Technische Universität Graz

Comparison of algebraic multigrid methods for an adaptive space–time finite element discretization of the heat equation in 3D and 4D

O. Steinbach, H. Yang

Berichte aus dem
Institut für Numerische Mathematik

Bericht 2017/5

Technische Universität Graz

Comparison of algebraic multigrid methods for an adaptive space–time finite element discretization of the heat equation in 3D and 4D

O. Steinbach, H. Yang

Berichte aus dem
Institut für Numerische Mathematik

Bericht 2017/5

Technische Universität Graz
Institut für Numerische Mathematik
Steyrergasse 30
A 8010 Graz

WWW: `http://www.numerik.math.tu-graz.at`

Comparison of algebraic multigrid methods for an adaptive space–time finite element discretization of the heat equation in 3D and 4D

Olaf Steinbach[*] Huidong Yang[†]

September 15, 2017

Abstract

The aim of this work is to compare algebraic multigrid preconditioned GMRES methods for solving the nonsymmetric and positive definite linear systems of algebraic equations, that arise from a space–time finite element discretization of the heat equation in 3D and 4D space–time domains. The finite element discretization is based on a Galerkin–Petrov variational formulation employing piecewise linear finite elements simultaneously in space and time. We focus on a performance comparison of conventional and modern algebraic multigrid methods for such finite element equations, as well as robustness with respect to the mesh discretization and the heat capacity constant. We discuss different coarsening and interpolation strategies in the algebraic multigrid methods for coarse grid selection and coarse grid matrix construction. Further, we compare algebraic multigrid performance for the space–time finite element discretization on both uniform and adaptive meshes consisting of tetrahedra and pentachora in 3D and 4D, respectively. The mesh adaptivity occurring in space and time is guided by a residual based a posteriori error estimation.

Keywords: algebraic multigrid, coarsening, interpolation, space–time finite element, adaptivity
2010 MSC: 65M60, 65M50

1 Introduction

Space–time finite element methods have received more and more interest since the pioneering work in [22]; see some recent work, e.g., [2, 5, 6, 13, 14, 15, 17, 19, 27, 34, 35, 41, 42, 47]. A common difficulty of these methods concerns the efficient solution of the related large scale linear system of algebraic equations which is often harder than the solution of the linear system from more

[*] Institut für Numerische Mathematik, Technische Universität Graz, Email: o.steinbach@tugraz.at

[†] Institut für Numerische Mathematik, Technische Universität Graz, Email: hyang@math.tugraz.at

conventional time stepping methods. Recently, in [19, 34, 35], robust parallel geometric multigrid methods for parabolic problems using a discontinuous Galerkin space–time finite element approach have been proposed, where a special coarsening strategy is determined by a certain precise criterion. In [13], a multilevel preconditioner for the linear first–order hyperbolic evolution systems by a discretization method combining discontinuous Galerkin in space and Petrov–Galerkin in time has been developed. In [2], a block–diagonal preconditioner resulting from a sparse algebraic wavelet–in–time transformation has been studied, where the individual spatial blocks are preconditioned by standard spatial multigrid in parallel. In [17], a multigrid reduction based optimal–scaling time–parallel method for solving diffusion equations has been presented.

In this work, we follow the Galerkin–Petrov space–time finite element method for the heat equation on arbitrary admissible finite element meshes [42], consisting of tetrahedra and pentachora in 3D and 4D space–time domains, respectively. We use non–tensor product based simplex meshes for the space–time finite element discretization. On the one hand this brings flexibility on the space–time finite element adaptivity using local mesh refinements, and on the other hand this introduces some challenges for solving the resulting large scale linear system of algebraic equations, that is nonsymmetric and positive definite. Unlike space–time geometrical multigrid methods [19], where semi–coarsening in time or full–coarsening in space–time can be determined by a precise criterion, we are not able to distinguish the spatial and temporal contributions easily on arbitrary finite element meshes.

Considering unstructured grids, that may come from local adaptivity, we use the algebraic multigrid (AMG [10, 12, 37, 45]) preconditioned GMRES method [38] to solve the nonsymmetric linear system. An important issue is how to select coarse grids properly in order to effectively interpolate the algebraically smoothing errors on the fine grids. As experimental tests of possible coarsening strategies for such nonsymmetric and positive definite systems, we implement several known methods as representatives of coarsening strategies in both classical Ruge–Stüben type and modern AMG methods, that are based on pure matrix–graph [26], a greedy coarse–grid selection [30], compatible relaxation [11], and Petrov–Galerkin smoothed aggregation [49]. Here, we mainly focus on performance comparison of different AMG methods for the adaptive space–time finite element equations. We mention here other well known coarsening methods, e.g., the classical Ruge–Stüben coarsening [37], the independent set–based coarse–grid selection [21, 1], and the Schur complement approach [51], to name a few.

The remainder of this paper is organized as follows: In Section 2, we present the adaptive space–time finite element approach while Section 3 deals with the algebraic multigrid methods for the solution of the related large scale linear systems. Some numerical experiments are described in Section 4. Finally, some conclusions are drawn in Section 5.

2 Adaptive space–time finite element methods

2.1 A space–time finite element discretization

Recently, a space–time finite element method has been analyzed in [42] for solving the Dirichlet boundary value problem for the heat equation,

$$\alpha \partial_t u(x,t) - \Delta_x u(x,t) = f(x,t) \quad \text{for } (x,t) \in Q := \Omega \times (0,T), \tag{1}$$

with given boundary and initial conditions, i.e., $u(x,t) = 0$ for $(x,t) \in \Sigma := \partial\Omega \times (0,T)$, and $u(x,0) = u_0(x)$ for $x \in \Omega$, respectively. Here, $\Omega \subset \mathbb{R}^n$, and therefore $Q \subset \mathbb{R}^{n+1}$, $n = 2,3$, is a bounded Lipschitz domain, and $\alpha \in \mathbb{R}_+$ is the heat capacity constant.

The Galerkin–Petrov variational formulation for the Dirichlet boundary value problem (1) is to find $\overline{u} \in X := \{v \in L^2(0,T;H^1_0(\Omega)) \cap H^1(0,T;H^{-1}(\Omega)), v(x,0) = 0 \text{ for } x \in \Omega\}$ such that

$$a(\overline{u}, v) = \langle f, v\rangle - a(\overline{u}_0, v) \tag{2}$$

is satisfied for all $v \in Y := L^2(0,T;H^1_0(\Omega))$, where

$$\begin{aligned} a(u,v) &:= \int_0^T \int_\Omega \Big[\alpha \partial_t u(x,t) v(x,t) + \nabla_x u(x,t)\cdot \nabla_x v(x,t)\Big]\, dx\, dt, \\ \langle f, v\rangle &:= \int_0^T \int_\Omega f(x,t) v(x,t)\, dx\, dt, \end{aligned}$$

and $\overline{u}_0 \in L^2(0,T;H^1_0(\Omega)) \cap H^1(0,T;H^{-1}(\Omega))$ denotes an arbitrary but fixed extension of the initial datum $u_0 \in H^1_0(\Omega)$, and $f \in L^2(0,T;H^{-1}(\Omega))$. Existence and uniqueness of the solution to (2) is provided in [42], see also [41, 47]. We mention that for $v \in X$, we have

$$\begin{aligned} a(v,v) &= \int_0^T \int_\Omega \partial_t v v \, dx\, dt + \int_0^T \int_\Omega \nabla_x v \cdot \nabla_x v \, dx\, dt \\ &= \frac{1}{2}\int_0^T \frac{d}{dt}\|v\|^2_{L^2(\Omega)} dt + \|v\|^2_{L^2(0,T;H^1_0(\Omega))} \\ &= \frac{1}{2}\|v(T)\|^2_{L^2(\Omega)} + \|v\|^2_{L^2(0,T;H^1_0(\Omega))} \\ &\geq \|v\|^2_{L^2(0,T;H^1_0(\Omega))}, \end{aligned}$$

which implies later that the system matrix is positive definite.

The related discrete Galerkin–Petrov variational formulation is to find $\overline{u}_h \in X_h \subset X$ such that

$$a(\overline{u}_h, v_h) = \langle f, v_h\rangle - a(\overline{u}_0, v_h) \tag{3}$$

is satisfied for all $v_h \in Y_h \subset Y$ where we assume $X_h \subset Y_h$. A standard stability and error analysis for the space–time finite element method was concluded from a discrete inf–sup condition as shown in [42]. In particular, the space–time cylinder Q is decomposed into admissible and shape regular finite elements q_ℓ, i.e. $Q_h = \cup_{\ell=1}^N \overline{q}_\ell$. For simplicity, we assume that Ω is polygonal ($n = 2$) or polyhedral ($n = 3$) bounded, i.e. $\overline{Q} = Q_h$. The finite element spaces are given by $X_h = S_h^1(Q_h) \cap X$ and $Y_h = X_h$ with $S_h^1(Q_h) = \text{span}\{\varphi_i\}_{i=1}^M$ being the span of piecewise linear and continuous basis functions φ_i. The following energy error estimate is proved in [42],

$$\|\overline{u} - \overline{u}_h\|_{L^2(0,T;H_0^1(\Omega))} \leq c\, h\, |\overline{u}|_{H^2(Q)}, \tag{4}$$

where $\overline{u} \in X$ and $\overline{u}_h \in X_h$ denote the unique solutions of the variational problems (2) and (3), respectively, $c > 0$ is a constant independent of the mesh size h, and we assume $\overline{u} \in H^2(Q)$.

In comparison with other space–time methods, this approach is very suitable for the development of h–adaptivity simultaneously in space and time. However, the solution of the global linear system in space and time requires the use of some efficient solver. Here we propose to use a GMRES method which is preconditioned by algebraic multigrid methods.

2.2 A heuristic residual based error indicator

Let $\overline{u}_h \in X_h$ be the space–time finite element solution of the variational problem (3) implying $u_h := \overline{u}_0 + \overline{u}_h$ for which we can define the local residuals

$$R_{q_\ell}(u_h) := [f + \Delta_x u_h - \alpha \partial_t u_h]|_{q_l}$$

on each tetrahedral/pentachoral element q_ℓ, and the jumps of the normal flux

$$J_\gamma(u_h) := [n_x \cdot \nabla_x u_h + n'_x \cdot \nabla_x u'_h]\,|_\gamma$$

in the spatial direction across any inner boundary γ shared by q_ℓ and its neighbouring element q'_ℓ. Here, n_x and n'_x denote the spatial components of the unit outward normal vectors to the inner boundary $\gamma \subset \partial q_\ell$ and $\gamma \subset \partial q'_\ell$, respectively, u'_h the discrete solution on the neighbouring element q'_l. Then the local error indicator on each element q_ℓ is defined as

$$\eta_{q_\ell} = \left\{ c_1\, h_{q_\ell}^2\, \|R_{q_\ell}(u_h)\|_{L^2(q_\ell)}^2 + c_2\, h_{q_\ell}\, \|J_\gamma(u_h)\|_{L^2(\partial q_\ell)}^2 \right\}^{\frac{1}{2}}, \tag{5}$$

with suitably chosen positive constants c_1, c_2, that may depend on the model problem and domain shape, and where h_{q_ℓ} denotes the local mesh size. For more details, we refer to our recent work [43]. In comparison with more conventional adaptive methods for time–dependent problems, see, e.g., [16, 40, 33], our method allows to perform the spatial and temporal adaptivity simultaneously.

2.3 Adaptive mesh refinement methods in 3D and 4D

The following local adaptive refinement methods for space–time meshes in 3D and 4D are adopted in this work, namely, 3DBey—3D tetrahedral grid refinement using octasection by J. Bey [7], 3DArnold — 3D tetrahedral grid refinement using bisection by D. N. Arnold et al. [4], 3DStevenson — 3D tetrahedral grid refinement using bisection by R. Stevenson [44], and 4DStevenson — 4D pentachoral grid refinement using bisection by R. Stevenson [44]. While they show very different local refinement patterns as illustrated in the numerical examples, all these methods keep shape regularity through the refining procedure.

2.4 The adaptive space–time finite element loop

The adaptive loop in the space–time finite element method follows the one in the standard adaptive finite element approach, see, e.g., [50], that consists of the following four main steps: Given a conforming decomposition Q_0 at the initial mesh level $k = 0$,

1. SOLVE: Solve the discrete problem (3) on the adaptive mesh level k, stop if the solution is accurate enough,
2. ESTIMATE: Compute the local error indicators (5) on each element q_ℓ,
3. MARK: Mark the elements for refinement using a proper marking strategy,
4. REFINE: Perform the local mesh refinement using octasection or bisection, increase level $k := k + 1$, obtain the conforming decomposition Q_k, and go to Step 1.

For the module MARK we use the maximal marking strategy: For a given parameter $\vartheta \in [0, 1]$, mark all elements q_k that fulfill

$$\eta_{q_k} \geq \vartheta \max_{\ell=1\ldots,M_k} \eta_{q_\ell}, \tag{6}$$

where M_k denotes the total number of space–time finite elements on the current level k. Those marked and the affected neighbouring elements will be refined on the next level $k + 1$. In our numerical experiments, $\vartheta = 0.5$. We mention that $\vartheta = 0$ corresponds to a conforming uniform refinement in the adaptive 3DBey, 3DStevenson and 4DStevenson methods.

3 Algebraic multigrid methods

We aim to solve the following large sparse $n \times n$ linear system of algebraic equations

$$Ax = b, \tag{7}$$

arising from the space–time finite element variational formulation (3). It is clear from [42] that the matrix A is nonsymmetric and positive definite. The linear system (7) is solved by the preconditioned GMRES method with different AMG preconditioners. In particular, we use two V(1,1)–cycles with one pre– and post–smoothing step as a preconditioner in the GMRES method. We do not assume any underlying mesh information except the stiffness matrix A. The coarsening and interpolation will be constructed based on A in a purely algebraic manner. For simplicity, we assume that all matrices and vectors in equations are on the same AMG level.

3.1 Smoother

In particular, we employ one sweep of the Kaczmarz relaxation scheme [23] as pre– and postsmoother: Let x^0 be a given initial guess, for $i = 1, ..., n$,

$$x^i = x^{i-1} + \frac{b_i - \langle A_i, x^{i-1} \rangle}{\|A_i\|_{l_2}^2} A_i, \tag{8}$$

with A_i being the ith row of A presented as a column vector, and b_i the ith component of b. The algebraic multigrid smoothing property of the Kaczmarz relaxation scheme for even more general nonsymmetric matrices has been discussed in [8, 36]. Although for a small heat conductivity parameter α, one sweep of the symmetric Gauss–Seidel method, i.e. one forward on the down cycle plus one backward on the up cycle, is sufficient, but it fails when α becomes large. Therefore, in all of our numerical experiments with varying α, we use the Kaczmarz relaxation scheme for smoothing, because it is a more robust method.

3.2 Coarsening

3.2.1 Pure matrix–graph (AMG_Graph)

Red–black coloring (Algorithm 2 in [26]) belongs to a class of simplest pure matrix–graph based AMG coarsening strategies:

1. Start with a black degree of freedom (Dof) (coarse),
2. all neighbouring degrees of freedom (Dofs) are marked in red color (fine),
3. go to Step 1 if some Dofs are not marked.

Then the fine Dofs (red) are interpolated by taking an average over their neighbouring coarse Dofs (black), which implies a linear interpolation operator P. It is easy to see that this simple strategy yields a very aggressive coarsening method, and rather low operator and grid complexities since it treats all connection equally without taking into account the strength of a connection in the classical Ruge–Stüben AMG [12, 37]. Such a coarsening strategy has also been applied to the AMG method for solving some coupled fluid–structure interaction problems [28].

3.2.2 A greedy coarse–grid selection (AMG_Greedy)

In the conventional AMG method, the strength of a connection has to be taken into account and only strong connections are to be considered in the definition of the interpolation operator. In particular, we consider a special greedy partitioning algorithm (Algorithm 3 in [30]), in which a dynamic measure

$$\gamma_i = \frac{|A_{ii}|}{\displaystyle\sum_{j \in F \cup U} |A_{ij}|} \tag{9}$$

is used to decide about the diagonal dominance of row i among those rows already selected as F–Dofs or undesignated U–Dofs. For a given threshold

$\gamma \in (\frac{1}{2}, 1]$, make an initial selection such that the Dofs with $\gamma_i > \gamma$ are selected as fine Dofs through all Dofs:

1. Select a Dof with minimal γ_i as a coarse Dof and remove it from the undecided Dof set,
2. update γ_i of all neighbouring Dofs of the previously selected coarse Dof, add neighbours with $\gamma_i > \gamma$ to the fine Dof set,
3. go to Step 1 if some Dofs are not selected.

In the AMG coarsening this greedy algorithm needs to be augmented by the second pass [12, 37] , changing certain F–Dofs to C–Dofs in order to properly define the classical interpolation operator P [12] for the F–Dofs. The set of Dofs that a Dof i strongly depends on is defined as in the classical Ruge–Stüben AMG:

$$S_i = \left\{ j : i \neq j \text{ and } -A_{ij} \geq \beta \max_{k \neq i} \{-A_{ik}\} \right\} \tag{10}$$

with β being a properly chosen parameter, e.g., $\beta = 0.25$ typically for second order elliptic problems.

3.2.3 Compatible relaxation (AMG_CR)

The concept of compatible relaxation for selecting coarse Dofs was proposed in [9]. The compatible relaxation acts as a modified relaxation scheme, in which the coarse variable stays invariant. It was shown in [18] that the quality of the set of coarse variables is measured by the convergence rate of compatible relaxation. More intuitively, the Dofs on which the smoothing is inefficient tend to be selected as coarse Dofs. The compatible relaxation is fundamentally different from the classical AMG method and does not depend on a strength of connection measure. In particular, we use the coarsening algorithm (Algorithm 3.1 in [11]) to select the coarse Dofs:

1. Run compatible relaxation on the homogeneous equation $Ae = 0$ with a constant vector as the initial guess, that represents a nonzero vector of a near null–space of the operator A,
2. compute a coarse Dof candidate set according to the candidate set measures (when the relative error at that Dof is larger than a given threshold),
3. add the independent set of candidate sets to the coarse Dof set,
4. go to Step 1 if the compatible relaxation convergence factor is above a threshhold.

After selection of coarse Dofs, we run the second pass as in classical AMG [12, 37] and construct the interpolation (prolongation) operator as in Section 3.2.2. The interpolation truncation [11, 20, 46] may be needed in order to reduce the operator complexity. In particular, we use a simple truncation approach as proposed in [46] to drop some small entries in the constructed interpolation operator.

3.2.4 Smoothed aggregation (AMG_SA)

In the smoothed aggregation AMG [39, 48, 49], the interpolation operator is constructed by smoothing a tentative interpolation operator (e.g., piecewise constant) on the Dof decomposition into small disjoint subsets. In particular, we use the greedy aggregation algorithm (Algorithm 2 in [49]) to form aggregates:

1. Collect an unaggregated node (as root-node) only if all of its strongly coupled neighboring nodes are unaggregated, and use this set to form an aggregate G_i, repeat until no new aggregate is formed,
2. add each remaining node to an existing aggregate, to which it is strongly coupled.

In a conventional way, a node j is in the strongly coupled neighborhood of i if

$$|A_{ij}| \geq \varepsilon \sqrt{|A_{ii}A_{jj}|} \tag{11}$$

is satisfied with $\varepsilon \in [0,1)$ being a chosen parameter for certain symmetric positive definite problems; see, e.g., [31, 49]. From the definition (11) it is obvious that this measure becomes nonsymmetric for nonsymmetric problems. Smaller ε tend to generate bigger aggregates containing more nodes, which may lead to too few coarse Dof selections, lower operator and grid complexity, but slower AMG convergence. While larger ε tend to generate smaller aggregates containing few nodes, which may lead to too many coarse node selections, higher operator and grid complexity, but faster AMG convergence. Thus, there is a trade–off between robustness and efficiency in choosing such an optimal parameter. In the extreme case $\varepsilon = 0$, every nodal connection is treated equally.

Next, a tentative prolongator $\tilde{P}$ is formed as a partition of unity. The tentative restriction operator $\tilde{R}$ is defined as $\tilde{R} = \tilde{P}^T$. We then use the prolongator smoothing and restriction smoothing to smooth the tentative operators, which is proposed in [39] for nonsymmetric problems. In the standard smoothed aggregation, global damping parameters are considered. Here, local damping parameters associated with single basis functions (columns of prolongator) are obtained by minimizing each quadratic scalar function $\|P_j\|^2_{A^TA}$ with P_j being jth column of P. A similar technique has been considered for the restriction operator; see more details in Section 4 [39].

4 Numerical experiments

4.1 A 3D example

As a 3D example, we consider $\Omega = (0,1)^2$ and $T = 1$, i.e., $Q = (0,1)^3$, and the solution u given by

$$u(x,t) = (x_1^2 - x_1)(x_2^2 - x_2)(t^2 - t)e^{-100.0((x_1-0.25)^2+(x_2-0.25)^2+(t-0.25)^2)}.$$

The solution has a "peak" near the point $(0.25, 0.25, 0.25)$ in the space–time domain, where local mesh refinements are needed; it is very "flat" in the remaining region, where only coarse mesh is sufficient; see the left plot in Fig. 1.

As we observe later, this behaviour can be nicely captured using our adaptive methods. Therefore, it is a good example to test our space–time adaptive refinements, and compare them with the uniform refinements. For more tests on our space–time adaptive schemes, we refer to a collection of examples considered in [43]; see also subsection 4.4 for comparison of uniform and adaptive refinements in 3D and 4D with low regularity. We mention that all the numerical tests are performed with our home–made package. It has included the finite element methods, the adaptive mesh refinements, and the AMG methods. The ultimate aim of the work is to provide users with such an open–source in the future. Some related work using different refining strategies for high dimension in the space–time finite element methods can be found in, e.g., [6, 24, 25, 35]. All our numerical tests are performed on a laptop computer with an Intel Core i5 of 1.60GHz and 8 GB memory.

As illustrated in Fig. 1, for the 3DBey and 3DArnold refinement methods, we start from an initial mesh with 125 vertices and 384 tetrahedral elements; while for the 3DStevenson method, we start from an initial mesh with 29 vertices and 60 tetrahedral elements, which fulfills the reflection condition of any two neighbouring elements [44]. The number of degrees of freedom on the finest level is about $200,000$. As an illustration of different adaptive mesh refinement methods, we plot adaptively refined meshes using 3DBey, 3DArnold, and 3DStevenson during one refining step in Fig. 2. We mention that for simplicity, we only consider isotropic mesh refinements in space and time. In some situations, we may face problems where the space–time domain could be completely anisotropic. To handle such problems, we may need anisotropic refinements [3] in temporal and spatial directions. Another approach is to rescale the time variable in order to get comparable length in space and time, as used in [5].

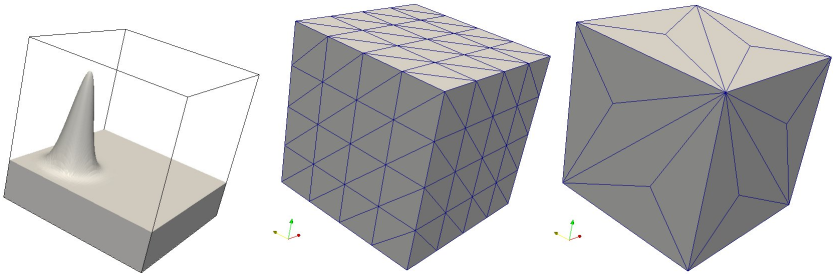

Figure 1: An illustration of the solution (left), and initial meshes for the 3DBey and 3DArnold methods (middle) and the 3DStevenson method (right).

To make a relatively fair comparison, we compare the iteration number and the computational cost in seconds of the preconditioned GMRES method using different AMG preconditioners, namely, AMG_Graph, AMG_Greedy, AMG_CR, and AMG_SA, in order to reach the same accuracy of the numerical solution. The error $e_h := u - u_h$ between the given and the discrete solutions is measured in the $L^2(0,T;H_0^1(\Omega))$–norm. For all the following tests, we use the relative

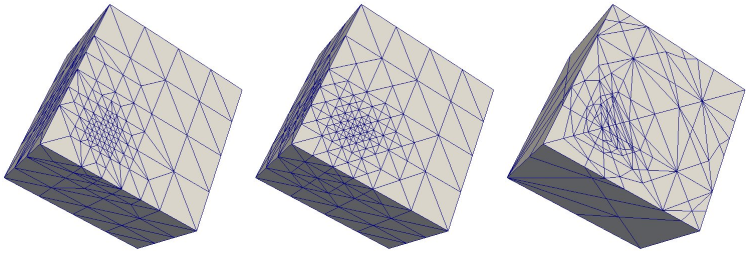

Figure 2: An illustration of the three local mesh refinements at a step of adaptive refinements: 3DBey, 3DArnold, and 3DStevenson (from left to right).

residual error $\varepsilon = 10^{-7}$ as the stopping criterion for GMRES, and run two V(1,1)–cycles in each preconditioning step. We show the AMG performance results for $\alpha = 1, 100$ since the solver is relatively robust for $\alpha = 10$ as well.

The estimated order of convergence (eoc) with respect to the number of degrees of freedom (#Dof) is illustrated in Fig. 3. We observe a linear order of convergence for both the uniform ($\vartheta = 0$) and adaptive refinements as expected. To reach the same accuracy, the adaptive refinement allows us to use much fewer degrees of freedom than the uniform refinement.

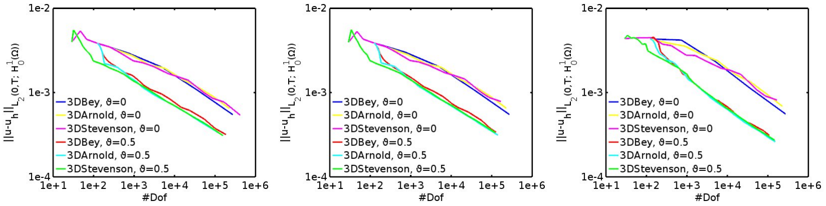

Figure 3: Estimated order of convergence (eoc) for $\alpha = 1, 10, 100$ (from left to right) on both uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) refinements.

The iteration number of the AMG preconditioned GMRES and the corresponding computational times in seconds are illustrated in Fig. 4 (AMG_Graph), Fig. 5 (AMG_Greedy), Fig. 6 (AMG_CR), and Fig. 7 (AMG_SA), respectively. We observe that all the adaptive refinements ($\vartheta = 0.5$) show more efficiency than the uniform refinements ($\vartheta = 0$) by reducing the number of GMRES iterations as well as computational costs. Except the AMG_Graph method, all the other three AMG methods show similar performance with respect to the preconditioned GMRES iterations and computational costs. The AMG_Graph preconditioned GMRES method needs more iterations and computational time. All the AMG methods show relatively good performance with respect to α, i.e., the iteration numbers of AMG preconditioned GMRES method stay in a similar range when α increases. The AMG_Greedy, AMG_CR and AMG_SA preconditioned GMRES methods show relatively good robustness with respect to the mesh discretization parameter, i.e., the iteration numbers stay in a similar range when the mesh is refined. This tells us that the strength of connection becomes important here in the coarsening even though the Poisson operator in space is isotropic. This is because the Poisson operator merely in space becomes a highly anisotropic operator in space and time, i.e., $-\Delta_x u(x,t) - \kappa \partial^2 u(x,t)/\partial t^2$ with $\kappa \to 0$. In addition, the first order time–derivative is involved as a kind of convection term (see [5]), which makes the coarsening even harder than classical diffusion problems.

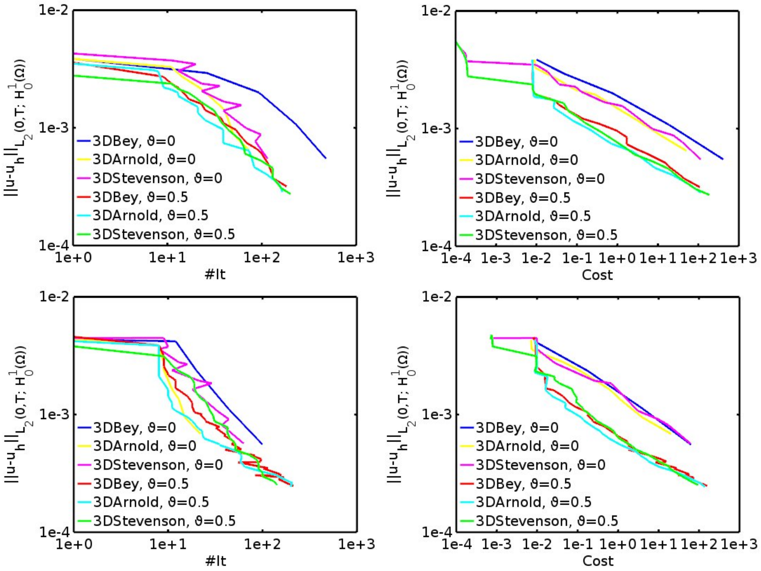

Figure 4: Comparison of AMG_Graph preconditioned GMRES iterations (#It) (left) and costs in seconds (right) with both the uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) refinements for $\alpha = 1$ (up) and 100 (down).

We observe that the AMG methods on the adaptive refinements show the superiority over the uniform refinements in terms of the error reduction in $L^2(0,T;H_0^1(\Omega))$–norm. That is mainly because the problem size of the adaptive refinements is much smaller than the uniform refinements to reach the same accuracy, which saves computational cost. We further observe that the AMG methods show comparable performance on both the uniform and adaptive refinements in terms of the number of degrees of freedom. We compare the iterations and cost in second for the AMG_Greedy preconditioned GMRES method on uniform and adaptive 3DStevenson refinements in Fig. 8. From this comparison, it is easy to see that the AMG method on both the refinements needs similar iterations and cost.

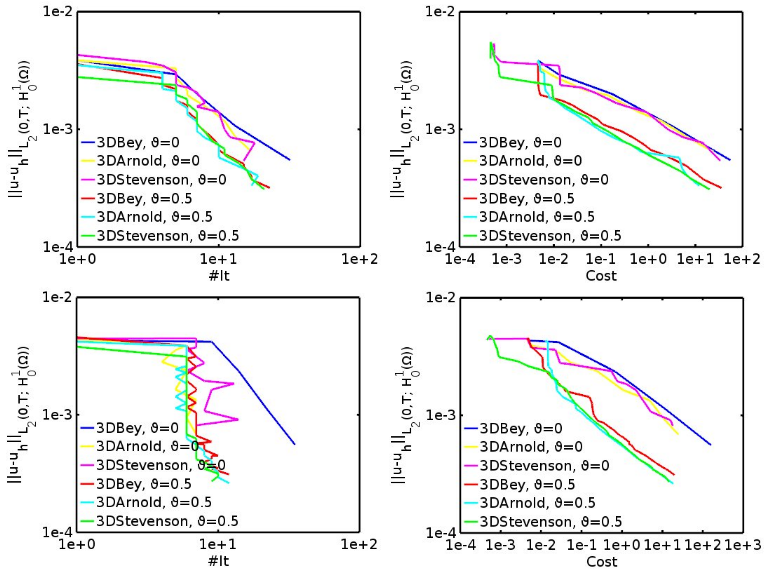

Figure 5: Comparison of AMG_Greedy preconditioned GMRES iterations (#It) (left) and costs in seconds (right) with both the uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) refinements for $\alpha = 1$ (up) and 100 (down).

4.2 A 4D example

As a 4D example, we consider $\Omega = (0,1)^3$ and $T = 1$, i.e., $Q = (0,1)^4$, and the given solution

$$u(x,t) = (x_1^2-x_1)(x_2^2-x_2)(x_3^2-x_3)(t^2-t)e^{-100.0((x_1-0.25)^2+(x_2-0.25)^2+(x_3-0.25)^2+(t-0.25)^2)}.$$

The initial mesh contains 178 vertices and 960 pentachorona. The uniform and adaptive refinements are realized by using the 4DStevenson method and choosing $\vartheta = 0$ and 0.5, respectively.

The estimated order of convergence (eoc) for $\alpha = 1$, 10 and 100 using both the uniform refinement ($\vartheta = 0$) and adaptive refinement ($\vartheta = 0.5$) is demonstrated in Fig. 9. We observe a linear order of convergence (in comparison with the solid black line in Fig. 9) and efficiency of the adaptive refinement in saving number of degrees of freedom. For the uniform refinements, we need to perform further refinements in order to see the linear order of convergence. That is because of the “peak” in the solution near the point $(0.25, 0.25, 0.25, 0.25)$ as in the 3D example, which requires sufficiently many degrees of freedom to resolve it.

For the tests in this section, we use the relative residual error $\varepsilon = 10^{-7}$ as the stopping criterion for GMRES, and run two V(1,1)–cycles in each preconditioning step. The iteration number of the preconditioned GMRES method and computational time measured in seconds using different AMG preconditioners

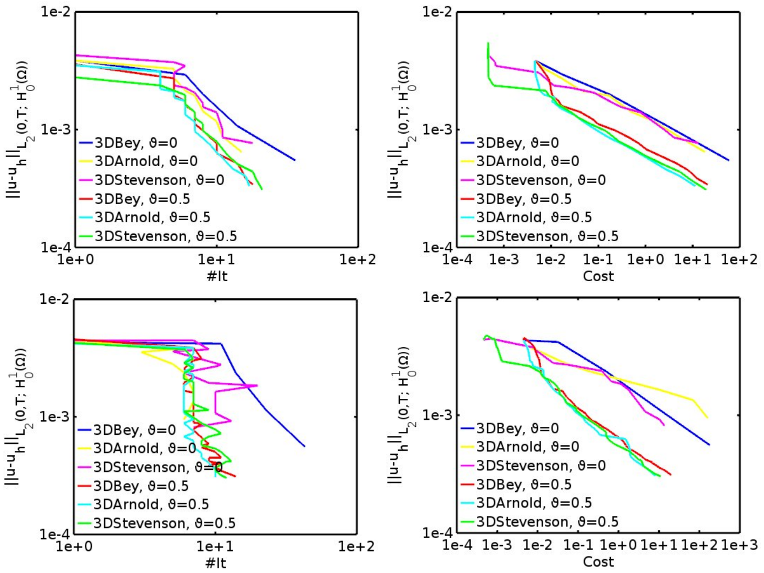

Figure 6: Comparison of AMG_CR preconditioned GMRES iterations (#It) (left) and costs in seconds (right) with both the uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) refinements for $\alpha = 1$ (up) and 100 (down).

(AMG_Greedy, AMG_CR, AMG_SA) are shown in Fig. 10. We mention that the poor performance of the AMG_Graph method carries over from 3D to 4D. Therefore, we do not show the 4D result using this method. In addition, we show results for $\alpha = 1, 100$. The solver is relatively robust for $\alpha = 10$ as well.

From our numerical experiments, we observe that the iteration numbers needed by the AMG_Greedy and AMG_CR methods stay in a relatively small range (about 10 on the finest level); the iteration number of the AMG_SA method slightly increases; the iteration number of the AMG_Graph method drastically increases. The AMG_Greedy and AMG_CR methods result in very similar AMG performance. That is because similar operator and grid complexities are obtained using coarsening of both methods. In addition, they show relatively good robustness with respect to the mesh discretization and material parameters. In all of the AMG methods, the adaptive refinement leads us to more efficiency than the uniform refinement by reducing both the number of preconditioned GMRES iterations and computational time. Finally, the performance of the AMG_Greedy and AMG_CR methods slightly drops with respect to a larger heat capacity constant α due to a growing operator complexity when α becomes larger. It will be further studied in future work to reduce the operator complexity for large α.

As in the 3D example, we further observe that the AMG methods show comparable performance on both the uniform and adaptive refinements in terms of the number of degrees of freedom; see the comparison in Fig. 11 using the

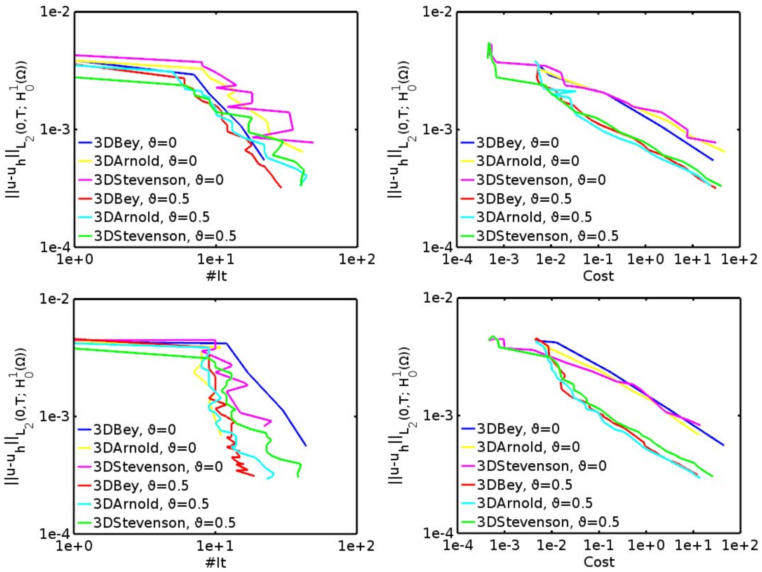

Figure 7: Comparison of AMG_SA preconditioned GMRES iterations (#It) (left) and costs in seconds (right) with both the uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) refinements for $\alpha = 1$ (up) and 100 (down).

AMG_Greedy preconditioned GMRES method on 4DStevenson refinements.

4.3 Further discussion on AMG performance

We perform further tests with AMG_Greedy and AMG_SA methods on 3DStevenson and 4DStevenson uniform refinements ($\vartheta = 0$) since AMG_CR has similar performance with AMG_Greedy, and AMG_Graph has poor performance in comparison with other methods. In the following tests, we use the relative residual error $\varepsilon = 10^{-7}$ as the stopping criterion for the AMG preconditioned GMRES methods. We use one V(1,1)–cycle and one V(2,2)–cycle as preconditioners, respectively.

For the 3D example as considered in Section 4.1, the number of AMG_Greedy and AMG_SA preconditioned GMRES iterations and cost in second with one V(1,1)–cycle and one V(2,2)–cycle are compared in Table 1 and Table 2, respectively. Analogously, the number of GMRES iterations and cost in second for the 4D example as considered in Section 4.2 are demonstrated in Table 3 and Table 4, respectively. On the one hand, we observe a mild mesh dependence of the AMG methods. On the other hand, as we pointed out earlier, for large α, the AMG_Greedy coarsening may lead to relatively high operator complexity, which requires further investigation on the improvement of the coarsening algorithm. With more pre– and post–smoothing steps, we may further reduce the number of iterations. However, the AMG performance may drop due to cost of

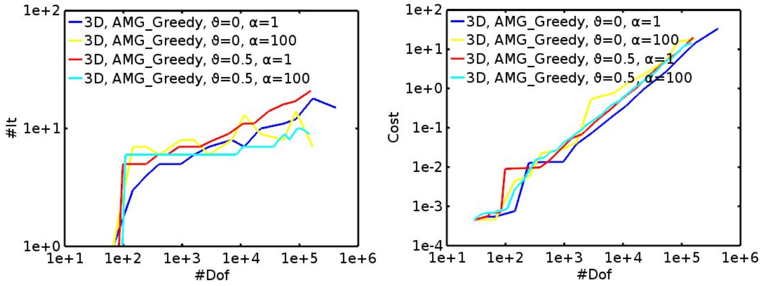

Figure 8: Number of AMG_Greedy preconditioned GMRES iterations (left) and cost in second (right) with respect to degrees of freedom for $\alpha = 1, 100$ on both uniform and adaptive 3DStevenson refinements.

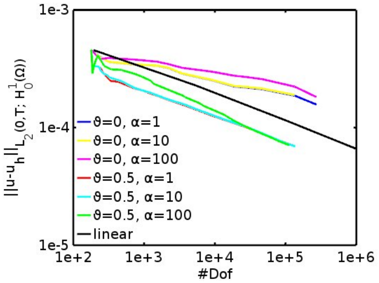

Figure 9: Estimated order of convergence (eoc) for $\alpha = 1, 10, 100$ on both uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) 4DStevenson refinements.

the Kaczmarz relaxation scheme.

4.4 Further convergence study with low regularity

In this subsection, we further compare the uniform and adaptive refinements when the solution has less regularity with respect to both space and time. We use the heat equation (1) with $\alpha = 1$. As a 3D example, we adopt the solution u given by

$$u(x,t) = \sin\left(\frac{1}{\frac{1}{10\pi} + \sqrt{x_1^2 + x_2^2 + t^2}}\right),$$

which is non–differentiable and singular at the original point $(0,0,0)$; see the left and right plots as an illustration in Fig. 12. For the 3DStevenson refinement, we use an initial mesh as shown in the right plot of Fig. 12, which results from a special decomposition of the mesh in the middle of Fig. 1 and fulfills certain reflection conditions required in [44]. For the 3DBey and 3DArnold refinements, we use the same initial mesh as in the middle of Fig. 1. The adaptive mesh refinements near the original point are shown in Fig. 13. Because of less regularity of the solution, the linear order of convergence will not be achieved using

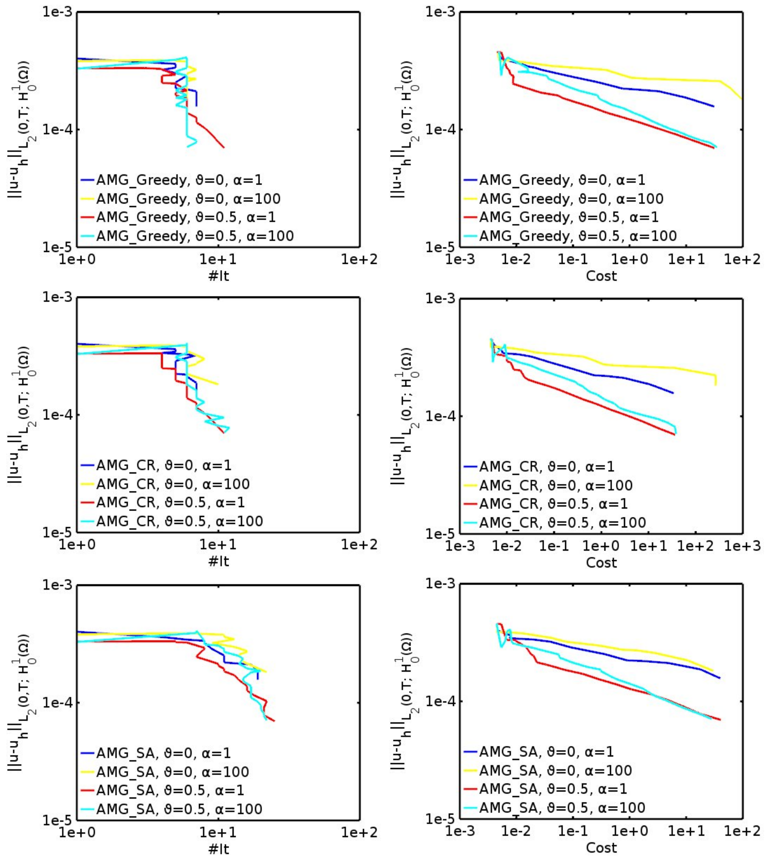

Figure 10: Comparison of AMG_Greedy (up), AMG_CR (middle) and AMG_SA (down) preconditioned GMRES iterations (#It) (left) and cost in second (right) to reach the same accuracy using both the uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) refinements for $\alpha = 1, 100$.

our space–time finite element methods on the uniform refinements; however, all the adaptive refinements recover the linear order of convergence. For this 3D example, the estimated order of convergence on both uniform and adaptive refinements is demonstrated in the left plot of Fig. 14.

Analogously, we further consider a 4D example with the solution u given by

$$u(x,t) = \sin\left(\frac{1}{\frac{1}{10\pi} + \sqrt{x_1^2 + x_2^2 + x_3^2 + t^2}}\right),$$

which is singular at the original point $(0, 0, 0, 0)$. We start from the same initial mesh as for the 4D example in subsection 4.2. Due to less regularity of the solution, we do not observe clear convergence order with our space–time finite

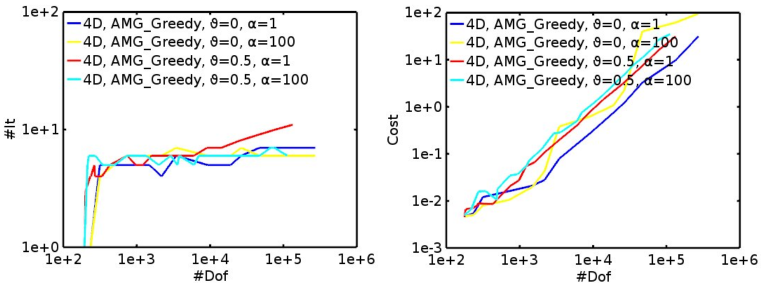

Figure 11: Number of AMG_Greedy preconditioned GMRES iterations (left) and cost in second (right) with respect to degrees of freedom for $\alpha = 1, 100$ on both uniform and adaptive 4DStevenson refinements.

#Dof	$\alpha = 1$		$\alpha = 10$		$\alpha = 100$	
	AMG_Greedy	AMG_SA	AMG_Creedy	AMG_SA	AMG_Greedy	AMG_SA
961	8 (7)	15 (12)	7 (6)	14 (11)	12 (8)	20 (13)
2881	10 (9)	18 (15)	10 (8)	17 (13)	10 (7)	15 (11)
11457	11 (10)	16 (13)	11 (9)	14 (11)	19 (13)	22 (16)
53569	17 (14)	43 (34)	15 (11)	34 (26)	12 (9)	18 (13)
168577	27 (23)	53 (40)	17 (15)	54 (41)	12 (9)	18 (14)

Table 1: Comparison of the GMRES iterations using one V(1,1)–cycle preconditioner and one V(2,2)–cycle preconditioner (in the bracket) on the 3DStevenson uniform refinement.

element methods on the uniform refinement up to nearly 10^6 degrees of freedom. However, with the adaptive refinement, the linear order of convergence is recovered. For this 4D example, the estimated order of convergence on both uniform and adaptive refinements is shown in the right plot of Fig. 14.

5 Conclusions

In this work, we have compared preconditioned GMRES methods for solving the adaptive space–time finite element discretized heat equation in 3D and 4D using different AMG preconditioners, namely, AMG_Graph, AMG_Greedy, AMG_CR and AMG_SA. These methods show relatively good performance with respect to the mesh discretization and material parameters, and local adaptive refinements. On the one hand, we have observed that the iteration numbers of the AMG preconditioned GMRES method stay in a similar range when the mesh is refined. On the other hand, the AMG performance does not deteriorate so much when the heat capacity constant α becomes large.

#Dof	$\alpha = 1$		$\alpha = 10$		$\alpha = 100$	
	AMG_Greedy	AMG_SA	AMG_Greedy	AMG_SA	AMG_Greedy	AMG_SA
961	0.01 (0.02)	0.04 (0.02)	0.01 (0.02)	0.02 (0.03)	0.02 (0.03)	0.02 (0.02)
2881	0.06 (0.08)	0.1 (0.1)	0.06 (0.07)	0.08 (0.09)	0.4 (0.5)	0.11 (0.07)
11457	0.3 (0.4)	0.5 (0.6)	0.3 (0.4)	0.4 (0.5)	1.0 (1.1)	0.5 (0.5)
53569	2.4 (2.8)	7.6 (8.8)	2.2 (2.7)	6.3 (7.2)	4.0 (4.9)	2.3 (2.2)
168577	13.6 (15.1)	27.8 (29.5)	9.1 (11.4)	28.8 (30.5)	16.3 (21.0)	10.2 (9.4)

Table 2: Comparison of cost in second of the GMRES iterations using one V(1,1)–cycle preconditioner and one V(2,2)–cycle preconditioner (in the bracket) on the 3DStevenson uniform refinement.

The ongoing work is to further study the coarsening strategies in order to reduce the operator complexity and meanwhile keep the robustness of the AMG methods for such nonsymmetric and positive definite linear systems of equations. We may consider some nonsymmetric AMG methods for solving the space–time finite element equations, e.g., the so-called root–node based AMG method [31], a combination of classical and aggregation–based multigrid. Further, it is obvious that our ultimate aim is to solve large scale systems of space–time finite element equations on parallel machines for realistic examples, in order to show the advantages of this kind of methods in comparision with classical time marching methods. In fact, the method used in this work is very suitable for parallelizaiton. We may consider parallel AMG methods for solving such problems. In particular, we will need to parallelize the coarsening strategies as in BoomerAMG [21]. Some variants of the Kaczmarz algorithms have been considered in parallelization; see, e.g., [29, 32], which may be adapted to our AMG methods.

Acknowledgements

This work has been supported by the Austrian Science Fund (FWF) under the Grant SFB Mathematical Optimisation and Applications in Biomedical Sciences. We thank two anonymous refereers for their very insightful comments and suggestions on improving the content of this paper.

#Dof	$\alpha = 1$		$\alpha = 10$		$\alpha = 100$	
	AMG_Greedy	AMG_SA	AMG_Creedy	AMG_SA	AMG_Greedy	AMG_SA
715	6 (5)	10 (7)	6 (5)	10 (6)	10 (7)	17 (10)
2185	6 (5)	12 (9)	6 (5)	13 (8)	9 (6)	20 (12)
9225	8 (7)	18 (13)	7 (6)	17 (12)	10 (7)	20 (13)
26593	8 (7)	16 (12)	8 (6)	17 (12)	10 (7)	25 (17)
134113	11 (10)	29 (21)	11 (9)	29 (21)	10 (7)	24 (15)

Table 3: Comparison of the GMRES iterations using one V(1,1)–cycle preconditioner and one V(2,2)–cycle preconditioner (in the bracket) on the 4DStevenson uniform refinement.

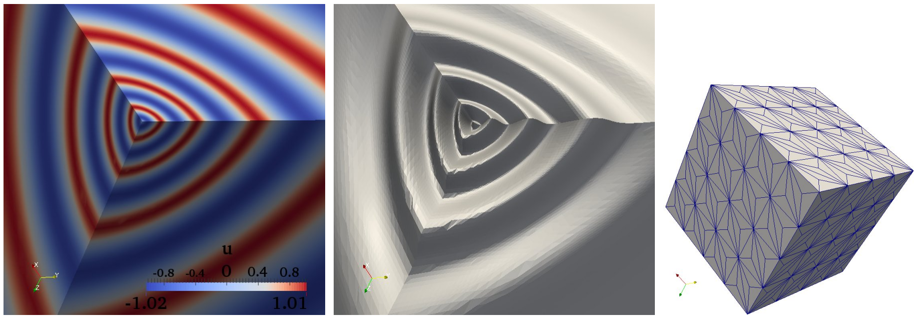

Figure 12: An illustration of solution near the original point $(0,0,0)$ (left), warped by a scalar 0.006 in the direction $(-1,-1,-1)$ (middle), the initial mesh for the 3DStevenson refinement (right).

References

[1] D. M. Alber and L. N. Olson. Coarsening invariance and bucket-sorted independent sets for algebraic multigrid. *Electron. Trans. Numer. Anal.*, 37:367–385, 2010.

[2] R. Andreev. Wavelet–in–time multigrid–in–space preconditioning of parabolic evolution equations. *SIAM J. Sci. Comput.*, 38(1):A216–A242, 2016.

[3] T. Apel. *Anisotropic finite elements: Local estimates and applications.* Teubner, Stuttgart, 1999.

[4] D. N. Arnold, A. Mukherjee, and L. Pouly. Locally adapted tetrahedral meshes using bisection. *SIAM J. Sci. Comput.*, 22(2):431–448, 2000.

#Dof	$\alpha = 1$		$\alpha = 10$		$\alpha = 100$	
	AMG_Greedy	AMG_SA	AMG_Greedy	AMG_SA	AMG_Greedy	AMG_SA
715	0.01 (0.02)	0.01 (0.02)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.02 (0.01)
2185	0.03 (0.03)	0.04 (0.04)	0.04 (0.04)	0.05 (0.05)	0.04 (0.04)	0.06 (0.03)
9225	0.2 (0.3)	0.4 (0.4)	0.3 (0.3)	0.3 (0.3)	0.6 (0.6)	0.3 (0.2)
26593	1.2 (1.4)	1.6 (1.7)	1.3 (1.5)	1.6 (1.6)	3.1 (2.8)	1.4 (1.3)
134113	8.1 (12.7)	15.2 (16.2)	9.0 (11.3)	14.8 (15.7)	266.0 (268.0)	9.8 (9.0)

Table 4: Comparison of cost in second of the GMRES iterations using one V(1,1)–cycle preconditioner and one V(2,2)–cycle preconditioner (in the bracket) on the 4DStevenson uniform refinement.

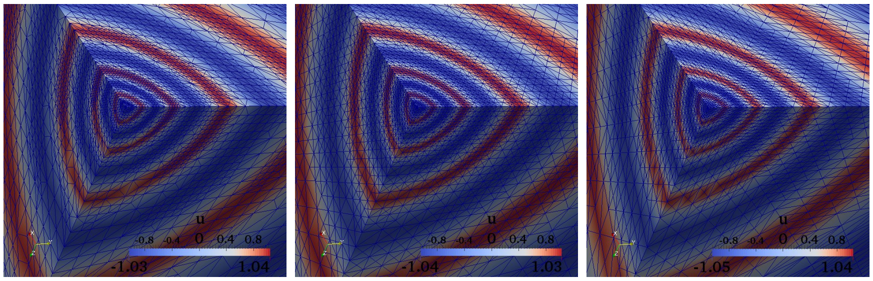

Figure 13: An illustration of adaptive refinements: Left (3DBey, 13rd step), middle (3DArnold, 31st step) and right (3DStevenson, 27th step).

[5] R. E. Bank, P. S. Vassilevski, and L. T. Zikatanov. Arbitrary dimension convection–diffusion schemes for space–time discretizations. *J. Comput. Appl. Math.*, 310:19–31, 2017.

[6] M. Behr. Simplex space–time meshes in finite element simulations. *Int. J. Numer. Meth. Fluids*, 57(9):1421–1434, 2008.

[7] J. Bey. Tetrahedral grid refinement. *Computing*, 55(4):355–378, 1995.

[8] A. Brandt. Algebraic multigrid theory: The symmetric case. *Appl. Math. Comput.*, 19(1):23–56, 1986.

[9] A. Brandt. General highly accurate algebraic coarsening. *Electron. Trans. Numer. Anal.*, 10:1–20, 2000.

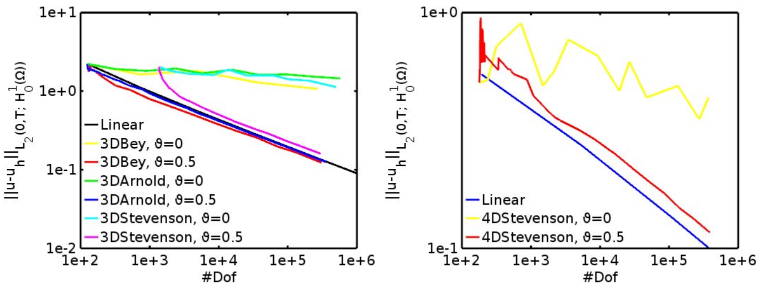

Figure 14: Estimated order of convergence (eoc) on both uniform ($\vartheta = 0$) and adaptive ($\vartheta = 0.5$) 3DBey, 3DArnold and 3DStevenson refinements (left), and 4DStevenson refinements (right).

[10] A. Brandt, S. F. McCormick, and J. W. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and Its Applications*, pages 257–284, Cambridge, 1984. Cambridge University Press.

[11] J. J. Brannick and R. D. Falgout. Compatible relaxation and coarsening in algebraic multigrid. *SIAM J. Sci. Comput.*, 32(3):1393–1416, 2010.

[12] W. Briggs, V. Henson, and S. McCormick. *A multigrid tutorial.* SIAM, Philadelphia, second edition, 2000.

[13] W. Dörfler, S. Findeisen, and C. Wieners. Space–time discontinuous Galerkin discretizations for linear first–order hyperbolic evolution systems. *Comput. Methods Appl. Math.*, 16:409–428, 2016.

[14] T. E. Ellis. *Space–time discontinuous Petrov–Galerkin finite elements for transient fluid mechanics.* PhD thesis, University of Texas at Austin, 2016.

[15] T. E. Ellis, L. Demkowicz, and J. Chan. Locally conservative discontinuous Petrov–Galerkin finite elements for fluid problems. *Comput. & Math. Appl.*, 68(11):1530–1549, 2014.

[16] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems I: A linear model problem. *SIAM J. Numer. Anal.*, 28(1):43–77, 1991.

[17] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder. Parallel time integration with multigrid. *SIAM J. Sci. Comput.*, 36(6):C635–C661, 2014.

[18] R. D. Falgout and P. S. Vassilevski. On generalizing the algebraic multigrid framework. *SIAM J. Numer. Anal.*, 42(4):1669–1693, 2004.

[19] M. J. Gander and M. Neumüller. Analysis of a new space–time parallel multigrid algorithm for parabolic problems. *SIAM J. Sci. Comput.*, 38(4):A2173–A2208, 2016.

[20] H. De Sterck, R. D. Falgout, J. W. Nolting, and U. M. Yang. Distance-two interpolation for parallel algebraic multigrid. *Numer. Linear Algebra Appl.*, 15(2-3):115–139, 2008.

[21] V. E. Henson and U. M. Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41(1):155–177, 2002.

[22] T. J. R. Hughes and G. M. Hulbert. Space–time finite element methods for elastodynamics: Formulations and error estimates. *Comput. Methods Appl. Math.*, 66(3):339–363, 1988.

[23] S. Kaczmarz. Angenäherte Auflösung von Systemen linearer Gleichungen. *Bull. Acad. Polon. Sci. et Lett. A*, 35:355–357, 1937.

[24] E. Karabelas and M. Neumüller. Generating admissible space-time meshes for moving domains in $d+1$-dimensions, 2015. arXiv:1505.03973 [math.NA].

[25] V. Karyofylli, M. Frings, S. Elgeti, and M. Behr. Simplex space-time meshes in two-phase flow simulations. *Int. J. Numer. Meth. Fluids*, 2017. Accepted.

[26] F. Kickinger. Algebraic multigrid for discrete elliptic second-order problems. In W. Hackbush, editor, *Multigrid Methods V. Proceedings of the 5th European Multigrid conference, Lecture Notes in Computational Sciences and Engineering, vol. 3*, pages 157–172, Berlin, Heidelberg, 1998. Springer.

[27] U. Langer, S. E. Moore, and M. Neumüller. Space–time isogeometric analysis of parabolic evolution problems. *Comput. Methods Appl. Math.*, 306:342–363, 2016.

[28] U. Langer and H. Yang. Robust and efficient monolithic fluid–structure–interaction solvers. *Int. J. Numer. Meth. Engng.*, 108(4):303–325, 2016.

[29] J. Liu, S. J. Wright, and S. Sridhar. An asynchronous parallel randomized Kaczmarz algorithm, 2014. arXiv:1401.4780 [math.NA].

[30] S. MacLachlan and Y. Saad. A greedy strategy for coarse–grid selection. *SIAM J. Sci. Comput.*, 29(5):1825–1853, 2007.

[31] T. A. Manteuffel, L. N. Olson, J. B. Schroder, and B. S. Southworth. A root-node based algebraic multigrid method. *SIAM J. Sci. Comput.*, 2016. Accepted.

[32] J. Martínez and R. J. B. De Raimundo. Parallel and sequential Kaczmarz methods for solving underdetermined nonlinear equations. *J. Comput. Appl. Math.*, 15(3):311–321, 1986.

[33] P. K. Moore. A posteriori error estimation with finite element semi– and fully discrete methods for nonlinear parabolic equations in one space dimension. *SIAM J. Numer. Anal.*, 31(1):149–169, 1994.

[34] M. Neumüller. *Space–time methods: Fast solvers and applications.* PhD thesis, TU Graz, 2013.

[35] M. Neumüller and O. Steinbach. Refinement of flexible space–time finite element meshes and discontinuous Galerkin methods. *Comput. Vis. Sci.*, 14:189–205, 2011.

[36] C. Popa. Algebraic multigrid smoothing property of Kaczmarz's relaxation for general rectangular linear systems. *Electron. Trans. Numer. Anal.*, 29:150–162, 2007.

[37] J. W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. F. McCormick, editor, *Multigrid Methods*, pages 73–130. 1987.

[38] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7(3):856–869, 1986.

[39] M. Sala and R. S. Tuminaro. A new Petrov–Galerkin smoothed aggregation preconditioner for nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 31(1):143–166, 2008.

[40] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space–time finite element discretizations of parabolic equations. *SIAM J. Sci. Comput.*, 30(1):369–393, 2008.

[41] C. Schwab and R. Stevenson. Space–time adaptive wavelet methods for parabolic evolution problems. *Math. Comp.*, 78(267):1293–1318, 2009.

[42] O. Steinbach. Space–time finite element methods for parabolic problems. *Comput. Methods Appl. Math.*, 15:551–566, 2015.

[43] O. Steinbach and H. Yang. An adaptive space–time finite element method for solving the heat equation. Technical report, TU Graz, 2017. In preparation.

[44] R. Stevenson. The completion of locally refined simplicial partitions created by bisection. *Math. Comp.*, 77(261):227–241, 2008.

[45] K. Stüben. A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128(1–2):281–309, 2001.

[46] K. Stüben. An introduction to algebraic multigrid. In U. Trottenberg, C. Oosterlee, and A. Schüller, editors, *Multigrid*, pages 413–528. Academic Press, San Diego, 2001.

[47] K. Urban and A. T. Patera. An improved error bound for reduced basis approximation of linear parabolic problems. *Math. Comp.*, 83(288):1599–1615, 2014.

[48] P. Vaněk. Acceleration of convergence of two-level algorithm by smoothing transfer operator. *Applic. Math.*, 37:265–274, 1992.

[49] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196, 1996.

[50] R. Verfürth. *A posteriori error estimation techniques for finite element methods.* Oxford Unversity Press, Oxford, 2013.

[51] T. A. Wiesner, R. S. Tuminaro, W. A. Wall, and M. W. Gee. Multigrid transfers for nonsymmetric systems based on Schur complements and Galerkin projections. *Numer. Linear Algebra Appl.*, 21(3):415–438, 2014.

Erschienene Preprints ab Nummer 2015/1

2015/1 O. Steinbach: Space-time finite element methods for parabolic problems

2015/2 O. Steinbach, G. Unger: Combined boundary integral equations for acoustic scattering-resonance problems problems.

2015/3 C. Erath, G. Of, F.–J. Sayas: A non-symmetric coupling of the finite volume method and the boundary element method

2015/4 U. Langer, M. Schanz, O. Steinbach, W.L. Wendland (eds.): 13th Workshop on Fast Boundary Element Methods in Industrial Applications, Book of Abstracts

2016/1 U. Langer, M. Schanz, O. Steinbach, W.L. Wendland (eds.): 14th Workshop on Fast Boundary Element Methods in Industrial Applications, Book of Abstracts

2016/2 O. Steinbach: Stability of the Laplace single layer boundary integral operator in Sobolev spaces

2017/1 O. Steinbach, H. Yang: An algebraic multigrid method for an adaptive space-time finite element discretization

2017/2 G. Unger: Convergence analysis of a Galerkin boundary element method for electromagnetic eigenvalue problems

2017/3 J. Zapletal, G. Of, M. Merta: Parallel and vectorized implementation of analytic evaluation of boundary integral operators

2017/4 S. Dohr, O. Steinbach: Preconditioned space-time boundary element methods for the one-dimensional heat equation

2017/5 O. Steinbach, H. Yang: Comparison of algebraic multigrid methods for an adaptive space–time finite element discretization of the heat equation in 3D and 4D