Numerical Results for Adaptive (Negative Norm) Constrained First Order System Least Squares Formulations

Andreas Schafelner[a,*], Panayot S. Vassilevski[b,c]

[a] *Doctoral Programm "Computational Mathematics", Johannes Kepler University Linz, Altenberger Strasse 69, 4040 Linz, Austria*
[b] *Fariborz Maseeh Department of Mathematics and Statistics, Portland State University, PO Box 751 (MTH), Portland, OR 97201, USA*
[c] *Center for Applied Scientific Computing, LLNL, Mail stop L-561, Livermore, CA 94550, USA*

Abstract

We perform a followup computational study of the recently proposed space-time first order system least squares (FOSLS) method subject to constraints referred to as CFOSLS where we now combine it with the new capability we have developed, namely, parallel adaptive mesh refinement (AMR) in 4D. The AMR is needed to alleviate the high memory demand in the combined space time domain and also allows general (4D) meshes that better follow the physics in space-time. With an extensive set of computational experiments, performed in parallel, we demonstrate the feasibility of the combined space-time AMR approach in both two space plus time and three space plus time dimensions.

Keywords: CFOSLS, Space-time, Adaptivity, Finite element method

1. Introduction

This study is a continuation of previous work, [8, 11, 15] for discretizations and solvers of time-dependent problems discretized in combined space-time domain. We pursue a constrained first order systems least-squares approach, referred to as CFOSLS [2], which in addition to the more classical FOSLS [1, 4, 9], imposes certain equations (such as divergence) as constraints for better conservation properties of the discretized problem. There has been a substantial interest for combined space-time approaches, especially in the solvers community motivated by the parallel-in-time approach for solving time-dependent PDEs, which is viewed as a feasible approach for better utilization of the computational power of the next generation (exascale) parallel computers. We build upon the

*Corresponding author
Email addresses: andreas.schafelner@dk-compmath.jku.at (Andreas Schafelner), panayot@pdx.edu, vassilevski1@llnl.gov (Panayot S. Vassilevski)

developed general classes of 4D finite element spaces (in fact for the whole 4D de Rham sequence) available in the highly scalable finite element software library MFEM [10]. The results in the present paper are natural followup extension of the results from [15] where now we utilize our newly developed capability of AMR (adaptive mesh refinement) for the same general classes of 4D finite element spaces. The AMR allows the use of general meshes in 4D without having the notion of time-stepping, thus having high resolution only in directions of high variation of the physical quantities modeled by the time dependent PDEs of interest. In this study, we consider parabolic and hyperbolic (transport) PDEs. In the CFOSLS formulation we also exploit the use of weaker, negative Sobolev norms to impose less smoothness requirements on the solution. Overall, our results indicate that the combined space-time AMR CFOSLS approach is feasible; it does capture the physics for problems with anisotropy and generates meshes with no time-stepping behavior, which in principle is an indication for alleviating the global CFL-condition (in the case of transport equation). The CFOSLS approach though, as noted in [15], poses the challenge to the solvers since the FOSLS functionals are not uniformly elliptic and standard multigrid (with geometric coarse spaces) is not algorithmically scalable although our tests do show reasonable performance (in timings). This challenge needs to be addressed further, with possibly exploiting ideas from adaptive AMG that can detect problem anisotropies (cf., e.g., [6] and [13])

The remainder of the paper consists of the following sections. In Section 2 we briefly describe the model problems we consider in this paper. In Section 3, we define the space-time CFOSLS problem and derive a variational formulation. In Section 4, we discretize the variational problems, and introduce the negative norm CFOSLS formulation. In Section 5, we present solvers for the linear systems arising from the finite element discretization. Section 6 contains three numerical experiments we conducted, two examples for the parabolic model problem and one for the transport problem. Finally, in Section 7, we draw some conclusions and provide an outlook on future work.

2. Model time dependent PDEs used in our study

Throughout this paper, we consider the following model evolution problem: Find u such that

$$\begin{array}{c} \partial_t u + \mathrm{div}_x(\mathbf{L}_x(u)) = f, \text{ in } Q = \Omega \times (0,T) \\ u = u_0 \text{ on } \Sigma_0, \ \mathrm{tr}(u) = g \text{ on } \Sigma, \end{array} \tag{1}$$

where $\Sigma_0 = \Omega \times \{0\}$, $\Sigma = \partial\Omega \times (0,T)$, $\mathbf{L}_x(\cdot)$ is a spatial differential operator, and $\mathrm{tr}(\cdot)$ represents a trace operator corresponding to the boundary condition(s). We rewrite this problem using the space-time divergence $\mathrm{div}_{x,t}$ into

$$\begin{array}{c} \mathrm{div}_{x,t}(\mathbf{L}(u)) = f, \text{ in } Q \\ + \text{ B.C.}, \end{array} \tag{2}$$

with the space-time differential operator $\mathbf{L}(u) = [\mathbf{L}_x(u) \quad u]^\top$ and a (possibly combined) space-time boundary condition (B.C.). We will consider two applications of the model problem (1), the parabolic *heat equation*

$$\begin{aligned} \partial_t u - \mathrm{div}_x(\nabla_x u) &= f, \text{ in } Q, \\ u = u_0 \text{ on } \Sigma_0, \ u &= g \text{ on } \Sigma, \end{aligned} \tag{3}$$

i.e., $\mathbf{L}_x(u) = -\nabla_x u$, and the hyperbolic *transport equation*

$$\begin{aligned} \partial_t u + \mathrm{div}_x(\boldsymbol{\beta} u) &= f, \text{ in } Q, \\ u = u_0 \text{ on } \Sigma_0, \ \boldsymbol{\beta} u \cdot \vec{n}_x &= g \text{ on } \Sigma_{in}, \end{aligned} \tag{4}$$

i.e., $\mathbf{L}_x(u) = \boldsymbol{\beta}\, u$, and where $\Sigma_{in} = \{x \in \partial\Omega : \boldsymbol{\beta}(x) \cdot \vec{n}_x(x) < 0\} \times (0, T)$.

3. Space-time CFOSLS

In this section, we will describe the space-time least squares formulations, following the work of Voronin et al. [15]. In the following, we will make use of the function spaces: the space of square integrable functions $L_2(Q)$, $H^1(Q) = \{u \in L_2(Q) : \nabla_{x,t} u \in [L_2(Q)]^{d+1}\}$, $H(\mathrm{div}, Q) = \{\boldsymbol{\sigma} \in L_2(Q) : \mathrm{div}_{x,t}(\boldsymbol{\sigma}) \in L_2(Q)\}$. Moreover, we need the following spaces with boundary conditions

$$\begin{aligned} R &= \{\boldsymbol{\tau} \in H(\mathrm{div}, Q) : \mathrm{tr}_\sigma(\boldsymbol{\tau}) = 0 \text{ on } \Sigma^{\boldsymbol{\sigma}}_{ess}\}, \\ V &= \{v \in H^1(Q) : \mathrm{tr}_u(v) = 0 \text{ on } \Sigma^u_{ess}\}, \end{aligned}$$

where

$$\Sigma^{\boldsymbol{\sigma}}_{ess} = \emptyset \quad \text{and} \quad \Sigma^u_{ess} = \Sigma_0 \cup \Sigma$$

for the heat equation (3), and

$$\Sigma^{\boldsymbol{\sigma}}_{ess} = \Sigma_{in} \quad \text{and} \quad \Sigma^u_{ess} = \Sigma_0,$$

for the transport problem (4). Introducing a new variable $\boldsymbol{\sigma} = \mathbf{L}(u)$, we rewrite the model problem (2) as a first order system,

$$\begin{aligned} \boldsymbol{\sigma} - \mathbf{L}(u) &= 0, \\ \mathrm{div}_{x,t}(\boldsymbol{\sigma}) &= f. \end{aligned} \tag{5}$$

The corresponding FOSLS-functional is given by

$$\mathcal{J}(\boldsymbol{\sigma}, u) := \|\boldsymbol{\sigma} - \mathbf{L}(u)\|_0^2 + \|\mathrm{div}_{x,t}(\boldsymbol{\sigma}) - f\|_0^2,$$

that is well defined if $f \in L_2(Q)$. Then, the constrained first order system least squares (CFOSLS) problem is given by: find $(\boldsymbol{\sigma}, u) \in R \times V$ such that

$$\begin{aligned} (\boldsymbol{\sigma}, u) = &\operatorname*{arg\,min}_{(\boldsymbol{\tau}, v) \in R \times V} \mathcal{J}(\boldsymbol{\tau}, v), \\ \text{subject to (s.t.) } &\left\{ \ \mathrm{div}_{x,t} \boldsymbol{\sigma} = f \ . \right. \end{aligned} \tag{6}$$

Note that, as we have the divergence constraint, an option could be to omit the second term in the FOSLS functional $\mathcal{J}$. We keep though the second term of $\mathcal{J}$ for two reasons. First is that for the remainder of the paper we impose the constraints weakly. Secondly, we use the localized functionals as indicators for adaptive refinement since the information from the data f is beneficial in that case.

3.1. Variational CFOSLS formulation

We introduce the notation $\mathcal{A}(\boldsymbol{\sigma}, u) = \boldsymbol{\sigma} - \mathbf{L}(u)$. The first order, or Karush-Kuhn-Tucker (KKT), optimality conditions for the constrained optimization problem (6) are given by the saddle point problem: find $(\boldsymbol{\sigma}, u, \lambda) \in R \times V \times W$, $W = L_2(Q)$, such that

$$\begin{aligned} (\mathcal{A}(\boldsymbol{\sigma}, u), \mathcal{A}(\boldsymbol{\tau}, v)) + (\lambda, \mathrm{div}_{x,t}(\boldsymbol{\tau})) &= 0 \qquad && \forall (\boldsymbol{\tau}, v) \in R \times V \\ (\mathrm{div}_{x,t}\boldsymbol{\sigma}, \mu) &= (f, \mu) && \forall \mu \in W. \end{aligned} \tag{7}$$

We assume that the primal formulation (1) is well-posed, hence the above saddle point problem will possess at least one solution in appropriate function spaces for the solution u and $\boldsymbol{\sigma}$, as well as for the Lagrange multipliers λ. The finite element discretization problem will also have at least one solution (if we do not overconstrain it), since it is a nonnegative quadratic constrained minimization problem in a finite dimensional space. To ensure uniqueness of the solution of the discrete problems, we can add some regularization, a topic touched upon later in Section 4.2. We note that these important theoretical issues are not addressed any further in the present paper. We note that in (7), we look for a solution $u \in V \subset H^1(Q)$, which may be too restrictive. For instance in case of the transport equation, the operator $\mathbf{L}(u)$ does not include any derivatives, hence it is enough to look also for solutions $u \in W$. Moreover, we can eliminate one variable from the system. In order to do that, we first deduce that we can express u in terms of $\boldsymbol{\sigma}$ by a simple manipulation of the first equation in (5), i.e.,

$$u = \frac{1}{\mathbf{b}^\top \mathbf{b}} \mathbf{b}^\top \boldsymbol{\sigma},$$

with $\mathbf{b} = \begin{bmatrix} \boldsymbol{\beta} & 1 \end{bmatrix}^\top$. Inserting the above term into the FOSLS-functional $\mathcal{J}(\boldsymbol{\sigma}, u)$ results in a slightly modified CFOSLS problem: find $\boldsymbol{\sigma} \in R$ such that

$$\begin{aligned} \boldsymbol{\sigma} &= \arg\min_{\boldsymbol{\tau} \in R} \mathcal{J}(\boldsymbol{\tau}) \\ &\text{s.t. } \left\{ \mathrm{div}_{x,t}(\boldsymbol{\sigma}) = f \right. \end{aligned} \tag{8}$$

with $\mathcal{J}(\boldsymbol{\tau}) := \|K\boldsymbol{\tau}\|_0^2 + \|\mathrm{div}_{x,t}(\boldsymbol{\tau}) - f\|_0^2$ and

$$K = I - \frac{1}{\mathbf{b}^\top \mathbf{b}} \mathbf{b}\mathbf{b}^\top.$$

The resulting variational saddle point problem reads: find $(\boldsymbol{\sigma}, \lambda) \in R \times W$ such that

$$\begin{aligned} (K\boldsymbol{\sigma}, \boldsymbol{\tau}) + (\lambda, \mathrm{div}_{x,t}(\boldsymbol{\tau})) &= 0 \qquad && \forall \boldsymbol{\tau} \in R \\ (\mathrm{div}_{x,t}\boldsymbol{\sigma}, \mu) &= (f, \mu) && \forall \mu \in W. \end{aligned} \tag{9}$$

Remark. *As already remarked in [15], we have to be careful with the above matrix K. The matrix is rank-deficient in any point (x,t) (as $K \cdot \mathbf{b} \equiv 0$), but the corresponding global operator is non-singular on the kernel of the divergence operator. This is due to the assumption that the underlying differential problem is well-posed. However, the rank deficiency will definitely influence the condition number of the resulting linear system.*

4. Finite element discretization

In this section, we will discretize the saddle point systems from the previous section using the finite element method (FEM). We need a conforming subdivision $\mathcal{T}_h$ of the space-time cylinder Q into $(d+1)$-dimensional simplices, i.e.,

$$\overline{Q} = \bigcup_{\tau \in \mathcal{T}_h} \overline{\tau},$$

and we define the space-time element diameter $h_\tau = \operatorname{diam}(\tau)$, $\tau \in \mathcal{T}_h$, and $h = \max_{\tau \in \mathcal{T}_h} h_\tau$. Moreover, we need the following finite element function spaces: $V_h \subset V$ is the space of globally continuous and piecewise polynomial functions, $R_h \subset R$ is the space of $H(\mathrm{div})$-conforming functions, and $W_h \subset W$ is the space of piecewise polynomial functions. The space R_h is also referred to as a Raviart-Thomas (RT) space. Applying the usual (mixed) finite element discretization to the systems (7) and (9) results in the linear systems

$$\begin{bmatrix} \mathcal{M} & \mathcal{G}^\top & \mathcal{D}^\top \\ \mathcal{G} & \mathcal{X} & \\ \mathcal{D} & & \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma_h} \\ \boldsymbol{u_h} \\ \boldsymbol{\lambda_h} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \\ \mathbf{f} \end{bmatrix} \tag{10}$$

and

$$\begin{bmatrix} \mathcal{K} & \mathcal{D}^\top \\ \mathcal{D} & \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma_h} \\ \boldsymbol{\lambda_h} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f} \end{bmatrix} \tag{11}$$

respectively. Here $\boldsymbol{\sigma_h}, \boldsymbol{u_h}, \boldsymbol{\lambda_h}$ are the coefficient vectors associated with the finite element functions $\boldsymbol{\sigma}_h, u_h, \lambda_h$, respectively. The matrices $\mathcal{M}, \mathcal{G}, \mathcal{X}$, and $\mathcal{K}$ result from the discretization of the operators $\mathcal{A}(\boldsymbol{\sigma}, u)$ and $K\sigma$, respectively, and $\mathcal{D}$ is the discrete space-time divergence operator.

4.1. Negative Norm CFOSLS

Instead of minimizing the L_2-norm of the two equations in the first order system (5), we can define an alternative FOSLS-functional, while keeping the constraint, i.e.,

$$\begin{aligned} \min_{(\boldsymbol{\sigma},u)\in R\times V} \widetilde{\mathcal{J}}(\boldsymbol{\sigma},u) &:= \|\boldsymbol{\sigma} - \mathbf{L}(u)\|^2_{-s} + \|\mathrm{div}_{x,t}(\boldsymbol{\sigma}) - f\|^2_{-s}, \\ \text{s.t. } \{\ \mathrm{div}_{x,t}\boldsymbol{\sigma} &= f\ , \end{aligned} \tag{12}$$

where $\|.\|_{-s}$ denotes the norm of the negative index Sobolev space H^{-s}, $0 < s < 1$. The finite element counterpart of (12) reads similarly, i.e.,

$$\begin{aligned} &\min_{(\boldsymbol{\sigma}_h, u_h) \in R_h \times V_h} \widetilde{\mathcal{J}}(\boldsymbol{\sigma}_h, u_h), \\ &\text{s.t. } \left\{ \ \mathrm{div}_{x,t} \boldsymbol{\sigma}_h = f \ , \right. \end{aligned} \tag{13}$$

with $\widetilde{\mathcal{J}}$ as above. In order to obtain a computationally feasible problem, we introduce two additional constraints,

$$\mathbf{Q}_h(\boldsymbol{\sigma}_h - \mathbf{L}(u_h)) = 0, \tag{14a}$$

$$\mathbf{Q}_h(\mathrm{div}_{x,t}(\boldsymbol{\sigma}_h) - f) = 0, \tag{14b}$$

where $\mathbf{Q}_h$ is the L_2-projection on the space of piecewise constants w.r.t. the elements of the mesh $\mathcal{T}_h$, i.e.,

$$\mathbf{Q}_h(\boldsymbol{\sigma}_h - \mathbf{L}(u_h)) := \int_\tau (\boldsymbol{\sigma}_h - \mathbf{L}(u_h)) \ \mathrm{d}(x,t) \text{ for all } \tau \in \mathcal{T}_h.$$

Note that the second additional constraint (14b) is automatically satisfied by the already existing divergence constraint, hence we can omit it. We refer to (13)–(14a) as *negative norm CFOSLS*. We will now derive a computationally feasible formulation of (13)–(14a). Using (14a) and the well known standard error estimate, we obtain for the first term in the negative norm functional

$$\|\boldsymbol{\sigma}_h - \mathbf{L}(u_h)\|_{-s}^2 = \|(I - \mathbf{Q}_h)(\boldsymbol{\sigma}_h - \mathbf{L}(u_h))\|_{-s}^2 \leq C \sum_{\tau \in \mathcal{T}_h} h_\tau^{2s} \|\boldsymbol{\sigma}_h - \mathbf{L}(u_h)\|_{0,\tau}^2. \tag{15}$$

Furthermore, as we are dealing with finite element functions, we can apply the inverse inequality [5, Thm. 4.6] with $\alpha = 0$, and obtain

$$\|h^s(\boldsymbol{\sigma}_h - \mathbf{L}(u_h))\|_0 \leq C \|\boldsymbol{\sigma}_h - \mathbf{L}(u_h)\|_{-s}.$$

This estimate holds for $\mathbf{L}(u_h) = [-\nabla_x u_h, u_h]^T$, and it also holds for $\mathbf{L}(u_h) = [\boldsymbol{\beta} u_h, u_h]^T$ for $\boldsymbol{\beta}$ being a polynomial (as it is in our test Example 2). The derivations in what follows are used as a motivation to introduce the locally weighted least squares functional (18). Summing up over all $\tau \in \mathcal{T}_h$, we can deduce the bound

$$\sum_{\tau \in \mathcal{T}_h} h_\tau^{2s} \|\boldsymbol{\sigma} - \mathbf{L}(u)\|_{0,\tau}^2 = \|h^s(\boldsymbol{\sigma}_h - \mathbf{L}(u_h))\|_0^2 \leq C \|\boldsymbol{\sigma}_h - \mathbf{L}(u_h)\|_{-s}^2. \tag{16}$$

Deriving analogous bounds for the divergence constraint, and combining them with (15) and (16), we obtain an equivalent, but computationally feasible formulation of (13), i.e.

$$\begin{aligned} (\boldsymbol{\sigma}_h, u_h) = &\operatorname*{arg\,min}_{(\boldsymbol{\tau}_h, v_h) \in R_h \times V_h} \widetilde{\mathcal{J}}_h(\boldsymbol{\tau}_h, v_h) \\ &\text{s.t. } \left\{ \begin{array}{c} \mathrm{div}_{x,t} \boldsymbol{\sigma} = f \\ \mathbf{Q}_h(\boldsymbol{\sigma}_h - \mathbf{L}(u_h)) = 0 \end{array} \right. , \end{aligned} \tag{17}$$

with the locally scaled least squares functional

$$\widetilde{\mathcal{J}}_h(\boldsymbol{\tau}_h, v_h) := \sum_{\tau\in\mathcal{T}_h} h_\tau^{2s} \left(\|\boldsymbol{\tau}_h - \mathbf{L}(v_h)\|_{0,\tau}^2 + \|\mathrm{div}_{x,t}(\boldsymbol{\tau}_h) - f\|_{0,\tau}^2 \right). \tag{18}$$

The saddle point problems corresponding to the negative norm formulations of (6) and (9) read then

$$\begin{bmatrix} \mathcal{M}_h & \mathcal{G}_h^\top & \mathcal{D}_h^\top & \mathcal{E}_h^\top \\ \mathcal{G}_h & \mathcal{X}_h & & \mathcal{F}_h^\top \\ \mathcal{D}_h & & & \\ \mathcal{E}_h & \mathcal{F}_h & & \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma_h} \\ \boldsymbol{u_h} \\ \boldsymbol{\lambda_h} \\ \boldsymbol{\mu_h} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g}_h \\ \mathbf{f}_h \\ \mathbf{0} \end{bmatrix} \tag{19}$$

and

$$\begin{bmatrix} \mathcal{K}_h & \mathcal{D}_h^\top & \mathcal{E}_h^\top \\ \mathcal{D}_h & & \\ \mathcal{E}_h & & \end{bmatrix} \begin{bmatrix} \boldsymbol{\sigma_h} \\ \boldsymbol{\lambda_h} \\ \boldsymbol{\mu_h} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_h \\ \mathbf{0} \end{bmatrix} \tag{20}$$

respectively. Here, subscript h indicates that the local element matrices and vectors have been scaled by h_τ^{2s}, and $\mathcal{E}$, $\mathcal{F}$ originate from the discretization of the additional constraint (14a).

4.2. Alternative Formulation for the Transport Problem

We already mentioned in Section 3.1, that in general there is no guarantee that the discretized problem has a unique solution. However using regularization, we can indeed find a unique solution. In particular, we look for a solution with *minimal oscilations*, i.e., minimal L_2 norm of its gradient, which can also be interpreted as adding artificial viscosity to the least squares functional $\|\,\mathrm{div}_{x,t}(\mathbf{b}v) - f\|_0^2$, which leads to the following constrained least squares problem: find $u \in V$ such that

$$\begin{aligned} u &= \arg\min_{v\in V} \|\nabla_{x,t} v\|_0^2, \\ \text{s.t. } &\left\{ \ \mathrm{div}_{x,t}(\mathbf{b}\, u) = f \ , \right. \end{aligned} \tag{21}$$

with the constraint enforced weakly. Later, we will introduce some positive regularization parameter $\varepsilon > 0$. The choice of ε is not much of an issue, as we will show in a moment. We comment, that more sophisticated regularization approaches are of interest, such as for example, the L_1 minimization of the gradient of u, which however leads to a nonlinear method. We do not pursue the L_1–norm minimization in the present paper.

The weak first order optimality system corresponding to (21) reads

$$\begin{aligned} (\nabla_{x,t} u, \nabla_{x,t} v) + (\lambda, \mathrm{div}_{x,t}(\mathbf{b}v)) &= 0, && \text{for all } v \in V, \\ (\mathrm{div}_{x,t}(\mathbf{b}u), \mu) &= (f, \mu), && \text{for all } \mu \in W. \end{aligned}$$

Now, given an $\varepsilon > 0$, we let $\mu = \mathrm{div}_{x,t}(\mathbf{b}v)$ in the second equation, multiply it by $1/\varepsilon$ and add it to the first equation, which gives

$$\begin{aligned} (\nabla_{x,t} u, \nabla_{x,t} v) + \frac{1}{\varepsilon}(\mathrm{div}_{x,t}(\mathbf{b}u), \mathrm{div}_{x,t}(\mathbf{b}v)) + (\lambda, \mathrm{div}_{x,t}(\mathbf{b}v)) &= \frac{1}{\varepsilon}(f, \mathrm{div}_{x,t}(\mathbf{b}v)), \\ (\mathrm{div}_{x,t}(\mathbf{b}u), \mu) &= (f, \mu), \end{aligned}$$

for all $v \in V$ and all $\mu \in W$. Rescaling the Lagrange multiplier, i.e., $\lambda := \varepsilon\lambda$, we finally arrive at

$$\begin{aligned} \varepsilon(\nabla_{x,t}u, \nabla_{x,t}v) + (\mathrm{div}_{x,t}(\mathbf{b}u), \mathrm{div}_{x,t}(\mathbf{b}v)) + (\lambda, \mathrm{div}_{x,t}(\mathbf{b}v)) &= (f, \mathrm{div}_{x,t}(\mathbf{b}v)), \\ (\mathrm{div}_{x,t}(\mathbf{b}u), \mu) &= (f, \mu), \end{aligned}$$

for all $v \in V$ and all $\mu \in W$. The last problem corresponds to the constrained minimization problem: find $u \in V$ such that

$$\begin{aligned} u &= \arg\min_{v \in V} \widehat{\mathcal{J}}_\varepsilon(v), \\ \text{s.t. } &\left\{ \ \mathrm{div}_{x,t}(\mathbf{b}\, u) = f \ , \right. \end{aligned} \tag{22}$$

with the LS functional

$$\widehat{\mathcal{J}}_\varepsilon(v) := \varepsilon\|\nabla_{x,t}v\|_0^2 + \|\mathrm{div}_{x,t}(\mathbf{b}\, v) - f\|_0^2,$$

for a given positive regularization parameter ε. Again, it is clear that the choice of ε only affects the scaling of the Lagrange multiplier.

At any rate based on formulation (21), proceeding as before, we define the bilinear forms, derive the KKT system, and after discretization, we obtain the saddle point system of equations

$$\begin{bmatrix} \mathcal{L} & \mathcal{D}^\top \\ \mathcal{D} & \end{bmatrix} \begin{bmatrix} \boldsymbol{u_h} \\ \boldsymbol{\lambda_h} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{f} \end{bmatrix},$$

where $\mathcal{L}$ is a discrete Laplace operator, which we can locally scale by ε.

5. Linear Solvers for CFOSLS

In contrast to FOSLS-discretizations, which result in a system of linear equations with a positive definite system matrix, we have to deal with an indefinite system matrix in the case of CFOSLS. Hence, in general we cannot use the conjugated gradient method (CG), but we have to use some solver suitable for indefinite problems, e.g. the minimal residual method (MINRES) or the (flexible) generalized minimal residual method (GMRES). In order to speed up the convergence of the iterative solver, we will employ different preconditioning strategies, which we describe in the following subsections.

5.1. Block Diagonal Preconditioners

One straightforward method of preconditioning the linear systems (7) and (9) is to use the following block diagonal preconditioners

$$\begin{bmatrix} \mathcal{B}_{\mathcal{M}}^{-1} & & \\ & \mathcal{B}_{\mathcal{X}}^{-1} & \\ & & \mathcal{B}_{\mathcal{S}(\mathcal{M})}^{-1} \end{bmatrix}, \qquad \text{and,} \qquad \begin{bmatrix} \mathcal{B}_{\mathcal{K}}^{-1} & \\ & \mathcal{B}_{\mathcal{S}(\mathcal{K})}^{-1} \end{bmatrix},$$

respectively, where $\mathcal{S}(\,\cdot\,) = \mathcal{D}\mathrm{diag}(\,\cdot\,)^{-1}\mathcal{D}^\top$. The choice for the block preconditioners can be made w.r.t. the discretized problem, i.e., for the heat equation, we choose $\mathcal{B}_\mathcal{M}$ to be a Jacobi-smoother, and $\mathcal{B}_\mathcal{X}$ and $\mathcal{B}_{\mathcal{S}(\mathcal{M})}$ to be algebraic multigrid (AMG) preconditioners.
In case of negative norm CFOSLS, the saddle point system (19) has an additional block. We again formulate a Schur complement preconditioner for the last block, i.e.

$$\begin{bmatrix} \mathcal{B}_{\mathcal{M}_h} & & & \\ & \mathcal{B}_{\mathcal{X}_h} & & \\ & & \mathcal{B}_{\mathcal{S}_1} & \\ & & & \mathcal{B}_{\mathcal{S}_2} \end{bmatrix}$$

with $\mathcal{B}_{\mathcal{S}_1}$ the locally scaled equivalent of $\mathcal{B}_{\mathcal{S}(\mathcal{M})}$ and $\mathcal{S}_2 = \mathcal{E}\mathrm{diag}(\mathcal{M})^{-1}\mathcal{E}^\top + \mathcal{F}\mathrm{diag}(\mathcal{X})^{-1}\mathcal{F}^\top$. We can derive an analogous preconditioner for (20), given by

$$\begin{bmatrix} \mathcal{B}_{\mathcal{K}_h} & & \\ & \mathcal{B}_{\mathcal{S}_1} & \\ & & \mathcal{B}_{\mathcal{S}_2} \end{bmatrix}$$

with $\mathcal{B}_{\mathcal{S}_1}$ and $\mathcal{B}_{\mathcal{S}_2}$ defined correspondingly.

5.2. Monolithic Geometric Multigrid Preconditioners

Instead of treating the preconditioning of each block in (7) separately, we can apply a geometric multigrid method to the whole system. Let $\{P^{(k)}_{R_h}\}_{k=0}^l$, $\{P^{(k)}_{V_h}\}_{k=0}^l$ and $\{P^{(k)}_{W_h}\}_{k=0}^l$ be the sequences of prolongation operators between the hierarchy of corresponding finite element spaces. Then the prolongation operators for the block system (7) are given by

$$\mathcal{P}_{(k)} = \begin{bmatrix} P^{(k)}_{R_h} & & \\ & P^{(k)}_{V_h} & \\ & & P^{(k)}_{W_h} \end{bmatrix}.$$

Moreover, as we apply the solver in an adaptive scheme, we reuse the already assembled block system matrices from the coarser refinement levels in our multigrid hierarchy. We describe the application of one multigrid V-cycle in the following algorithm.

Algorithm 5.1. *Assume we have a hierarchy of matrices $\{\mathbf{A}_{(k)}\}_{k=0}^l$, prolongation operators $\{\mathcal{P}_{(k)}\}_{k=0}^l$ and smoothers $\{M_{(k)}\}_{k=0}^l$. At level k for given $\boldsymbol{v}$, the computation of $\mathcal{B}^{(k)}_{GMG}\boldsymbol{v}$ consists of:*

- *Pre-smooth, i.e., solve $M_{(k)}\boldsymbol{x} = \boldsymbol{v}$*
- *Restrict residual, i.e., compute $\boldsymbol{r}_c = \mathcal{P}^\top_{(k)}(\boldsymbol{v} - \mathbf{A}_{(k)}\boldsymbol{x})$*
- *Coarse grid correction, i.e.,*

  if $k+1 = l$, solve exactly, i.e, set $\boldsymbol{x}_c = \mathbf{A}^{-1}_{(l)}\boldsymbol{r}_c$

  else, apply the multigrid recursively, i.e, set $\boldsymbol{x}_c = \mathcal{B}^{(k+1)}_{GMG}\boldsymbol{r}_c$

- *Prolongate and update, i.e.,* $\boldsymbol{x} = \boldsymbol{x} + \mathcal{P}_{(k)}\boldsymbol{x}_c$,
- *Post-smooth, i.e., solve* $M_{(k)}(\boldsymbol{y} - \boldsymbol{x}) = \boldsymbol{v} - \mathbf{A}_{(k)}\boldsymbol{x}$.
- *This results in* $\mathcal{B}^{(k)}_{GMG}\boldsymbol{v} := \boldsymbol{y}$.

We still need to choose a smoother, or sequence of smoothers. A suitable choice is to use some least squares method, e.g. MINRES, preconditioned by the BDP-preconditioner from Section 5.1 as a smoother. We stop the iterative smoother either after reducing the relative residual by a factor $\gamma^{(k)}$, or after reaching a fixed, but small number of maximum iterations $n_M^{(k)}$. In Section 6 we solve the linear system (7) be means of flexible GMRES with one V-cycle of the multigrid described above, using $\gamma^{(k)} \equiv 0.01$ and $n_M^{(k)} \equiv 100$. We can derive analogous preconditioners for the systems (9), (19), and (20).

5.3. Divergence Free Solvers

An alternative way of solving systems (7) and (9) is to reformulate the problem already on the continuous level into a divergence-free setting. We will briefly summarize the method, for details see [15, Section 3]. The key idea is to split the function

$$\boldsymbol{\sigma} = \widetilde{\boldsymbol{\sigma}} + \overline{\boldsymbol{\sigma}} \tag{23}$$

into a function $\widetilde{\boldsymbol{\sigma}}$ satisfying the divergence constraint, i.e.,

$$(\mathrm{div}_{x,t}(\widetilde{\boldsymbol{\sigma}}), \mu) = (f, \mu) \quad \forall \mu \in W,$$

and a divergence free function $\overline{\boldsymbol{\sigma}}$ fulfilling $\mathrm{tr}_\sigma(\overline{\boldsymbol{\sigma}}) = 0$. Next, we recall the properties of the de Rham sequence, i.e., there exists an differential operator d and a Sobolev space N, such that for any $\boldsymbol{\psi} \in N$ with $\mathrm{tr}_\mathrm{d}(\boldsymbol{\psi}) = 0$, the image is in R, i.e., $\mathrm{d}\boldsymbol{\psi} \in R$ and, in addition, $\mathrm{div}_{x,t}(\mathrm{d}\boldsymbol{\psi}) \equiv 0$. For example, in 3D space-time, we use the canonical choice $\mathrm{d} = \mathrm{curl}$ and $N = H(\mathrm{curl}, Q)$, whereas for 4D space-time, we use $\mathrm{d} = \mathrm{Div}$ and $N = H(\mathrm{Div}, Q, \mathbb{K})$, with Div the row-wise matrix divergence, and

$$H(\mathrm{Div}, Q, \mathbb{K}) = \{\kappa \in L_2(\mathbb{K}) \ : \ \mathrm{Div}(\kappa) \in L_2(\mathbb{R}^4)\},$$

where $\mathbb{K}$ is the vector space of 4-by-4 skew-symmetric matrices; see [8] for details on this space. Now we can formulate the solving procedure in two steps:

Step 1 Find any $\widetilde{\boldsymbol{\sigma}} \in R$ satisfying

$$(\mathrm{div}_{x,t}(\widetilde{\boldsymbol{\sigma}}), \mu) = (f, \mu) \quad \forall \mu \in L_2(Q)$$

Step 2 Find the divergence free correction $\overline{\boldsymbol{\sigma}} = \mathrm{d}\boldsymbol{\psi}$. We use the decomposition (23) to rearrange system (7) into

$$(\mathcal{A}(\overline{\boldsymbol{\sigma}}, u), \mathcal{A}(\boldsymbol{\tau}, v)) + (\lambda, \mathrm{div}_{x,t}(\boldsymbol{\tau})) = -(\mathcal{A}(\widetilde{\boldsymbol{\sigma}}, u), \mathcal{A}(\boldsymbol{\tau}, v)) \quad \forall (\boldsymbol{\tau}, v) \in R \times V \tag{24}$$

$$(\mathrm{div}_{x,t}\overline{\boldsymbol{\sigma}}, \mu) = (f - \mathrm{div}_{x,t}\widetilde{\boldsymbol{\sigma}}, \mu) = 0 \quad \forall \mu \in W. \tag{25}$$

Now we insert $\widetilde{\boldsymbol{\sigma}}$ from **Step 1**, and replace $\overline{\boldsymbol{\sigma}} = \mathrm{d}\boldsymbol{\psi}$, which automatically fulfills the second equation. Moreover, it is sufficient to only test with $\boldsymbol{\tau} = \mathrm{d}\boldsymbol{\phi}$, $\boldsymbol{\phi} \in N$. Thus, we have to solve the following problem: Find $(\boldsymbol{\psi}, u) \in N \times V$ such that

$$(\mathcal{A}(\mathrm{d}\boldsymbol{\psi}, u), \mathcal{A}(\mathrm{d}\boldsymbol{\phi}, v)) = -(\mathcal{A}(\widetilde{\boldsymbol{\sigma}}, u), \mathcal{A}(\mathrm{d}\boldsymbol{\phi}, v)) \quad \forall (\boldsymbol{\phi}, v) \in N \times V.$$

Then the solution $\boldsymbol{\sigma}$ is obtained by summing up $\widetilde{\boldsymbol{\sigma}} + \overline{\boldsymbol{\sigma}}$.

We can again derive an analogous scheme for system (9).

6. Numerical Experiments

6.1. Implementation and Experimental Setup

We implemented the finite element methods described in Section 4 by means of the finite element library MFEM [10], which has also an implementation for 4D adaptive finite elements in a development branch[1]. In order to realize the solvers described in Section 5, we use the AMG implementation *boomerAMG* provided by the linear solver library hypre[2]. Both libraries are developed at LLNL. All of our numerical experiments are performed on the `Quartz` cluster, also located at LLNL.
We tested the following combinations of functionals, constraints and spaces in the numerical experiments:

- CFOSLS (6) for the heat equation (3)

$$\begin{aligned} &\min_{(\boldsymbol{\sigma}_h, u_h) \in R_h \times V_h} \mathcal{J}(\boldsymbol{\sigma}_h, u_h), \\ &\text{s.t. } \left\{ \ \mathrm{div}_{x,t} \boldsymbol{\sigma}_h = f \ , \right. \end{aligned} \tag{P1}$$

- CFOSLS (8) for the transport equation (4)

$$\begin{aligned} &\min_{\boldsymbol{\sigma}_h \in R_h} \mathcal{J}(\boldsymbol{\sigma}_h), \\ &\text{s.t. } \left\{ \ \mathrm{div}_{x,t} \boldsymbol{\sigma}_h = f \ . \right. \end{aligned} \tag{P2}$$

- Negative Norm CFOSLS (13) for the transport equation (4)

$$\begin{aligned} &\min_{\boldsymbol{\sigma}_h \in R_h} \widetilde{\mathcal{J}}_h(\boldsymbol{\sigma}_h), \\ &\text{s.t. } \left\{ \begin{array}{l} \mathrm{div}_{x,t} \boldsymbol{\sigma}_h = f, \\ \mathbf{Q}_h(K\boldsymbol{\sigma}_h) = 0. \end{array} \right. \end{aligned} \tag{$\widetilde{\text{P2}}$}$$

[1] `https://github.com/mfem/mfem/tree/4d_dev`
[2] `http://www.llnl.gov/casc/hypre/`

- Regularized constrained LS functional (22) for the transport equation (4)

$$\begin{aligned} &\min_{u_h \in V_h} \widehat{\mathcal{J}}_\varepsilon(u_h), \\ &\text{s.t. } \left\{ \ \operatorname{div}_{x,t}(\mathbf{b}\, u_h) = f, \right. \end{aligned} \tag{P3}$$

In all numerical experiments we use the localized least squares functional $\mathcal{J}_\tau$ as an indicator for refinement, e.g., for (P1) we define $\mathcal{J}_\tau(\boldsymbol{\sigma}_h, u_h) = \mathcal{J}(\boldsymbol{\sigma}_h, u_h)|_\tau$, $\tau \in \mathcal{T}_h$. The elements then are marked using Dörfler's marking strategy [7], i.e., we mark all elements $\tau \in \widetilde{\mathcal{T}}_h \subseteq \mathcal{T}_h$, where $\widetilde{\mathcal{T}}_h$ is the set with minimal cardinality, which, for given $\theta \in (0,1)$, fulfills

$$\theta \mathcal{J}(\boldsymbol{\sigma}_h, u_h) \leq \sum_{\tau \in \widetilde{\mathcal{T}}_h} \mathcal{J}_\tau(\boldsymbol{\sigma}_h, u_h).$$

In particular, we implemented the algorithm recently proposed by Pfeiler et al. [12], which in addition is very easy to parallelize. Once we have a set of marked elements, we apply the bisection algorithm by Arnold et al. [3] in 3D space-time, and the bisection algorithm by Stevenson [14] in 4D space-time, to subdivide the marked elements. Both algorithms ensure that the family of adaptive refined meshes $(\mathcal{T}_h)_h$ stays conforming, i.e. we do not introduce any hanging nodes, and that the mesh elements do not degenerate, i.e. there is a lower bound for the interior angles for all h. Unless stated otherwise, we will always use a bulk parameter $\theta = 0.25$.
We also compared the convergence rates for different polynomial degree of the finite element ansatz functions. With $p = 0$ we denote the lowest order finite elements, i.e., piecewise constants for L_2-conforming elements, and piecewise linear, globally continuous functions for H^1-conforming elements. The use of next-to-lowest-order elements is denoted by $p = 1$.
When we refer to the solvers and preconditioners presented in Section 5, we will use the following abbreviations: **BDP** for MINRES preconditioned by the block diagonal preconditioner described in Section 5.1, **MGMG** for flexible GMRES preconditioned by the monolithic geometric multigrid introduced in Section 5.2, and **MG** for the divergence free solver defined in Section 5.3.

6.2. Example 1: Parabolic Evolution Problem

We consider the computational domain $Q = (0,1)^{d+1}$, $d = 2, 3$, and use the manufactured solution

$$u(x,t) = t^2 e^t \prod_{i=1}^{d} \sin\big((4-i)x_i\pi\big).$$

We consider only the heat equation (3), thus we use formulation (P1). The manufactured solution is very smooth and has no distinct features, so adaptive refinement will improve the overall convergence rate only by a constant factor. In Table 1, we present the total number of iterations needed to reduce the initial

residual by a factor of 10^{-6}, with the elapsed time in parentheses, using 8 nodes (or 288 cores) on `Quartz`. While we can observe stable iteration counts for the monolithic geometric multigrid, the cost per iteration is significantly higher than for the other solvers.

Table 1: Example 1: Iteration numbers and solving times.

(a) $p = 0$, $d = 2$

#dofs	BDP		MGMG		MG	
80 177	105	(0.45 s)	5	(1.33 s)	32	(0.53 s)
631 905	145	(0.83 s)	6	(2.31 s)	46	(0.79 s)
5 017 793	193	(2.22 s)	7	(5.33 s)	65	(1.53 s)
39 993 729	262	(12.49 s)	8	(24.68 s)	92	(6.97 s)
319 357 697	366	(145.67 s)	8	(241.61 s)	135	(74.50 s)

(b) $p = 0$, $d = 3$

#dofs	BDP		MGMG		MG	
169 281	229	(0.64 s)	3	(1.49 s)	28	(2.22 s)
332 481	256	(0.71 s)	3	(3.15 s)	23	(1.99 s)
668 097	234	(0.99 s)	3	(5.00 s)	31	(2.96 s)
1 336 881	274	(1.59 s)	3	(8.13 s)	33	(3.64 s)
2 656 865	298	(2.15 s)	3	(12.30 s)	34	(4.93 s)
5 268 065	312	(3.23 s)	4	(24.66 s)	30	(5.98 s)
10 594 913	293	(4.71 s)	3	(28.19 s)	32	(10.13 s)
21 195 809	339	(9.86 s)	4	(60.93 s)	34	(19.16 s)
42 133 569	376	(20.46 s)	4	(101.56 s)	35	(38.19 s)
83 912 769	410	(45.29 s)	5	(231.08 s)	31	(69.64 s)
168 798 273	437	(99.63 s)	5	(467.87 s)	-	(-)
337 645 377	490	(237.41 s)	-	(-)	-	(-)

For the sake of completeness, we also include plots of the convergence rates; see Fig. 1 and Fig. 2. As expected, for problems without singularities or without localized (anisotropic) features, adaptive refinement improves the convergence rates only by a constant factor.

6.3. Example 2: The Transport equation

For our second example, we will consider only the transport equation (4). Here, the space-time cylinder is given by $Q = \{x \in \mathbb{R}^d : \|x\|_{\ell_2} < 1\} \times (0, 2)$. We choose the vector fields

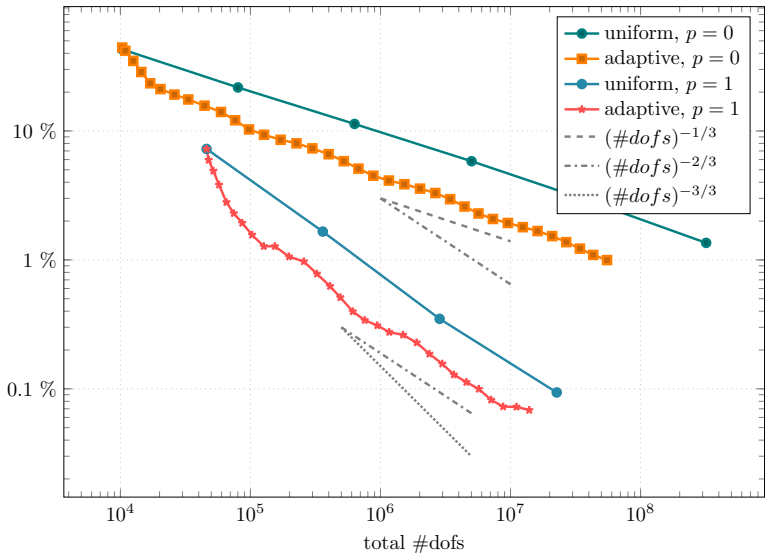
$$\boldsymbol{\beta} = 2 \begin{bmatrix} -x_2 \\ x_1 \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\beta} = (1 - x_3^2) \begin{bmatrix} -x_2 \\ x_1 \\ 0 \end{bmatrix},$$

and the initial data

$$u_0(x) = e^{-100\left((x_1-0.5)^2+x_2^2\right)}, \quad \text{and} \quad u_0(x) = e^{-100\left((x_1-0.5)^2+x_2^2+x_3^2\right)},$$

for $d = 2$ and $d = 3$, respectively. We prescribe no source terms, i.e., $f \equiv 0$. Hence, we expect that the profile of the initial data should only be transported along the vector field $\boldsymbol{\beta}$. Moreover, as we consider a hyperbolic evolution equation, there should be no dissipation. This solution is very localized and has sharp gradients, thus we expect a significant improvement in the convergence rates by using adaptive refinement. We will compare formulations (P2)–(P3), as well as different values of the index s for formulation ($\widetilde{\text{P2}}$).
First we consider the case $d = 2$. Here, the spatial domain is the unit circle and the flow field is a counter-clockwise, circular motion. In Fig. 3a, we plot the relative L_2-error in the variable u. We can observe that, in comparison with uniform refinement, more than one order of magnitude less #dofs is needed

(a) Relative error $\|u - u_h\|/\|u\|$.



(b) Relative error $\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|/\|\boldsymbol{\sigma}\|$.

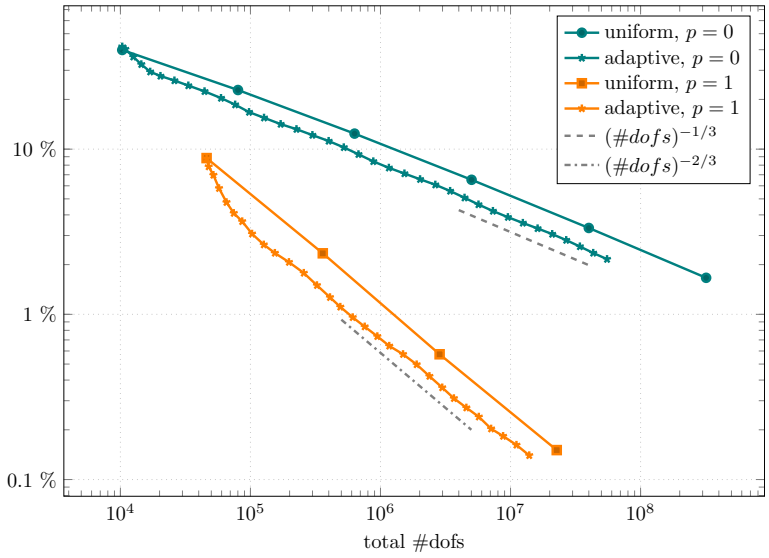
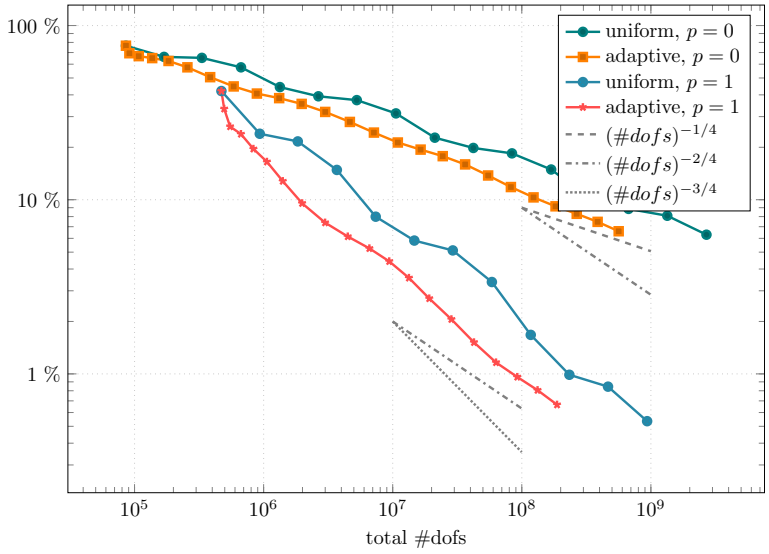


Figure 1: Example 1: Convergence rates of uniform and adaptive refinements, for $d = 2$.

(a) Relative error $\|u - u_h\|/\|u\|$.



(b) Relative error $\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|/\|\boldsymbol{\sigma}\|$.

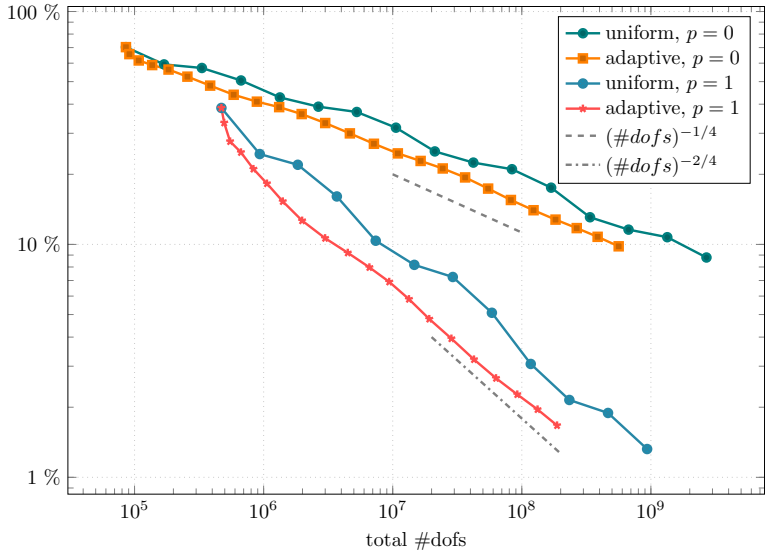


Figure 2: Example 1: Convergence rates of uniform and adaptive refinements, for $d = 3$.

to reach a certain threshold with adaptive refinement. Moreover, comparing formulations (P2) and ($\widetilde{\text{P2}}$), we deduce that the additional constraint does not really improve the convergence behavior for this example. In Fig. 4, we present cuts through the space-time mesh at certain times t. The mesh was obtained after 17 adaptive refinements. We can clearly see that the mesh is very fine along the movement of the peak, whereas stays relatively coarse where nothing happens. Next we consider the case $d = 3$, where the spatial domain is the unit ball, and the flow field is again a circular motion in the x_1-x_2-plane, with a varying intensity depending on the x_3 variable. In Fig. 3b, we also plot the relative L_2-error in u. Again, the adaptive refinement results in a huge improvement in the convergence rates. However, here, formulations (P2) and ($\widetilde{\text{P2}}$) differ more than previously. The adaptive refinement for negative norm CFOSLS seems to result in worse convergence behavior than adaptivity for standard CFOSLS. In Fig. 5, we present the finite element solution obtained after 13 adaptive refinements. The 4D hyper-cylinder Q was cut at times t, and the finite element solution was then projected on these 3D cuts.

6.4. Example 3: Adaptive Benchmark

For the third example we consider the NIST adaptive benchmark problem[3] "Moving Circular Wave Front". The space-time cylinder is given by $Q = (0, 10) \times (-5, 5) \times (0, 10)$, i.e., $d = 2$. We again use a manufactured solution

$$u(x,t) = \frac{B(x)\arctan(t)\left(\frac{\pi}{2} - \arctan\left(\alpha\left(\|x - x_c\|_{\ell_2}\right)\right)\right)}{C},$$

with a bubble function

$$B(x) = x_1(x_1 - 10)(x_2 + 5)(x_2 - 5)$$

and where x_c is the origin of the wavefront, α controls the steepness of the wave, and C is a constant to scale the function value. For this example, we choose $\alpha = 20$, $C = 10000$, and $x_c = \mathbf{0}$. The original benchmark is for a parabolic model problem, i.e., the heat equation (3). However, we can use the manufactured solution u also for the transport problem (4), with the flow field

$$\boldsymbol{\beta} = \begin{bmatrix} \sin(\frac{\pi}{5}x_1)\cos(\frac{\pi}{5}(x_2+5)) \\ -\cos(\frac{\pi}{5}x_1)\sin(\frac{\pi}{5}(x_2+5)) \end{bmatrix}.$$

We again compare the convergence rates in the L_2-norm of u_h and $\boldsymbol{\sigma}_h$ for the different formulations (P1)–(P3), as well as different values of s. For lowest order elements, i.e. $p = 0$, we observe an improvement in the convergence rates for both, the parabolic and the hyperbolic problem. In particular, for the heat equation, we need more than one order of magnitude less dofs to reach a certain

[3] https://math.nist.gov/amr-benchmark/index.html

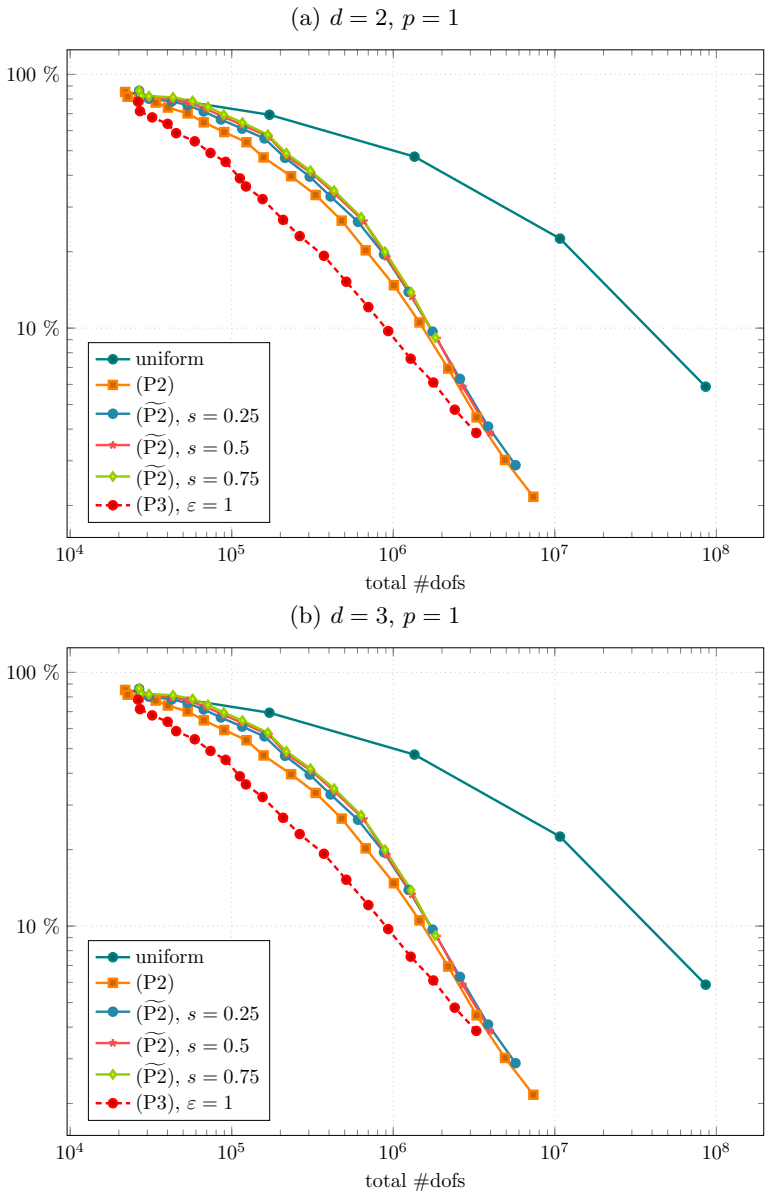


Figure 3: Example 2: Error rates in $\|u - u_h\|_0/\|u\|_0$ for uniform and adaptive refinements.

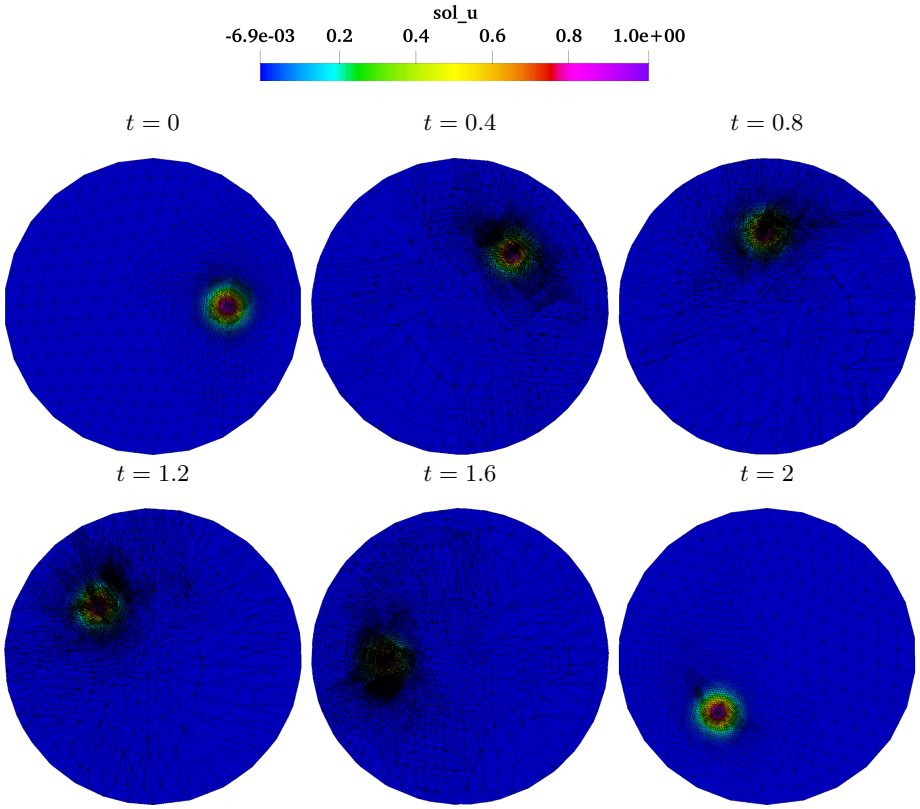


Figure 4: Example 2: Finite element solution u_h plotted over the space-time mesh after 17 adaptive refinements, for $d = 2$, cut at times t.

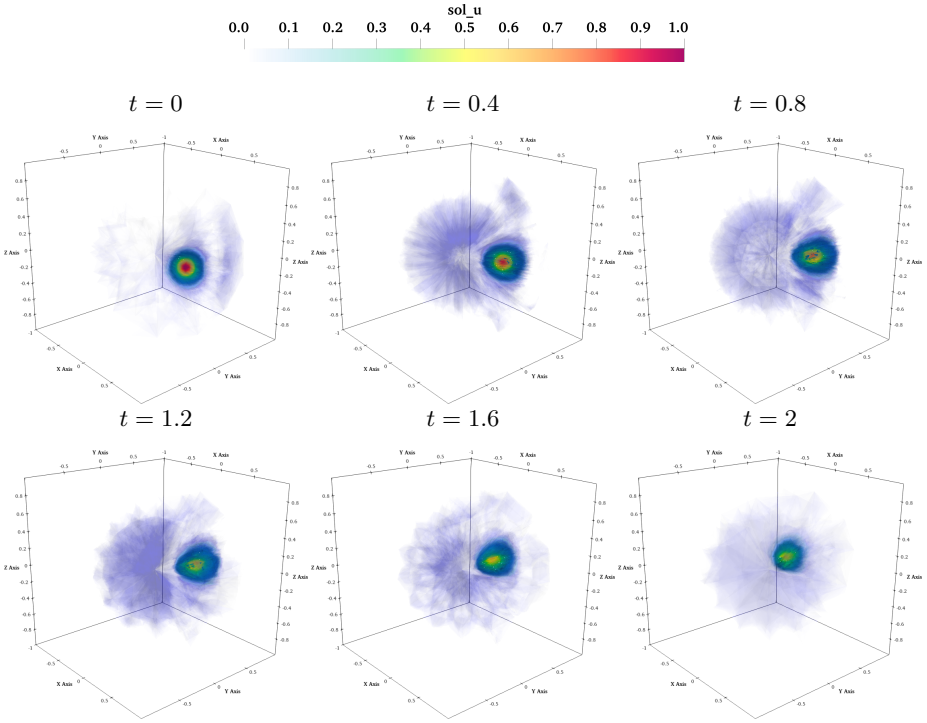


Figure 5: Example 2: Finite element solution u_h plotted over the space-time mesh after 13 adaptive refinements, for $d = 3$, cut at times t.

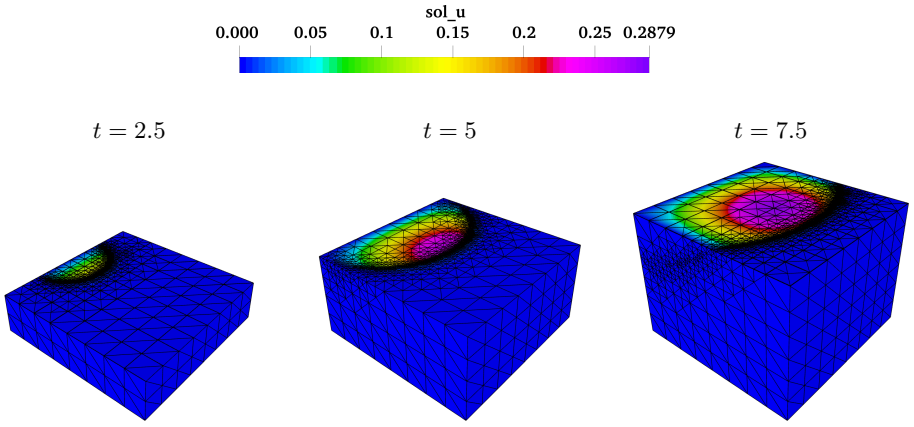


Figure 6: Example 3: Finite element solution u_h obtained by formulation (P1) with $p = 1$, plotted over the whole space-time mesh after 15 adaptive refinements with bulk parameter $\theta = 0.1$, cut off at t.

threshold compared to uniform refinement; see Fig. 7b. For the hyperbolic problem, we can observe a similar behavior, where we also save roughly one order of magnitude of dofs when comparing adaptive and uniform refinement.

For next-to-lowest-order elements, i.e. $p = 1$, we now in addition compare different values of the negative norm index s. The parabolic problem again shows a sharp improvement in the convergence rates for adaptive refinement, e.g., in order to reach an relative L_2-error of 10% for $\boldsymbol{\sigma}$, we need around 5×10^6 dofs for adaptive refinement, compared to $>10^8$ dofs needed in the case of uniform refinement; see Fig. 8b. In Fig. 6, we present plots of finite element solution u_h to formulation (P1) over the whole space-time mesh $\mathcal{T}_h$, where cut off parts of the space-time cylinder to visualize the spatial profile of the solution for non-zero times t are shown. The hyperbolic transport problem behaves a little different from in the lowest order case. In order to see an improvement in the convergence rates, we first need to reach certain amount of dofs. Then we again need around one order of magnitude less #dofs to reach a certain threshold in the errors. When comparing the standard CFOSLS to negative norm CFOSLS, we can further reduce the needed number of degrees of freedom by a factor of 2, depending on the choice of s; see Fig. 8.

7. Conclusion

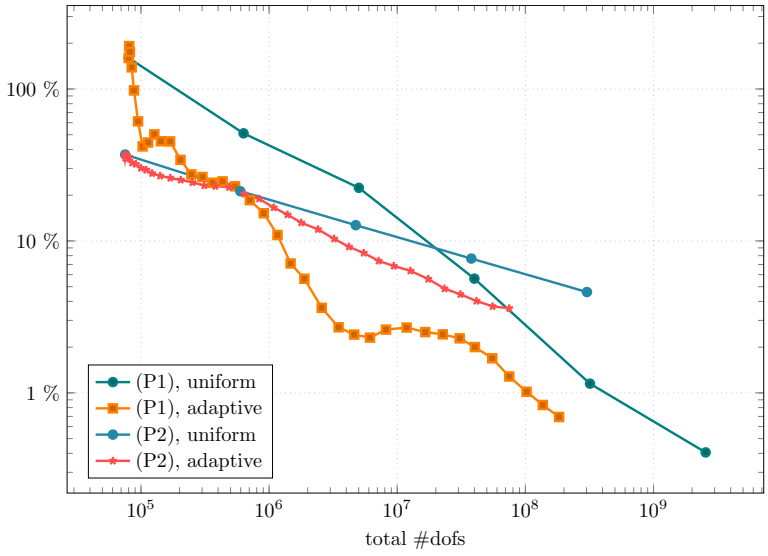
The overall conclusion of the presented computational study is that the space-time FOSLS technique poses in addition to the main constraint of high memory demand the challenge of finding scalable solvers which is due to the lack of full ellipticity of the FOSLS functionals. As we have demonstrated, that with the help of AMR, the severe memory constraints can be alleviated and also with the AMR we can benefit in terms of generating meshes (not necessarily exploiting time-stepping) that follow the physics and not posing a global CFL constraint on the discretization. Future research is needed in the direction of designing more scalable solvers and one possible venue for that is to exploit recent advances of adaptive AMG methods that can take advantage of specialized coarsening to detect possible anisotropy [6, 13] and not necessarily require ellipticity of the bilinear forms. Moreover, in order to take full advantage of the parallel AMR, we need to introduce some kind of load balancing to ensure that the work is equally distributed among the cores.

Acknowledgments

We would like to thank the anonymous referees for their helpful remarks.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. One author was supported by the Austrian Science Fund (FWF) under the grant DK W1214-04.
LLNL Release Number: LLNL-JRNL-806077

(a) $\|u - u_h\|_0 / \|u\|_0$



(b) $\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_0 / \|\boldsymbol{\sigma}\|_0$

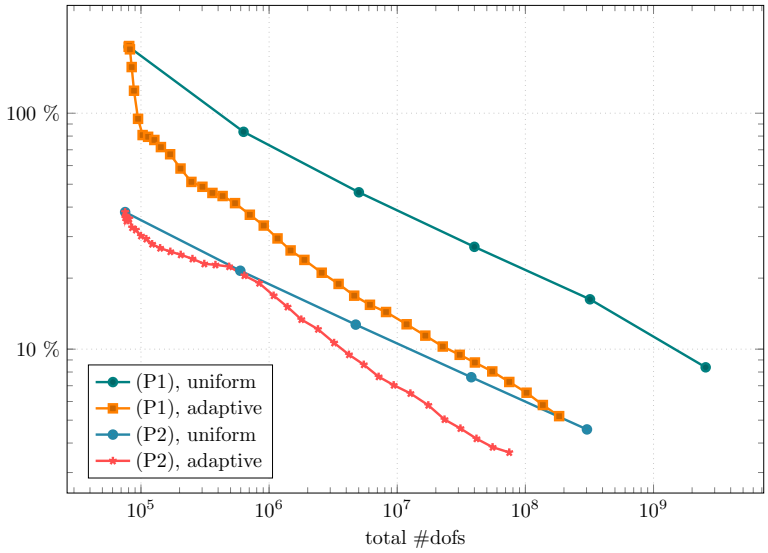
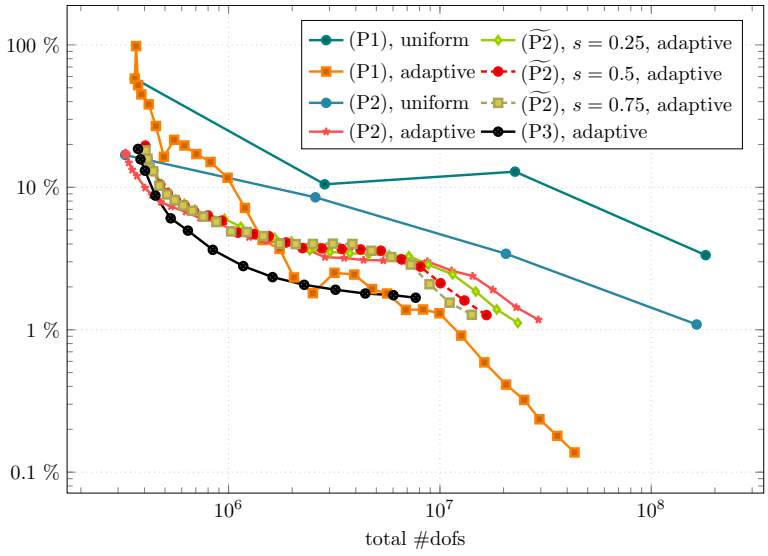


Figure 7: Example 3: Convergence rates for $d = 2$, $p = 0$.

(a) $\|u - u_h\|_0 / \|u\|_0$



(b) $\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_0 / \|\boldsymbol{\sigma}\|_0$

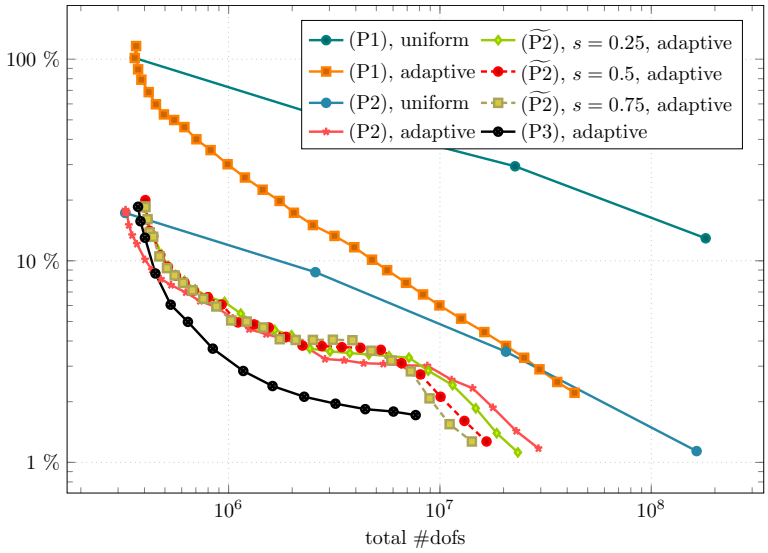


Figure 8: Example 3: Convergence rates for $d = 2$, $p = 1$.

References

[1] Adler, J. H., Manteuffel, T. A., McCormick, S. F., Nolting, J. W., Ruge, J. W., and Tang, L. Efficiency based adaptive local refinement for first-order system least-squares formulations. *SIAM J. Sci. Comput. 33*, 1 (2011), 1–24.

[2] Adler, J. H., and Vassilevski, P. S. Error analysis for constrained first-order system least-squares finite-element methods. *SIAM J. Sci. Comput. 36*, 3 (2014), a1071–a1088.

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

[4] Berndt, M., Manteuffel, T. A., and McCormick, S. F. Local error estimates and adaptive refinement for first-order system least squares (FOSLS). *ETNA, Electron. Trans. Numer. Anal. 6* (1997), 35–43.

[5] Dahmen, W., Faermann, B., Graham, I. G., Hackbusch, W., and Sauter, S. A. Inverse inequalities on non-quasi-uniform meshes and application to the mortar element method. *Math. Comput. 73*, 247 (2004), 1107–1138.

[6] D'Ambra, P., and Vassilevski, P. S. Adaptive AMG with coarsening based on compatible weighted matching. *Comput. Vis. Sci. 16*, 2 (2013), 59–76.

[7] Dörfler, W. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal. 33*, 3 (1996), 1106–1124.

[8] Gopalakrishnan, J., Neumüller, M., and Vassilevski, P. S. The auxiliary space preconditioner for the de Rham complex. *SIAM J. Numer. Anal. 56*, 6 (2018), 3196–3218.

[9] Manteuffel, T., McCormick, S., Nolting, J., Ruge, J., and Sanders, G. Further results on error estimators for local refinement with first-order system least squares (FOSLS). *Numer. Linear Algebra Appl. 17*, 2-3 (2010), 387–413.

[10] MFEM: Modular finite element methods library. mfem.org.

[11] Neumüller, M., Vassilevski, P. S., and Villa, U. E. Space-time CFOSLS methods with AMGe upscaling. In *Domain decomposition methods in science and engineering XXIII. Proceedings of the 23rd international conference, Jeju Island, South Korea, July 6–10, 2015*. Cham: Springer, 2017, pp. 253–260.

[12] Pfeiler, C.-M., and Praetorius, D. Drfler marking with minimal cardinality is a linear complexity problem, 2019.

[13] Quiring, B. Q., and Vassilevski, P. S. Properties of the graph modularity matrix and its applicaions. Tech. Rep. LLNL-TR-779424, Lawrence Livermore National Laboratory, 2019.

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

[15] Voronin, K., Lee, C. S., Neumüller, M., Sepulveda, P., and Vassilevski, P. S. Space-time discretizations using constrained first-order system least squares (CFOSLS). *J. Comput. Phys. 373* (2018), 863–876.