphi_h \in V_h, & \qquad (14)\\
(\boldsymbol{g}_h - \nabla u_h, \boldsymbol{\psi}_h) &= 0 & \forall \boldsymbol{\psi}_h \in (V_h)^d &
\end{aligned}
$$

is satisfied, where $\boldsymbol{g}_h \colon (0, T) \to (V_h)^d$ corresponds to the projected gradient of u_h and ensures that the introduced stabilization term vanishes in the continuous problem. In contrast to the projection-based VMS stabilization technique as published in [22] and [21], the gradient is approximated using the same FE space defined on the triangulation $\mathcal{T}_h$. This procedure is also considered in [30, Section 5].

After discretization in space, the problem in matrix form reads as

$$\begin{aligned} \boldsymbol{M}_h \partial_t \boldsymbol{u}_h(t) + \varepsilon \boldsymbol{L}_h \boldsymbol{u}_h(t) + \boldsymbol{K}_h \boldsymbol{u}_h(t) + \alpha_{add} \boldsymbol{L}_h \boldsymbol{u}_h(t) - \alpha_{add} \boldsymbol{B}_h^\top \boldsymbol{g}_h(t) &= \boldsymbol{f}_h(t), \\ \boldsymbol{N}_h \boldsymbol{g}_h(t) - \boldsymbol{B}_h \boldsymbol{u}_h(t) &= 0, \end{aligned} \tag{15}$$

where $\boldsymbol{B}_h \in \mathbb{R}^{N \times N}$ is the discrete counterpart of the gradient and $\boldsymbol{N}_h$ is the mass matrix corresponding to the vector-valued FE space $(V_h)^d$. We eliminate the second equation by substituting $\boldsymbol{g}_h(t)$ into the first equation. This results in the system to be solved:

$$\boldsymbol{M}_h \partial_t \boldsymbol{u}_h(t) + \varepsilon \boldsymbol{L}_h \boldsymbol{u}_h(t) + \boldsymbol{K}_h \boldsymbol{u}_h(t) + \alpha_{add} \boldsymbol{W}_h \boldsymbol{u}_h(t) = \boldsymbol{f}_h(t), \tag{16}$$

where the stabilization matrix $\boldsymbol{W}_h := \boldsymbol{L}_h - \boldsymbol{B}_h^\top \boldsymbol{N}_h^{-1} \boldsymbol{B}_h$ can be explicitly determined for a diagonal mass matrix $\boldsymbol{N}_h$. As before, the Crank–Nicolson scheme is used as the time integrator. Then the discrete counterpart of (16) is given by

$$\boldsymbol{A}_I \boldsymbol{u}_h^m + \boldsymbol{A}_E \boldsymbol{u}_h^{m-1} + \frac{\Delta t}{2} \alpha_{add} \boldsymbol{W}_h \boldsymbol{u}_h^m + \frac{\Delta t}{2} \alpha_{add} \boldsymbol{W}_h \boldsymbol{u}_h^{m-1} = \boldsymbol{f}^m, \quad m = 1, \dots, K \tag{17}$$

as a straightforward extension of (5). Generally, the sparsity pattern of the stabilization matrix $\boldsymbol{W}_h$ is more dense than the one of $\boldsymbol{A}_I$ and $\boldsymbol{A}_E$ due to the fact that the multiplication of FE matrices is involved in the definition of $\boldsymbol{W}_h$. Thus, stabilizing the system comes at the expense of a more complex iterative solution strategy. We will come back to this observation when discussing the matrix structures in more detail. Finally, in this work, we treat the stabilization fully implicitly to reduce the computational effort. This results in the final formulation of the stabilized FE discretization to (1):

$$\boldsymbol{A}_I \boldsymbol{u}_h^m + \boldsymbol{A}_E \boldsymbol{u}_h^{m-1} + \Delta t \alpha_{add} \boldsymbol{W}_h \boldsymbol{u}_h^m = \boldsymbol{f}^m, \quad m = 1, \dots, K. \tag{18}$$

In the next section, we will argue that this fully implicit treatment is reasonable and actually does not reduce the accuracy of the numerical solution. We will then discuss the level-dependent choice of the stabilization parameter α_{add} in the context of the multigrid approach and study the stabilization numerically.

3.2. Interpretation using finite differences in 1D

To explore the fully implicit treatment of $\boldsymbol{W}_h$ in more detail, we first consider the time discretization of the stabilization.

Time discretization In Section 2.1, we introduced the Crank–Nicolson scheme for time integration of the original semi-discrete formulation. Since the stabilization term is treated fully implicitly, we now investigate its effect on the order of accuracy starting from the discrete form (18), where $\boldsymbol{W}_h$ is added to the unknown of the mth time step with time step size Δt. For this purpose, we algebraically transform the problem at hand into the following form:

$$\begin{aligned} \boldsymbol{A}_I \boldsymbol{u}_h^m + \boldsymbol{A}_E \boldsymbol{u}_h^{m-1} + \Delta t \alpha_{add} \boldsymbol{W}_h \boldsymbol{u}_h^m &= \boldsymbol{f}^m \\ \Leftrightarrow \qquad \hat{\boldsymbol{A}}_I \boldsymbol{u}_h^m + \hat{\boldsymbol{A}}_E \boldsymbol{u}_h^{m-1} &= \boldsymbol{f}^m, \end{aligned} \tag{19}$$

where the matrices $\hat{\boldsymbol{A}}_I$ and $\hat{\boldsymbol{A}}_E$ are given by

$$\begin{aligned} \hat{\boldsymbol{A}}_I &:= (\boldsymbol{M}_h + \alpha_{add} \tfrac{\Delta t}{2} \boldsymbol{W}_h) + \tfrac{\Delta t}{2} (\varepsilon \boldsymbol{L}_h + \boldsymbol{K}_h + \alpha_{add} \boldsymbol{W}_h), \\ \hat{\boldsymbol{A}}_E &:= -(\boldsymbol{M}_h + \alpha_{add} \tfrac{\Delta t}{2} \boldsymbol{W}_h) + \tfrac{\Delta t}{2} (\varepsilon \boldsymbol{L}_h + \boldsymbol{K}_h + \alpha_{add} \boldsymbol{W}_h). \end{aligned} \tag{20}$$

This is nothing else but the Crank–Nicolson discretization of

$$\begin{aligned} (\boldsymbol{M}_h + \alpha_{add} \tfrac{\Delta t}{2} \boldsymbol{W}_h) \partial_t \boldsymbol{u}_h(t) + (\varepsilon \boldsymbol{L}_h + \boldsymbol{K}_h + \alpha_{add} \boldsymbol{W}_h) \boldsymbol{u}_h(t) &= \boldsymbol{f}_h(t) \\ \Leftrightarrow \quad \boldsymbol{M}_h \partial_t \boldsymbol{u}_h(t) + (\varepsilon \boldsymbol{L}_h + \boldsymbol{K}_h) \boldsymbol{u}_h(t) + \alpha_{add} \boldsymbol{W}_h (\tfrac{\Delta t}{2} \partial_t \boldsymbol{u}_h(t) + \boldsymbol{u}_h(t)) &= \boldsymbol{f}_h(t), \end{aligned} \tag{21}$$

and therefore guarantees that the order of convergence of the numerical approximation is not reduced if the semi-discrete solution to (21) converges to the exact solution of (1) with second order in space. This derivation motivates the more general variational formulation of the VMS stabilization in the d-dimensional case:

$$\begin{aligned} &(\partial_t u_h, \varphi_h) + \varepsilon(\nabla u_h, \nabla \varphi_h) + (\boldsymbol{v} \cdot \nabla u_h, \varphi_h) \\ &\quad + \alpha_{add} [(\nabla [u_h + \tfrac{\Delta t}{2} \partial_t u_h], \nabla \varphi_h) - (\boldsymbol{g}_h, \nabla \varphi_h)] = (f, \varphi_h) \quad \forall \varphi_h \in V_h, \\ &\qquad (\boldsymbol{g}_h - \nabla [u_h + \tfrac{\Delta t}{2} \partial_t u_h], \boldsymbol{\psi}_h) = 0 \quad \forall \boldsymbol{\psi}_h \in (V_h)^d. \end{aligned} \tag{22}$$

Discretization in time using the Crank–Nicolson scheme results in a fully implicit stabilization term and, hence, reduces the numerical complexity compared to problem (17). Next, we focus on the spatial accuracy of (21) and, for this, consider an FD interpretation of the stabilization matrix under investigation.

Space discretization Using one-dimensional linear FEs, a uniform grid, and quadrature-based mass lumping, both mass matrices $\boldsymbol{M}_h$ and $\boldsymbol{N}_h$ correspond to a scaled identity matrix. More precisely, the matrices under consideration for the inner degrees of freedom are given by

$$\boldsymbol{N}_h = \boldsymbol{M}_h = \begin{pmatrix} h & & \\ & \ddots & \\ & & h \end{pmatrix}, \qquad \boldsymbol{B}_h = \frac{1}{2}\begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix}, \tag{23}$$

so that the second part of the stabilization matrix is given by

$$\tilde{\boldsymbol{L}}_h := \boldsymbol{B}_h^\top \boldsymbol{N}_h^{-1} \boldsymbol{B}_h = -\frac{1}{4h}\begin{pmatrix} -3 & 0 & 1 & & & & \\ 0 & -2 & 0 & 1 & & & \\ 1 & 0 & -2 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & 0 & -2 & 0 & 1 \\ & & & 1 & 0 & -2 & 0 \\ & & & & 1 & 0 & -3 \end{pmatrix}. \tag{24}$$

Using the definition of the tridiagonal matrix $\boldsymbol{L}_h$ as the negative of the discrete Laplacian and already known from the diffusive part, we conclude that the stabilization matrix $\boldsymbol{W}_h$ reads as

$$\boldsymbol{W}_h := \boldsymbol{L}_h - \tilde{\boldsymbol{L}}_h = -\frac{1}{h}\begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} + \frac{1}{4h}\begin{pmatrix} -3 & 0 & 1 & & & & \\ 0 & -2 & 0 & 1 & & & \\ 1 & 0 & -2 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & 0 & -2 & 0 & 1 \\ & & & 1 & 0 & -2 & 0 \\ & & & & 1 & 0 & -3 \end{pmatrix}. \tag{25}$$

To interpret this matrix in the context of FDs, we multiply equation (21) by $\boldsymbol{M}_h^{-1} = h^{-1}\boldsymbol{I}$. This results in the FD matrices $\boldsymbol{L}_h^* := \boldsymbol{M}_h^{-1}\boldsymbol{L}_h$ and $\tilde{\boldsymbol{L}}_h^* := \boldsymbol{M}_h^{-1}\tilde{\boldsymbol{L}}_h$, which are equivalent to discrete diffusion operators applied to two different mesh sizes h and $2h$. In summary, the FD stabilization matrix is pentadiagonal and represented by

$$\boldsymbol{W}_h^* := \boldsymbol{M}_h^{-1}\boldsymbol{W}_h = \boldsymbol{L}_h^* - \tilde{\boldsymbol{L}}_h^* = \frac{1}{(2h)^2}\begin{pmatrix} 5 & -4 & 1 & & & & \\ -4 & 6 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 6 & -4 \\ & & & & 1 & -4 & 5 \end{pmatrix} \in \mathbb{R}^{N\times N}. \tag{26}$$

Adding this stabilization to the original problem leads to a larger bandwidth of the system matrix and therefore increases the cost of the iterative solution strategy. The fully implicit treatment of the stabilization, as introduced at the beginning of this section, minimizes this additional effort.

A brief look at the Taylor expansion of the two central difference quotients,

$$\begin{aligned} \boldsymbol{L}_h^*\boldsymbol{u}_h &\sim \tfrac{1}{h^2}\left(h^2 u_{xx} + Ch^4 u_{xxxx} + O(h^6)\right) \\ \tilde{\boldsymbol{L}}_h^*\boldsymbol{u}_h &\sim \tfrac{1}{4h^2}\left(4h^2 u_{xx} + 2^4 Ch^4 u_{xxxx} + O(h^6)\right) \\ \Rightarrow \quad \boldsymbol{W}_h^* := (\boldsymbol{L}_h^* - \tilde{\boldsymbol{L}}_h^*)\boldsymbol{u}_h &\sim \tilde{C}h^2 u_{xxxx} + O(h^4), \end{aligned} \tag{27}$$

where $\tilde{C} := -3C$, illustrates that the stabilization matrix $\boldsymbol{W}_h^*$ corresponds to a scaled FD discretization of u_{xxxx} while the factor h^2 guarantees second order of accuracy in space.

The d-dimensional stabilization problem was already given at the beginning of this section in (22) where the time discretization was considered. This problem corresponds in one dimension to the continuous case of the modified convection–diffusion equation

$$\partial_t u - \varepsilon u_{xx} + v u_x + \alpha_{add}\tilde{C}h^2\left[u + \tfrac{\Delta t}{2}\partial_t u\right]_{xxxx} = f. \tag{28}$$

Therefore, the stabilization is a perturbation of the continuous problem of order h^2, and the solution of (19) converges to the exact solution of (1) with second order. Although (28) is continuous in space, the fixed factor h^2 corresponds to the grid used to discretize the problem in hand. This is the main difference to the d-dimensional form, which is not derived from an FD point of view, and is a crucial aspect in the next section, where the choice of the stabilization parameter is discussed.

3.3. Choice of stabilization parameter

At this point, we motivate our level-dependent choice of the stabilization parameter α_{add} for the time-simultaneous multigrid approach applied to the global space–time system with space-major ordering resulting from the stabilized FE discretization (18). For this purpose, we consider the situation of the multigrid algorithm, where the mesh sizes of a fine level and some coarser level are given by h and H, respectively. The time step size Δt stays constant for all levels since coarsening is applied only in space. According to the FD interpretation mentioned above, the stabilized one-dimensional problem corresponding to the coarse level then reads as

$$\partial_t u - \varepsilon u_{xx} + v u_x + \alpha_{add}\tilde{C}H^2[u + \tfrac{\Delta t}{2}\partial_t u]_{xxxx} = f \tag{29}$$

and, hence, differs from the problem of the fine level given in (28), where the scaling of the stabilization term still depends on the mesh size h of the fine level. However, the aim is to solve the same continuous problem on each level within the multigrid algorithm, i.e., the fine-grid problem (28). By the choice of $\alpha_{add} = \alpha\left(\frac{h}{H}\right)^2$ in (29), this requirement is satisfied due to the fact that

$$\begin{aligned} & \partial_t u - \varepsilon u_{xx} + v u_x + \alpha\left(\frac{h}{H}\right)^2\tilde{C}H^2[u + \tfrac{\Delta t}{2}\partial_t u]_{xxxx} &= f \\ \Leftrightarrow \quad & \partial_t u - \varepsilon u_{xx} + v u_x + \alpha\tilde{C}h^2[u + \tfrac{\Delta t}{2}\partial_t u]_{xxxx} &= f. \end{aligned} \tag{30}$$

In the special case of the fine level with mesh size h, this parameter simplifies to $\alpha_{add} = \alpha$ so that the continuous problems resulting from the coarse level (29) and the fine level (28) coincide. Using some parameters $\alpha \geq 0$, $\gamma \geq 0$, we introduce a more general definition of the level-dependent stabilization parameter

$$\alpha_{add} := \alpha\left(\frac{h}{H}\right)^{\gamma}, \tag{31}$$

which is exploited to scale the stabilization term on the coarser level with mesh size H. While a reasonable value for α will be evaluated in the numerical studies, one possible choice of γ was already motivated above. According to (30), the same continuous problem is solved on each level for $\gamma = 2$. In this case, the value of α_{add} decreases for a larger mesh size H so that less stabilization is added on coarser levels. Corresponding effects can also be observed in the following numerical studies of the time-simultaneous multigrid method. Another obvious choice resulting from the d-dimensional stabilized problem (22) without the FD interpretation is given by $\gamma = 0$, which results in a level-independent value of α_{add}. The intermediate state $\gamma = 1$ of both derivations does not guarantee to solve the same continuous problem but keeps the stabilization parameter larger on the coarser levels than with $\gamma = 2$.

In the following section, we study the stabilization technique and the choice of different stabilization parameters numerically.

3.4. Numerical studies

This section focuses on the qualitative and quantitative effects on the solution behavior of the stabilization technique introduced above. For this purpose, the influence on the accuracy of the solution as well as the performance of the time-simultaneous multigrid solver applied to the stabilized system is investigated. As before, we again use linear FEs for discretization in space in 1D while the time integrator is given by the Crank–Nicolson scheme.

3.4.1. Accuracy of the solution

Our study on the quality of the solution computed by the stabilized method consists of two parts, which differ mainly in the smoothness of the considered exact solution. The choice of the parameter γ can be neglected at this point because we are only interested in the fine-grid solution, which is not affected by γ due to the definition of α_{add}.

Heaviside step function By considering the Heaviside step function and the coefficients $v = 1$ and $\varepsilon = 0$, we focus on the quality of the numerical solution as already discussed in Section 2.3.2. Since the higher-order stabilization is studied in combination with the Galerkin discretization, the corresponding solution of the unstabilized case, polluted by artificial oscillations, is the point of reference. In Figure 4, we observed these oscillations for the Galerkin and hence unstabilized discretization of the convective term while the first-order upwind scheme provided a highly diffusive result. Furthermore, this figure shows the numerical approximation of the stabilized configuration at the final time $T = 1$. In this case, the artifacts of the unstabilized solution are smoothed by the stabilization using $\alpha = 0.1$, which seems to be an appropriate choice as we will see below.

Order of convergence We now come back to the smooth solution (12) introduced in the numerical studies of the time-simultaneous algorithm in Section 2.3 and focus on the order of convergence, which was an important aspect for the choice of the stabilization technique. In the following investigations, the convection-dominated region with $\varepsilon = 10^{-3}$ and velocity field $v = 1$ is considered. In Table 1, we summarize the error for the final time $T = 2$ in the discrete L_2-norm for different values of the stabilization parameter α. The error is reduced by a factor of 4 when the mesh size and the time step size are both halved, no matter how the stabilization parameter is chosen. Since we consider linear FEs in space and the Crank–Nicolson scheme in time, this confirms the theoretical expectations with and without stabilizing the problem. Additionally, the error for the upwind discretization of the convective

Table 1. Discrete L_2-error at final time $T = 2$ in the case of the smooth solution, $v = 1$, and $\varepsilon = 10^{-3}$.

$h = \Delta t$	$\alpha = 0$	10^{-2}	10^{-1}	10^{0}	10^{1}	Upwind
1/64	1.8×10^{-3}	1.9×10^{-3}	6.4×10^{-3}	4.9×10^{-2}	1.6×10^{-1}	2.0×10^{-2}
1/128	4.6×10^{-4}	4.8×10^{-4}	1.7×10^{-3}	1.5×10^{-2}	8.8×10^{-2}	1.0×10^{-2}
1/256	1.1×10^{-4}	1.2×10^{-4}	4.2×10^{-4}	4.0×10^{-3}	3.5×10^{-2}	5.1×10^{-3}
1/512	2.9×10^{-5}	3.0×10^{-5}	1.0×10^{-4}	1.0×10^{-3}	9.9×10^{-3}	2.6×10^{-3}
1/1024	7.2×10^{-6}	7.6×10^{-6}	2.6×10^{-5}	2.5×10^{-4}	2.5×10^{-3}	1.3×10^{-3}

term can be found in the same table. We observe a factor of 2 for the reduction of the error as expected by the theoretical investigations in Section 2.3.2.

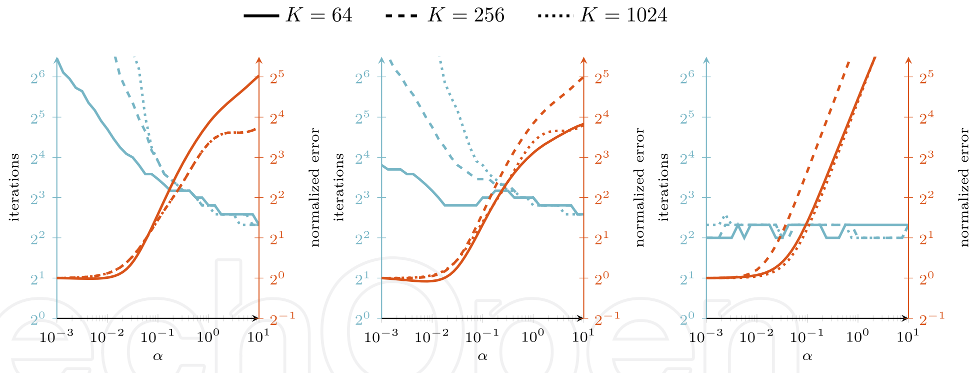
So far, we did not argue how to choose the stabilization parameter. This can be now done by considering Table 1 more precisely. First of all, we note that the accuracy of the solution deteriorates as α increases. For example, for $\alpha = 10^{-1}$, the numerical solution is approximately as accurate as in the case of $\alpha = 0$ for twice larger time increments and mesh sizes, i.e., we lose one level of mesh refinement, which we assume to be acceptable. Another comparison shows that the numerical solution for this amount of stabilization is even more accurate than in the upwind case. Therefore, as the first observation, choosing α not too large is reasonable from the point of view of accuracy. We will come back to this when discussing the convergence behavior of the multigrid solver below due to the fact that an accurate solution is an important aspect of the stabilization method.

3.4.2. Performance of the solver

In previous studies on the time-simultaneous method, a slow iterative convergence behavior was observed for the convection-dominated case when the diffusion coefficient was at most $\varepsilon = 10^{-3}$, the velocity field was set to 1, and 64 or more time steps were treated simultaneously. Therefore, we now focus on the effect of the stabilization method for this parameter setting and, especially, discuss the influence of the stabilization parameter α. In addition to the convergence behavior of the iterative solver, which is illustrated by the number of iterations necessary for the time-simultaneous method to solve the problem in hand, the behavior of the error is also considered for accuracy reasons. Furthermore, we are interested in finding criteria how to choose the stabilization parameter α: while the number of required iterations should not grow arbitrarily for different numbers of time steps K, the error is intended to be as close as possible to that of the Galerkin approximation for smooth solutions.

We first focus on the two-grid solution algorithm and then discuss the influence of its multigrid extension on the choice of the stabilization parameter.

Two-grid algorithm To illustrate how to read the subsequent figures, we explain the context of Figure 5a in detail. The numbers of iterations for the two-grid



(a) $h = \Delta t = \frac{1}{32}$. (b) $h = \frac{1}{32}, \Delta t = \frac{1}{128}$. (c) $h = \frac{1}{512}, \Delta t = \frac{1}{128}$.

Figure 5. Number of iterations and normalized error for two-grid method in the case of the smooth solution, $\nu = 1$, $\varepsilon = 10^{-3}$, and stabilization parameter $\gamma = 2$.

algorithm for three different numbers of simultaneously treated time steps K are shown in blue lines while the stabilization parameter α varies between 10^{-3} and 10^1. The results for the smallest value of $\alpha = 10^{-3}$ are in good agreement with the unstabilized ones since the convergence behavior is very similar in both cases. This means in this case that the solver does not converge within the maximum number of iterations. However, the solver converges significantly faster when more stabilization is incorporated into the system. For stabilization with $\alpha = 10^{-1}$, the lines for the different values of K meet, i.e., the algorithm converges independently of the number of time steps treated simultaneously and requires only a few iterations ($<2^4$) to solve the problem in hand. At the same time, and for this $\alpha = 10^{-1}$, the errors are about ("only") four times larger than those without stabilization, corresponding to a loss of accuracy of one mesh level. This can be observed in the same figure, where the discrete L_2-error normalized with respect to the error for $\alpha = 10^{-3}$ is plotted in red. Again, the errors for $\alpha = 10^{-3}$ are very close to the ones corresponding to $\alpha = 0$, which do not exploit any stabilization at all. Overall, the results presented in Figure 5a illustrate the desired effect of stabilization with $\alpha = 10^{-1}$ on the two-grid solver. The associated loss of accuracy seems to be acceptable if the algorithm recovers the original quality of the solution at a finer resolution while requiring significantly less iterations.

Next, the results for a finer time step size are summarized in Figure 5b. In this case, the convergence behavior improves slightly for smaller values of K and without stabilization (on the left of the x-axis). However, the remaining convergence issues can be further reduced by the stabilization so that the choice of $\alpha = 10^{-1}$ might still be beneficial.

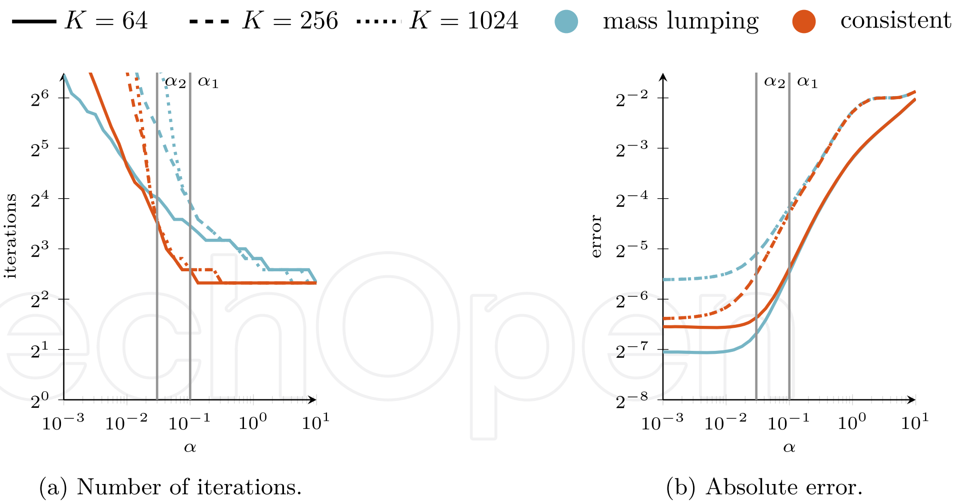
Finally, the case of a finer spatial mesh is presented in Figure 5c. We observe that a finer level improves the convergence behavior even without stabilization.

Although stabilization would not be necessary for this configuration, the figure illustrates that stabilization does not worsen the convergence behavior.

In summary, the stabilization parameter α should not be chosen too large to achieve accurate solutions while small values of α might not sufficiently improve the performance of the iterative solver. In this trade-off situation, the choice of $\alpha_1 : = 10^{-1}$ as an upper bound for α seems to be a good compromise as illustrated in Figure 5.

The previous results were computed using quadrature-based mass lumping, which allowed us to exploit an FD interpretation for the analysis of the stabilization. Furthermore, the employed preconditioner $\boldsymbol{D}$ becomes exact as $\Delta t \to 0$, which is not satisfied anymore when all integrals are computed exactly. In Figure 6, the behavior of the stabilization is shown for the discretization with the consistent mass matrix $\boldsymbol{M}_h^c$ in front of the time derivative, i.e., $\boldsymbol{M}_h^c \partial_t \boldsymbol{u}_h(t)$, while mass lumping is still exploited for the computation of $\boldsymbol{N}_h$ in (16). The number of iterations is compared for the lumped and consistent mass matrices $\boldsymbol{M}_h$ and $\boldsymbol{M}_h^c$ in Figure 6a, and Figure 6b illustrates the corresponding discrete L_2-errors. A vertical line marks the value of $\alpha_1 = 10^{-1}$, which was derived in previous investigations as an appropriate balance between accuracy and convergence behavior in the case of the lumped mass matrix. Especially, the number of iterations stays bounded above independently of the number of time steps K. However, the replacement of the lumped mass matrix by the consistent one further reduces the numerical effort to solve the system in hand by a factor of 2. In this case, the number of iterations already stays bounded independently of K for a smaller value of α. The new upper bound α_2 is marked by another vertical line in the figure and indicates that the number of iterations is still lower than the ones using the lumped mass matrix for the stabilization parameter α_1. On the other hand, the errors are approximately the same for α_1 and both choices of the mass matrix. In the case of $\boldsymbol{M}_h^c$, even smaller values of α are acceptable, resulting in an improved accuracy of the numerical approximation. As a conclusion, by using the consistent mass matrix, we can achieve a similar error for α_1 but with less numerical effort. On the other hand, there is the possibility of choosing α even smaller for more accurate solutions while still achieving iteration numbers that are independent of the number of time steps. Due to this observation, the following investigations are performed using the consistent mass matrix and especially focusing on $\alpha = 10^{-1}$.

Multigrid algorithm Time-simultaneous multigrid results are shown in Figure 7 and illustrate the number of iterations and the normalized error behavior in the same way as in the two-grid analysis above. However, the behavior of the discrete L_2-error is slightly different due to the use of the consistent mass matrix instead of its lumped counterpart. The discrepancies have already been discussed in the previous section, and remain valid here since we are still solving the same problem (on the finest level) in both the two-grid and multigrid approaches. Therefore, we

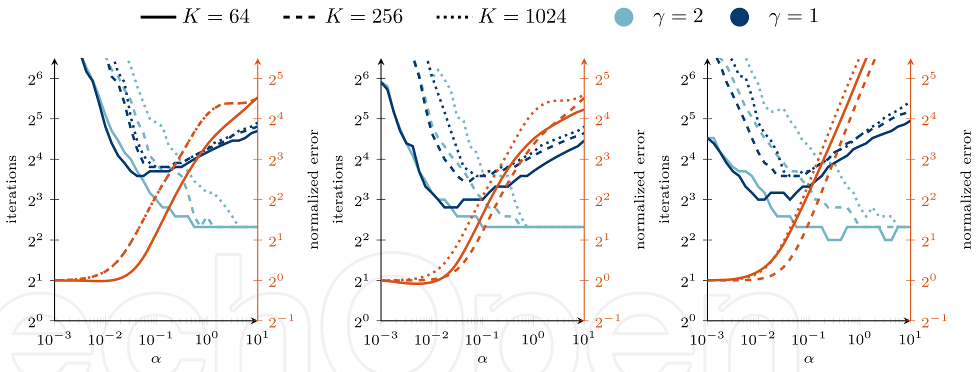


(a) Number of iterations.

(b) Absolute error.

Figure 6. Number of iterations and discrete L_2-error for two-grid method in the case of the smooth solution, $\nu = 1$, $\varepsilon = 10^{-3}$, stabilization parameter $\gamma = 2$, and $h = \Delta t = \frac{1}{32}$. Vertical lines mark $\alpha_1 = 0.1$ and $\alpha_2 = 0.03$.

focus mainly on the convergence behavior of the iterative solver in the following investigations. In Figures 7a and 7c, the fine mesh size and the time step size Δt coincide again. We first focus on the multigrid results for $\gamma = 2$ presented in Figure 7c, where finer resolutions are considered. From the two-grid findings, we conclude that the choice of $\alpha = 10^{-1}$ results in a good balance between accuracy of the solution and performance of the solver. Looking at the multigrid results stabilized with this value of α, we note that the number of iterations for different values of K can still be improved compared to the unstabilized case but increases when more time steps are treated simultaneously. The reason for this might be that the stabilization on a coarse grid seems not to be sufficient for uniform convergence behavior. Thus, the stabilization parameter α has to be chosen much larger in these cases leading to a loss of accuracy again. In Section 3.3, we already mentioned that the parameter $\gamma = 2$ implies less stabilization on coarser levels, which occurs especially in the multigrid case where level 1 corresponds to the coarsest mesh. To keep the stabilization parameter $\alpha_{add} := \alpha(\frac{h}{H})^{\gamma}$ larger on the coarser levels while still using the same value on the fine level, $\gamma = 1$ is additionally considered and shown in dark blue in the same figure. For this setup, there is indeed an improvement due to the fact that the number of iterations is again bounded for different values of K and the stabilization parameter $\alpha = 10^{-1}$. This behavior can also be observed in Figures 7a and 7b, where a coarser mesh size and time step sizes are investigated. In summary, the stabilized multigrid results are comparable to those of the two-grid algorithm, but the numbers of iterations are slightly higher.



(a) $h = \Delta t = \frac{1}{32}$. (b) $h = \frac{1}{32}, \Delta t = \frac{1}{128}$. (c) $h = \Delta t = \frac{1}{128}$.

Figure 7. Number of iterations and normalized error for multigrid method using F-cycle in the case of the smooth solution, $\nu = 1$, $\varepsilon = 10^{-3}$, and the consistent mass matrix.

These observations can also be confirmed with the help of Table 2b, which additionally contains results for the stabilization parameter $\alpha = 10^{-1}$ and, in particular, $\gamma = 0$. In certain multigrid configurations, especially for significantly larger values of K, the choice of $\gamma = 0$ provides convergence rates that are uniformly bounded above while the solver does not converge within the maximum number of iterations for $\gamma = 1$ and $\gamma = 2$. In contrast to that, the two-grid solver provides the lowest numbers of iterations bounded above for $\gamma = 2$. The corresponding results can be found in Table 2a.

Finally, we highlight two further scenarios. The first setup in Figure 8a deals with difficulties that occur when the fine mesh size is much smaller than the time step size Δt. In this case, neither the choice of $\gamma = 2$ nor $\gamma = 1$ will give the desired K-independent convergence behavior for $\alpha = 10^{-1}$. Choosing a slightly larger α would satisfy this criterion for $\gamma = 1$. However, the number of iterations is still quite large while the error is relatively small. Since the choice of a time step size larger than the mesh resolution is not necessarily physically reasonable for convection-dominated problems, the practical relevance of this consideration remains questionable. Even in the multigrid approach, the case where the spatial mesh size is much smaller than the time step size does not arise, since coarsening is only performed in space, but could occur for other coarsening strategies (in time). The second setup considered in Figure 8b fixes the final time $T = 2$ but increases the number of time steps K for smaller time step sizes Δt. In this case, the convergence behavior for both values of γ can be improved in the same way for the different numbers of time steps. For $\alpha = 10^{-1}$, we observe the desired effect of the stabilization even in the case of the multigrid algorithm while the solver does not converge within the maximum number of iterations in most unstabilized cases.

Table 2. Number of iterations in the case of the smooth solution, $\nu = 1$, $\varepsilon = 10^{-6}$, the consistent mass matrix, and stabilization parameter $\alpha = 10^{-1}$. A dash "-" indicates that the solver did not converge within the maximum number of 100 iterations.

(a) Two-grid

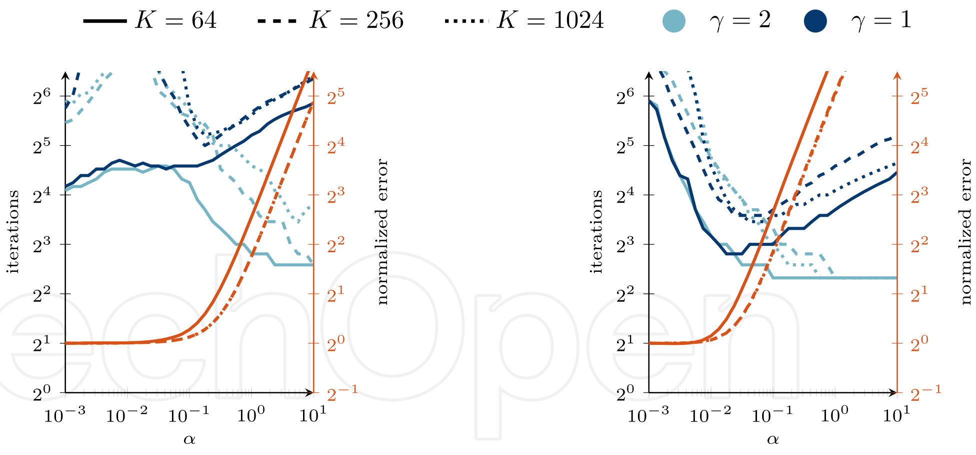
	$\gamma = 0$			$\gamma = 1$			$\gamma = 2$		
$K/h = \Delta t$	**1/32**	**1/128**	**1/512**	**1/32**	**1/128**	**1/512**	**1/32**	**1/128**	**1/512**
1	4	2	2	4	2	2	3	1	2
4	6	5	4	5	4	4	4	3	3
16	10	10	6	8	7	4	5	3	3
64	20	21	8	13	11	6	6	5	3
256	28	29	21	14	13	11	6	5	4
1024	28	28	24	14	12	12	6	5	5
4096	28	26	23	14	12	12	6	5	5
16384	28	25	21	14	11	11	6	5	4

(b) Multigrid (F-cycle)

	$\gamma = 0$			$\gamma = 1$			$\gamma = 2$		
$K/h = \Delta t$	**1/32**	**1/128**	**1/512**	**1/32**	**1/128**	**1/512**	**1/32**	**1/128**	**1/512**
1	1	2	2	3	1	1	2	1	2
4	6	5	6	5	4	4	4	3	3
16	10	10	14	8	7	9	5	3	3
64	20	20	19	13	11	8	9	5	4
256	28	34	33	15	15	12	19	12	8
1024	28	36	45	16	15	18	32	26	23
4096	29	35	48	29	21	19	–	51	42
16384	29	34	46	–	–	28	–	–	55

4. Conclusion and outlook

The time-simultaneous multigrid algorithm and its application to the one-dimensional heat equation were presented in the first part of this work. This investigation, as well as the performance of the solver for the convection–diffusion equation when the diffusion parameter is sufficiently large, illustrated that the rate of convergence is uniformly bounded above independently of the number of simultaneously treated time steps. Since convergence issues arise for the iterative solution strategy and convection-dominated problems, we introduced a VMS-type stabilization technique of higher order, which intends to improve the convergence behavior of the multigrid solver. The convergence behavior of the time-simultaneous two-grid algorithm could be extremely improved when using the stabilized system. In most cases, even the number of iterations of the multigrid algorithm is bounded



(a) Fixed grid sizes $\Delta t > h$.

(b) Fixed final time $T = 2$.

Figure 8. Number of iterations for multigrid method using F-cycle in the case of further setups with the smooth solution, $v = 1$, $\varepsilon = 10^{-3}$, and the consistent mass matrix.

above while still leading to second order of accuracy in space and time. To obtain those findings, the choice of the stabilization parameter is a crucial question. An FD interpretation of the stabilization was used to derive a level-dependent parameter for which the number of iterations in the two-grid studies was small and bounded above. A similar behavior has been found in the numerical studies of the multigrid case when more stabilization is added on the coarser grids, which is accompanied by a neglect of the consistency of the coarse-grid problem. Finally, as a side effect, it was observed that artifacts occurring in the solutions for the standard Galerkin discretization of convection-dominated problems can be smoothed by stabilization.

Future investigations on the described stabilization technique and its application to the presented multigrid algorithm include extensions to 2D and 3D problems as well as studies on the computational efficiency of the time-simultaneous approach. To exploit even more parallelism, this time-simultaneous multigrid algorithm could be combined with other approaches, such as parareal, MGRIT, or another multigrid version of parareal [31], which would extend the method by working parallel in time. As a forthcoming stabilization technique, it is convenient to apply the stabilization in time rather than in space since all time steps are treated simultaneously in a global system in this method. Overall, it is reasonable to explore especially the multigrid case for convection-dominated problems in more detail since less stabilization on coarser levels does not always show the desired effect; for example, adaptive control of the stabilization parameter or level-dependent numbers of smoothing steps might offer potential for improvement.

Conflict of interest

The authors declare no conflict of interest.

Acknowledgment

This work was supported by the Federal Ministry of Education and Research (BMBF) through the project "StroemungsRaum" 16ME0706K, which is part of the initiative "Neue Methoden und Technologien für das Exascale-Höchstleistungsrechnen" (SCALEXA) that aims at tackling the various challenges of large-scale scientific computing.

References

1 Gander MJ. 50 years of time parallel time integration. In: Carraro T, Geiger M, Körkel S, Rannacher R, editors. *Multiple shooting and time domain decomposition methods*. Cham: Springer International Publishing; 2015. p. 69–113.

2 Ong BW, Schroder JB. Applications of time parallelization. *Comput Visualization Sci*. 2020;**23**: 11.

3 Lions J, Maday Y, Turinici G. Résolution d'edp par un schéma en temps « pararéel ». *C R Acad Sci Sér I: Mathématique*. 2001;**332**(7):661–668.

4 Falgout RD, Friedhoff S, Kolev TzV, MacLachlan SP, Schroder JB. Parallel time integration with multigrid. *SIAM J Sci Comput*. 2014;**36**(6):C635–C661.

5 Gander MJ, Neumüller M. Analysis of a new space-time parallel multigrid algorithm for parabolic problems. *SIAM J Sci Comput*. 2016;**38**(4):A2173–A2208.

6 Lubich C, Ostermann A. Multi–grid dynamic iteration for parabolic equations. *BIT Numer Math*. 1987 Jun;**27**: 216–234.

7 Dünnebacke J, Turek S, Lohmann C, Sokolov A, Zajac P. Increased space-parallelism via time-simultaneous newton-multigrid methods for nonstationary nonlinear PDE problems. *Int J High Perform Comput Appl*. 2021 Apr;**35**(3):211–225.

8 Vandewalle S, Van de Velde E. Space-time concurrent multigrid waveform relaxation. *Ann Numer Math*. 1994;**1**: 347–360.

9 Vandewalle S, Piessens R. Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. *SIAM J Sci Stat Comput*. 1992;**13**(6):1330–1346.

10 Janssen J, Vandewalle S. Multigrid waveform relaxation on spatial finite element meshes: The continuous-time case. *SIAM J Numer Anal*. 1996;**33**(2):456–474.

11 Notay Y. Rigorous convergence proof of space-time multigrid with coarsening in space. *Numer Algorithms*. 2022;**89**(2):675–699.

12 Lohmann C, Dünnebacke J, Turek S. Fourier analysis of a time–simultaneous two–grid algorithm using a damped Jacobi waveform relaxation smoother for the one–dimensional heat equation. *J Numer Math*. 2022 Sept;**30**(3):173–207.

13 Farhat C, Chandesris M. Time-decomposed parallel time-integrators: Theory and feasibility studies for fluid, structure, and fluid-structure applications. *Int J Numer Methods Eng*. 2003 Nov;**58**: 1397–1434.

14 Bal G. *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, vol. 40, Berlin: Springer; 2005 Jan. p. 425–432.

15 De Sterck H, Falgout RD, Friedhoff S, Krzysik OA, MacLachlan SP. Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection. *Numer Linear Algebra Appl*. 2021 Mar;**28**(4):e2367.

16 De Sterck H, Falgout RD, Krzysik OA, Schroder JB. Efficient multigrid reduction-in-time for method-of-lines discretizations of linear advection. *J Sci Comput*. 2023;**96**: 1.

17 Gander MJ, Halpern L. Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. *SIAM J Numer Anal*. 2007;**45**(2):666–697.

18 Dong W, Tang H. Convergence analysis of waveform relaxation method to compute coupled advection-diffusion-reaction equations [Internet]. arXiv; 2022. Available from: https://arxiv.org/2205.01708 [math.NA].

19 Quarteroni A. *Numerical models for differential problems*. 2nd ed. NY: Springer Publishing Company, Incorporated; 2013.

20 Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Comput Methods Appl Mech Eng*. 1995;**127**(1):387–401.

21 John V, Kaya S, Layton W. A two-level variational multiscale method for convection-dominated convection–diffusion equations. *Comput Methods Appl Mech Eng*. 2006;**195**(33):4594–4603.

22 Layton W. A connection between subgrid scale eddy viscosity and mixed methods. *Appl Math Comput*. 2002;**133**(1):147–157.

23 Braess D. *Finite elements: theory, fast solvers, and applications in solid mechanics*. 3rd ed. Cambridge: Cambridge University Press; 2007.

24 Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *Siam J Sci Stat Comput*. 1986;**7**: 856–869.

25 Dünnebacke J, Turek S, Zajac P, Sokolov A. A time-simultaneous multigrid method for parabolic evolution equations. In: Vermolen FJ, Vuik C, editors. *Numerical mathematics and advanced applications ENUMATH 2019*. Cham: Springer International Publishing; 2021. p. 333–342.

26 Vandewalle S, Horton G. Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. *Computing*. 1995;**54**: 317–130.

27 Janssen J, Vandewalle S. Multigrid waveform relaxation on spatial finite element meshes: the discrete-time case. *SIAM J Sci Comput*. 1996;**17**: 133–155.

28 Quarteroni A, Valli A. *Numerical approximation of partial differential equations*. Berlin, Heidelberg: Springer; 1994.

29 John V, Schmeyer E. A two-level variational multiscale method for convection-dominated convection–diffusion equations. *Comput Methods Appl Mech Eng*. 2008;**198**: 475–494.

30 Lohmann C, Kuzmin D, Shadid JN, Mabuza S. Flux-corrected transport algorithms for continuous Galerkin methods based on high order Bernstein finite elements. *J Comput Phys*. 2017;**344**: 151–186.

31 Wambach L, Turek S. Numerical studies of a multigrid version of the parareal algorithm. In: *Tech. rep. Ergebnisberichte des Instituts für Angewandte Mathematik, Nummer 650*. TU Dortmund: Fakultät für Mathematik; 2022 Mar.