JOHANNES KEPLER UNIVERSITY LINZ

Institute of Computational Mathematics

A Multigrid Fourier Analysis of a Multigrid Method using Symbolic Computation

Veronika Pillwein
Research Institute for Symbolic Computation
Johannes Kepler University
Schloss Hagenberg (Castle of Hagenberg)
Kirchenplatz 5b, 4232 Hagenberg im Mühlkreis, Austria

Stefan Takacs
Doctoral Program "Computational Mathematics" (W1214)
Johannes Kepler University
Altenberger Str. 69, 4040 Linz, Austria

NuMa-Report No. 2012-05 April 2012

Technical Reports before 1998:

1995

95-1	Hedwig Brandstetter *Was ist neu in Fortran 90?*	March 1995
95-2	G. Haase, B. Heise, M. Kuhn, U. Langer *Adaptive Domain Decomposition Methods for Finite and Boundary Element Equations.*	August 1995
95-3	Joachim Schöberl *An Automatic Mesh Generator Using Geometric Rules for Two and Three Space Dimensions.*	August 1995

1996

96-1	Ferdinand Kickinger *Automatic Mesh Generation for 3D Objects.*	February 1996
96-2	Mario Goppold, Gundolf Haase, Bodo Heise und Michael Kuhn *Preprocessing in BE/FE Domain Decomposition Methods.*	February 1996
96-3	Bodo Heise *A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation.*	February 1996
96-4	Bodo Heise und Michael Jung *Robust Parallel Newton-Multilevel Methods.*	February 1996
96-5	Ferdinand Kickinger *Algebraic Multigrid for Discrete Elliptic Second Order Problems.*	February 1996
96-6	Bodo Heise *A Mixed Variational Formulation for 3D Magnetostatics and its Finite Element Discretisation.*	May 1996
96-7	Michael Kuhn *Benchmarking for Boundary Element Methods.*	June 1996

1997

97-1	Bodo Heise, Michael Kuhn and Ulrich Langer *A Mixed Variational Formulation for 3D Magnetostatics in the Space $H(rot) \cap H(div)$*	February 1997
97-2	Joachim Schöberl *Robust Multigrid Preconditioning for Parameter Dependent Problems I: The Stokes-type Case.*	June 1997
97-3	Ferdinand Kickinger, Sergei V. Nepomnyaschikh, Ralf Pfau, Joachim Schöberl *Numerical Estimates of Inequalities in $H^{\frac{1}{2}}$.*	August 1997
97-4	Joachim Schöberl *Programmbeschreibung NAOMI 2D und Algebraic Multigrid.*	September 1997

From 1998 to 2008 technical reports were published by SFB013. Please see
`http://www.sfb013.uni-linz.ac.at/index.php?id=reports`
From 2004 on reports were also published by RICAM. Please see
`http://www.ricam.oeaw.ac.at/publications/list/`
For a complete list of NuMa reports see
`http://www.numa.uni-linz.ac.at/Publications/List/`

A LOCAL FOURIER CONVERGENCE ANALYSIS OF A MULTIGRID METHOD USING SYMBOLIC COMPUTATION

VERONIKA PILLWEIN AND STEFAN TAKACS

Abstract. For iterative solvers, besides convergence proofs that may state qualitative results for some classes of problems, straight-forward methods to compute (bounds for) convergence rates are of particular interest. A widely-used straight-forward method to analyze the convergence of numerical methods for solving discretized systems of partial differential equations (PDEs) is local Fourier analysis (or local mode analysis). The rates that can be computed with local Fourier analysis are typically the supremum of some rational function. In the past this supremum was merely approximated numerically by interpolation. We are interested in resolving the supremum exactly using a standard tool from symbolic computation: cylindrical algebraic decomposition (CAD). In this paper we work out the details of this symbolic local Fourier analysis for a multigrid solver applied to a PDE-constrained optimization problem.

1. Introduction

As mentioned in the abstract, in this work we introduce symbolic local Fourier analysis (sLFA) for analyzing the convergence behavior of a numerical method. Local Fourier analysis (or local mode analysis) is a commonly used approach for designing multigrid methods and analyzing their convergence properties. It dates back to the late 1970s when A. Brandt [2] proposed to use Fourier series in the analysis of multigrid methods. Local Fourier analysis provides a framework to analyze various numerical methods with a unified approach that gives quantitative statements on the methods under investigation, i.e., it leads to the determination of sharp convergence rates. Typically in multigrid theory the fact of convergence is shown and neither sharp nor realistic bounds for convergence rates are provided, cf. standard analysis [7].

Local Fourier analysis can be justified rigorously only in special cases, e.g., on rectangular domains with uniform grids and periodic boundary conditions. However, results obtained with local Fourier analysis can be carried over rigorously to more general classes of problems, cf. [3]. Moreover, it can be viewed as heuristic approach for a wide class of applications.

We understand local Fourier analysis as a machinery, which we apply in this paper to a model problem and a particular multigrid solver. A similar viewpoint was taken in [19], where also a local Fourier analysis software was presented that can be configured using a graphical user interface and allows ot approximate (numerically) smoothing and convergence rates based on local Fourier analysis approaches for various problems and different multigrid-methods.

1991 *Mathematics Subject Classification.* Primary 68W30, 65N12, 26D05, 65N55.

The research was funded by the Austrian Science Fund (FWF): W1214-N15, projects DK6 and DK12, and grant P22748-N18.

As a model problem for the proposed machinery, we choose the application of a multigrid method to a particular PDE-constrained optimization problem. This problem is characterized by a parameter-dependent linear system which is not positive definite. Therefore the construction of a parameter-robust numerical method for such a problem is non-standard. More or less the same problem an the same smoother have been analyzed in [1] using classical local Fourier analysis, i.e., there the smoothing rate was approximated numerically. Here we present a symbolic analysis providing sharp, exact upper bounds. Still, we keep in mind that this analysis can be carried over to a variety of other problems and other (multigrid) solvers.

In [12] we used the proposed method for a one dimensional problem and proving smoothing properties only, which was a much easier task to handle. The goal of the present paper is to demonstrate that the analysis can be carried out in an *entirely symbolic way* and as such leads to *sharp estimates* for the convergence rates also for two dimensional problems. Aiming at an audience from both numerical and symbolic mathematics we try to stay at an elementary level and keep this note self-contained.

For deriving the explicit representation of the convergence rates, we use Cylindrical Algebraic Decomposition (CAD), a well established method in symbolic computation. It was introduced for solving the problem of quantifier elimination in the theory of real numbers. Below we see that the bound for the convergence rate is defined as the supremum over a certain rational function (after an appropriate rewriting). This allows us to invoke CAD in the solution of the problem. CAD has been applied earlier in the analysis of (systems of) ordinary and partial differential-difference equations [8], where the necessary conditions for stability, asymptotic stability and well-posedness of the given systems were transformed into statements on polynomial inequalities using Fourier or Laplace transforms.

This paper is organized as follows. In Section 2 the numerical method analyzed in this paper using symbolic local Fourier analysis is introduced. We recall the method of classical local Fourier analysis and present the setup for the model problem in Section 3. In Sections 4 and 5 the symbolic local Fourier analysis is carried out to give sharp bounds for the convergence rates in the one and two dimensional case, respectively.

2. Model problem

The convergence analysis presented in this paper should be understood as a machinery that can be applied to various problems and solvers. The analysis is carried out for the optimization problem constrained to a partial differential equation (PDE-constrained optimization problem), introduced in Subsection 2.1, and a multigrid solver, introduced in Subsection 2.2.

2.1. A PDE-constrained optimization problem.

The analysis is carried out for the following model problem, which is an optimal control problem of tracking type.

Problem 2.1. Find state y and control u such that they minimize

$$J(y,u) := \frac{1}{2}\int_\Omega (y(\mathbf{x}) - y_D(\mathbf{x}))^2 \,\mathrm{d}\mathbf{x} + \frac{\alpha}{2}\int_\Omega u^2(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

subject to the elliptic boundary value problem (BVP)

$$-\Delta y = u \text{ in } \Omega \qquad \text{and} \qquad y = 0 \text{ on } \partial\Omega.$$

Here, for $d = 1, 2, 3$ the set $\Omega \subset \mathbb{R}^d$ is a given domain with sufficiently smooth boundary $\partial\Omega$ and Δ is the standard Laplace operator, i.e., $\Delta := (\frac{\partial^2}{\partial \mathbf{x}_1^2} + \cdots + \frac{\partial^2}{\partial \mathbf{x}_d^2})$. Moreover, the desired state y_D and the regularization or cost parameter $\alpha > 0$ are assumed to be given.

The solution of the optimal control problem is characterized by its optimality system (Karush Kuhn Tucker system or KKT-system), which consists of state y, control u and and Lagrange multiplier, say p. The relation $u = \alpha^{-1}p$ follows directly from the KKT-system, which allows to eliminate u. This leads to the reduced KKT-system, which reads as follows.

Problem 2.2. Let $y_D \in L^2(\Omega)$ and $\alpha > 0$ be given. Find $(y, p) \in X := V \times V := H_0^1(\Omega) \times H_0^1(\Omega)$ such that

$$\begin{array}{ccccc} (y, \widetilde{y})_{L^2(\Omega)} & + & (\nabla p, \nabla \widetilde{y})_{L^2(\Omega)} & = & (y_D, \widetilde{y})_{L^2(\Omega)} \\ (\nabla y, \nabla \widetilde{p})_{L^2(\Omega)} & - & \alpha^{-1}(p, \widetilde{p})_{L^2(\Omega)} & = & 0 \end{array}$$

holds for all $(\widetilde{y}, \widetilde{p}) \in X$.

Here, $\nabla := (\frac{\partial}{\partial \mathbf{x}_1}, \ldots, \frac{\partial}{\partial \mathbf{x}_d})$ is the (weak) gradient and $(u, v)_{L^2(\Omega)} := \int_\Omega u(\mathbf{x})\, v(\mathbf{x})\, \mathrm{d}\mathbf{x}$ is the standard scalar product on $L^2(\Omega)$. The details can be found in literature, cf. [11, 13, 12] and others.

For finding the approximate solution to this problem, we discretize the problem using finite elements. We assume to have for $k = 0, 1, 2, \ldots$ a sequence of rectangular grids partitioning the given domain Ω. The finite dimensional subspaces $V_k \subset V$ consist of continuous functions which are bilinear on each rectangular element. Here, the dimension N_k depends on the grid level k. By Galerkin principle, we replace the original Hilbert space V by the subspaces V_k in Problem 2.2. Assuming to have a nodal basis $\Phi_k := (\varphi_{k,i})_{i=1}^{N_k}$ for V_k, we can rewrite the (discretized) optimality system in matrix-vector notation as follows:

$$\underbrace{\begin{pmatrix} M_k & K_k \\ K_k & -\alpha^{-1}M_k \end{pmatrix}}_{\mathcal{A}_k :=} \underbrace{\begin{pmatrix} \underline{y}_k \\ \underline{p}_k \end{pmatrix}}_{\underline{x}_k :=} = \underbrace{\begin{pmatrix} \underline{g}_k \\ 0 \end{pmatrix}}_{\underline{f}_k :=},$$

where the *mass matrix* M_k and the *stiffness matrix* K_k are given by

$$M_k := ((\varphi_{k,j}, \varphi_{k,i})_{L^2(\Omega)})_{i,j=1}^{N_k} \quad \text{and} \quad K_k := ((\nabla\varphi_{k,j}, \nabla\varphi_{k,i})_{L^2(\Omega)})_{i,j=1}^{N_k},$$

respectively, and the right hand side vector $\underline{g}_k$ is given by

$$\underline{g}_k := ((y_D, \varphi_{k,i})_{L^2(\Omega)})_{i=1}^{N_k}.$$

The symbols $\underline{y}_k$ and $\underline{p}_k$ denote the coordinate vectors of the corresponding functions y_k and p_k with respect to the nodal basis Φ_k.

2.2. **A multigrid method.** In this subsection we briefly introduce the multigrid framework that we want to analyze. Multigrid methods consist of two main parts: smoothing and coarse-grid correction. Intuitively speaking, the smoother is applied in order to reduce the amplitude of high-frequency modes of the defect, whereas the coarse-grid correction takes care of the low-frequency modes of the overall defect. The local Fourier analysis ensures this in a formal way.

Starting from an initial approximation $\underline{x}_k^{(0)}$, one step of the multigrid method with $\nu_{pre} + \nu_{post}$ smoothing steps for solving a discretized equation $\mathcal{A}_k\, \underline{x}_k = \underline{f}_k$ on grid level k is given by:

- Apply ν_{pre} (pre-)smoothing steps

$$\underline{x}_k^{(0,m)} := \underline{x}_k^{(0,m-1)} + \tau\, \hat{\mathcal{A}}_k^{-1}(\underline{f}_k - \mathcal{A}_k\, \underline{x}_k^{(0,m-1)}) \qquad \text{for } m = 1, \ldots, \nu_{pre} \tag{2.1}$$

  with $\underline{x}_k^{(0,0)} := \underline{x}_k^{(0)}$, where the choice of the damping parameter τ and the preconditioner $\hat{\mathcal{A}}_k$ is discussed below.

- Apply coarse-grid correction
  - Compute the defect $\underline{f}_k - \mathcal{A}_k\, \underline{x}_k^{(0,\nu_{pre})}$ and restrict to the coarser grid (grid level $k-1$)

$$\underline{r}_{k-1}^{(1)} := \mathcal{P}_k^{k-1}(\underline{f}_k - \mathcal{A}_k\, \underline{x}_k^{(0,\nu_{pre})}).$$

  - Solve the problem

$$\mathcal{A}_{k-1}\underline{p}_{k-1}^{(1)} = \underline{r}_{k-1}^{(1)}$$

    on the coarser grid.
  - Prolongate the correction step to the finer grid (level k) and update the iterate

$$\underline{x}_k^{(1,-\nu_{post})} := \underline{x}_k^{(0,\nu_{pre})} + \mathcal{P}_{k-1}^{k}\underline{p}_{k-1}^{(1)}.$$

  If the problem on the coarser grid is solved exactly (two-grid method), then we obtain

$$\underline{x}_k^{(1,-\nu_{post})} := \underline{x}_k^{(0,\nu_{pre})} + \mathcal{P}_{k-1}^{k}\mathcal{A}_{k-1}^{-1}\mathcal{P}_k^{k-1}(\underline{f}_k - \mathcal{A}_k\, \underline{x}_k^{(0,\nu_{pre})}).$$

- Apply ν_{post} (post-)smoothing steps

$$\underline{x}_k^{(1,m)} := \underline{x}_k^{(1,m-1)} + \tau\, \hat{\mathcal{A}}_k^{-1}(\underline{f}_k - \mathcal{A}_k\, \underline{x}_k^{(1,m-1)}) \qquad \text{for } m = -\nu_{post}+1, \ldots, 0 \tag{2.2}$$

  to obtain the next iterate $\underline{x}_k^{(1)} := \underline{x}_k^{(1,0)}$.

Here, the intergrid-transfer operators $\mathcal{P}_{k-1}^{k}$ and $\mathcal{P}_k^{k-1}$ are chosen to be block-diagonal, i.e.,

$$\mathcal{P}_{k-1}^{k} = \begin{pmatrix} P_{k-1}^{k} & \\ & P_{k-1}^{k} \end{pmatrix} \qquad \text{and} \qquad \mathcal{P}_k^{k-1} = \begin{pmatrix} P_k^{k-1} & \\ & P_k^{k-1} \end{pmatrix},$$

which means that the intergrid-transfer is done for the functions y and p separately. Because we have nested spaces, i.e., $V_k \subseteq V_{k+1}$, the intergrid-transfer operators P_{k-1}^{k} and P_k^{k-1} can be chosen in a canonical way: we choose the canonical embedding as P_{k-1}^{k} and its transpose as restriction P_k^{k-1}.

In practice the problem on grid level $k-1$ is handled by applying one step (V-cycle) or two steps (W-cycle) of the proposed method, recursively, and just on the coarse grid level $k = 0$ the problem is solved exactly. We restrict our analysis to two-grid iteration, i.e., we assume that the problem on the coarser grid is solved exactly.

For the choice of the smoother, several possibilities have been proposed in the literature. Here the analysis is carried out for collective point smoothers which have

been proposed in, e.g., [18, 1, 10, 17] to name a few. The smoother is characterized by the following choice of the preconditioning matrix $\hat{\mathcal{A}}_k$:

$$\hat{\mathcal{A}}_k = \begin{pmatrix} \operatorname{diag} M_k & \operatorname{diag} K_k \\ \operatorname{diag} K_k & -\frac{1}{\alpha} \operatorname{diag} M_k \end{pmatrix} \qquad \text{and} \qquad \tau > 0 \text{ properly chosen.}$$

The local Fourier analysis yields how the parameter τ shall be chosen. Moreover, we note that this smoother is easy to implement in an efficient way, see, e.g., [10]. Numerical examples show good behavior of multigrid methods using such iterations as smoothing procedures and have been discussed in, e.g., [1, 10].

3. symbolic local Fourier analysis (sLFA)

In this section, we recall the method of local Fourier analysis and give some preliminary results that are used to derive convergence results for the model problem in Sections 4 and 5. For this purpose in Subsection 3.1, we introduce the iteration matrix and the convergence rate. In the following subsections, we use local Fourier analysis to derive the symbol of the the iteration matrix. In Subsection 3.5, we comment on the use of CAD.

3.1. **Iteration matrix.**

For sake of simplicity, we restrict ourselves in this work to the two-grid analysis, i.e., we assume that the problem is solved exactly on the coarser grid within the approximation step. Still we are interested in convergence properties independent of the grid level k and of the choice of the parameter α.

The main goal of a convergence analysis is to find a (sharp) bound for the convergence rate. The sharp bound for the convergence rate is the smallest factor q such that the norm of the error after the $n+1$-st iterate can be bounded by q times the error after the n-th iterate, i.e., such that

$$\|\underline{x}_k^{(n+1)} - \underline{x}_k\|_X \leq q \, \|\underline{x}_k^{(n)} - \underline{x}_k\|_X$$

is satisfied, where $\underline{x}_k := \mathcal{A}_k^{-1} \underline{f}_k$ is the exact solution and $\|\cdot\|_X$ is an appropriate norm. The choice of this norm is discussed below.

Using (2.1) – (2.2) and $\underline{x}_k = \mathcal{A}_k^{-1} \underline{f}_k$, we obtain

$$\underline{x}_k^{(n+1)} - \underline{x}_k = TG_k^{k-1}(\underline{x}_k^{(n)} - \underline{x}_k),$$

where the iteration matrix TG_k^{k-1} is given by

$$TG_k^{k-1} := S_k^{\nu_{post}} \underbrace{\left(I - \mathcal{P}_{k-1}^k \mathcal{A}_{k-1}^{-1} \mathcal{P}_k^{k-1} \mathcal{A}_k\right)}_{CG_k^{k-1} :=} S_k^{\nu_{pre}}$$

and the iteration matrix of the smoother, S_k, is given by

$$S_k := I - \tau \, \hat{\mathcal{A}}_k^{-1} \mathcal{A}_k.$$

Certainly, the convergence rate can be bounded from above by the matrix norm of the iteration matrix, i.e.,

$$q \leq q_{TG} = \|TG_k^{k-1}\|_X \tag{3.1}$$

holds. This estimate is sharp if we consider the supremum over all possible starting values or, equivalently, all possible right-hand sides. If $q_{TG} < 1$ is satisfied, the method converges for all starting values. If q_{TG} is independent of h_k and α, (or

any other parameter) we have robust and optimal convergence behavior, i.e., the number of iterations is independent of those parameters.

3.2. **Local Fourier analysis framework.** The idea of (local) Fourier analysis is to simplify the problem such that the eigenvectors and the eigenvalues of mass and stiffness matrix can be written down explicitly. Therefore, typically uniform grids are assumed. Whereas more rigorous approaches of Fourier analysis assume the domain to be an interval or a rectangular domain and incorporate corresponding boundary terms, in *local* Fourier analysis the boundary is neglected by assuming periodic boundary conditions.

This allows to extend a bounded domain Ω to the entire space $\mathbb{R}^d$, see [3]. In this section, we restrict for sake of simplicity to the two dimensional case, so we consider the case $\Omega := \mathbb{R}^2$. Let us stress once more that good convergence and smoothing rates computed using local Fourier analysis for simple cases, typically also indicates good behavior of the analyzed methods in more general cases.

We assume to have on each grid-level $k = 0, 1, 2, \ldots$ a uniform grid with nodes

$$\mathbf{x}_{k,n_1,n_2} := (h_k\, n_1, h_k\, n_2) \qquad \text{for } (n_1, n_2) \in \mathbb{Z}^2,$$

where the uniform grid size is given by $h_k = 2^{-k}$. The discretization is done using the Courant element, so the functions in V_k are given by fixing the values on the grid points only, i.e., using a grid function

$$\begin{array}{rccc} \underline{u_k} & : & h_k\,\mathbb{Z}^2 & \to & \mathbb{R} \\ & & \mathbf{x}_{k,n_1,n_2} & \to & u_k(\mathbf{x}_{k,n_1,n_2}). \end{array}$$

Within the quadrilaterals in between, the functions in V_k are bilinear. Certainly, for (bounded) domains the number of nodes is bounded and therefore all the nodes can be enumerated. So, the grid functions are basically just vectors where the dimension equals the number of nodes.

The local Fourier analysis is based on Fourier grid functions, which are defined as follows. For every $\theta \in \Theta := [-\pi, \pi)^2$ and every grid level k, the Fourier grid function $\underline{\varphi_k(\theta)}$ is given by

$$\underline{\varphi_k(\theta)}(\mathbf{x}_{k,\mathbf{n}}) := \mathrm{e}^{\mathrm{i}(\theta\cdot\mathbf{x}_{k,\mathbf{n}})/h_k}.$$

It is easy to see that every grid function can be expressed as linear combination of countable infinitely many Fourier grid functions. In case of a bounded domain, just finitely many Fourier grid functions would be necessary. Nonetheless for the analysis, all $\theta \in \Theta = [-\pi, \pi)^2$ are considered.

3.3. **Symbols of mass and stiffness matrix.** Using standard techniques, one can explicitly write down the matrices M_k and K_k for the discretization which we have introduced in subsection 3.2, see, e.g., [18]. Based on such an explicit representation, we obtain

$$M_k\, \underline{\varphi_k(\theta)} = \overline{M_k}(\theta)\, \underline{\varphi_k(\theta)} \qquad \text{and} \qquad K_k\, \underline{\varphi_k(\theta)} = \overline{K_k}(\theta)\, \underline{\varphi_k(\theta)},$$

where

$$\begin{aligned} \overline{M_k}(\theta_1,\theta_2) = \tfrac{h_k^2}{36}\big(& \mathrm{e}^{-\theta_1\mathrm{i}-\theta_2\mathrm{i}} + 4\mathrm{e}^{-\theta_1\mathrm{i}} + \mathrm{e}^{-\theta_1\mathrm{i}+\theta_2\mathrm{i}} \\ & + 4\mathrm{e}^{-\theta_2\mathrm{i}} + 16 + 4\mathrm{e}^{\theta_2\mathrm{i}} \\ & + \mathrm{e}^{\theta_1\mathrm{i}-\theta_2\mathrm{i}} + 4\mathrm{e}^{\theta_1\mathrm{i}} + \mathrm{e}^{\theta_1\mathrm{i}+\theta_2\mathrm{i}} \big) \end{aligned}$$

and

$$\overline{K_k}(\theta_1,\theta_2) = \tfrac{1}{3}\big(\begin{array}{ccccccc} - & \mathrm{e}^{-\theta_1\mathrm{i}-\theta_2\mathrm{i}} & - & \mathrm{e}^{-\theta_1\mathrm{i}} & - & \mathrm{e}^{-\theta_1\mathrm{i}+\theta_2\mathrm{i}} \\ - & \mathrm{e}^{-\theta_2\mathrm{i}} & + & 8 & - & \mathrm{e}^{\theta_2\mathrm{i}} \\ - & \mathrm{e}^{\theta_1\mathrm{i}-\theta_2\mathrm{i}} & - & \mathrm{e}^{\theta_1\mathrm{i}} & - & \mathrm{e}^{\theta_1\mathrm{i}+\theta_2\mathrm{i}} \end{array} \big).$$

The scalars $\overline{M_k}(\theta)$ and $\overline{K_k}(\theta)$, which are obviously eigenvalues, are called symbols.

The preconditioning matrices $\hat{M}_k = \operatorname{diag} M_k$ and $\hat{K}_k = \operatorname{diag} K_k$ are just scaled identity matrices and the symbol equals the scaling factor, i.e.,

$$\overline{\hat{M}_k}(\theta) = \overline{\hat{M}_k} = \tfrac{4}{9}h_k^2 \qquad \text{and} \qquad \overline{\hat{K}_k}(\theta) = \overline{\hat{K}_k} = \tfrac{8}{3}.$$

3.4. **Symbol of the system matrix $\mathcal{A}_k$ and of the iteration matrix.** In the last subsection we have shown that the vector $\underline{\varphi_k(\theta)}$ is invariant under the action of M_k, K_k, $\hat{M}_k$ and $\hat{K}_k$ for all $\theta \in \Theta$. This can be extended to the block-matrices: for all $\theta \in \Theta$ the linear span of the vectors

$$\begin{pmatrix} \underline{\varphi_k(\theta)} \\ \hline \vec{0} \end{pmatrix} \text{ and } \begin{pmatrix} \vec{0} \\ \underline{\varphi_k(\theta)} \end{pmatrix} \tag{3.2}$$

is invariant under the action of the block matrices $\mathcal{A}_k$ and $\hat{\mathcal{A}}_k$. Again, we can introduce the symbol:

(3.3)

$$\overline{\mathcal{A}_k}(\theta) = \begin{pmatrix} \overline{M_k}(\theta) & \overline{K_k}(\theta) \\ \overline{K_k}(\theta) & -\alpha^{-1}\overline{M_k}(\theta) \end{pmatrix} \text{ and } \overline{\hat{\mathcal{A}}_k}(\theta) = \begin{pmatrix} \overline{\hat{M}_k}(\theta) & \overline{\hat{K}_k}(\theta) \\ \overline{\hat{K}_k}(\theta) & -\alpha^{-1}\overline{\hat{M}_k}(\theta) \end{pmatrix}.$$

Here, the symbol is a 2-by-2 matrix and therefore it cannot be explained as an eigenvalue anymore. The symbol $\overline{\mathcal{A}_k}(\theta)$ is the representation of the block matrix $\mathcal{A}_k$ with respect to the basis formed by the vectors in (3.2).

Using the symbols of $\mathcal{A}_k$ and $\hat{\mathcal{A}}_k$, we derive the symbol of the iteration matrix of the smoother:

$$\overline{S_k}(\theta) = I - \tau\, \overline{\hat{\mathcal{A}}_k}(\theta)^{-1}\overline{\mathcal{A}_k}(\theta).$$

As we are interested in a whole two-grid step, we also have to take the matrix $\mathcal{A}_{k-1}$, i.e., the system matrix on the coarser grid, and the intergrid transfer operators into account. The coarse-grid correction operator does not preserve the linear span given by the vectors in (3.2). One can verify that for all $\theta \in \Theta^{(low)} := [-\pi/2, \pi/2)^2$, the restriction operator P_k^{k-1} maps both basis functions,

$$\underline{\varphi_k(\theta_1,\theta_2)}, \underline{\varphi_k(\overline{\theta_1},\theta_2)} \text{ and } \underline{\varphi_k(\theta_1,\overline{\theta_2})}, \underline{\varphi_k(\overline{\theta_1},\overline{\theta_2})}, \tag{3.4}$$

to the same function

$$\underline{\varphi_{k-1}(2\theta_1, 2\theta_2)} \tag{3.5}$$

on the coarser grid. Here and in what follows, $\overline{\theta}$ is given by

$$\overline{\theta} := \begin{cases} \theta + \pi & \text{for} \quad \theta < 0, \\ \theta - \pi & \text{for} \quad \theta \geq 0. \end{cases}$$

Such a result also holds for the prolongation operator P_k^{k-1}: this operator maps the function given in (3.5) to a combination of functions in (3.4). Therefore, we

cannot represent the two-grid correction operator with respect to the basis stated in (3.2) but with respect to the basis

$$\begin{aligned}(3.6)\qquad &\begin{pmatrix} \varphi_k(\theta_1,\theta_2) \\ \hline \vec{0} \end{pmatrix}, \begin{pmatrix} \vec{0} \\ \hline \varphi_k(\theta_1,\theta_2) \end{pmatrix}, \begin{pmatrix} \varphi_k(\overline{\theta_1},\theta_2) \\ \hline \vec{0} \end{pmatrix} \begin{pmatrix} \vec{0} \\ \hline \varphi_k(\overline{\theta_1},\theta_2) \end{pmatrix}, \\ &\begin{pmatrix} \varphi_k(\theta_1,\overline{\theta_2}) \\ \hline \vec{0} \end{pmatrix}, \begin{pmatrix} \vec{0} \\ \hline \varphi_k(\theta_1,\overline{\theta_2}) \end{pmatrix}, \begin{pmatrix} \varphi_k(\overline{\theta_1},\overline{\theta_2}) \\ \hline \vec{0} \end{pmatrix}, \begin{pmatrix} \vec{0} \\ \hline \varphi_k(\overline{\theta_1},\overline{\theta_2}) \end{pmatrix},\end{aligned}$$

see, e.g., [18, 1]. The symbol of the matrix $\mathcal{A}_k$ with respect to this basis has a block-diagonal form:

$$(3.7)\quad \overline{\overline{\mathcal{A}_k}}(\theta_1,\theta_2) = \begin{pmatrix} \overline{\mathcal{A}_k}(\theta_1,\theta_2) & & & \\ & \overline{\mathcal{A}_k}(\overline{\theta_1},\theta_2) & & \\ & & \overline{\mathcal{A}_k}(\theta_1,\overline{\theta_2}) & \\ & & & \overline{\mathcal{A}_k}(\overline{\theta_1},\overline{\theta_2}) \end{pmatrix} \in \mathbb{R}^{8\times 8}.$$

The symbol of the intergrid transfer operator has a rectangular form, as the intergrid transfer operator maps from the basis given in (3.6) (for some (θ_1,θ_2)) to the basis given in (3.2) (for $(2\theta_1, 2\theta_2)$).

$$\overline{\mathcal{P}^k_{k-1}}(\theta) = \begin{pmatrix} \overline{P^k_{k-1}}(\theta_1,\theta_2) & \\ & \overline{P^k_{k-1}}(\theta_1,\theta_2) \\ \overline{P^k_{k-1}}(\overline{\theta_1},\theta_2) & \\ & \overline{P^k_{k-1}}(\overline{\theta_1},\theta_2) \\ \overline{P^k_{k-1}}(\theta_1,\overline{\theta_2}) & \\ & \overline{P^k_{k-1}}(\theta_1,\overline{\theta_2}) \\ \overline{P^k_{k-1}}(\overline{\theta_1},\overline{\theta_2}) & \\ & \overline{P^k_{k-1}}(\overline{\theta_1},\overline{\theta_2}) \end{pmatrix},$$

where $\overline{P^k_{k-1}}$ is given by

$$\begin{aligned}\overline{P^k_{k-1}}(\theta_1,\theta_2) = \tfrac{1}{4}(\quad & \mathrm{e}^{-\theta_1\mathrm{i}-\theta_2\mathrm{i}} + 2\mathrm{e}^{-\theta_1\mathrm{i}} + \mathrm{e}^{-\theta_1\mathrm{i}+\theta_2\mathrm{i}} \\ + \quad & 2\mathrm{e}^{-\theta_2\mathrm{i}} + 4 + 2\mathrm{e}^{\theta_2\mathrm{i}} \\ + \quad & \mathrm{e}^{\theta_1\mathrm{i}-\theta_2\mathrm{i}} + 2\mathrm{e}^{\theta_1\mathrm{i}} + \mathrm{e}^{\theta_1\mathrm{i}+\theta_2\mathrm{i}} \quad).\end{aligned}$$

Using these matrices, we can represent the symbol of the two-grid operator by

$$\overline{TG^{k-1}_k}(\theta) = \overline{\overline{S_k}}(\theta)^{\nu_{post}} \left(I - \overline{\mathcal{P}^k_{k-1}}(\theta) \left(\overline{\mathcal{A}_{k-1}}(2\theta)\right)^{-1} \left(\overline{\mathcal{P}^k_{k-1}}(\theta)\right)^T \overline{\overline{\mathcal{A}_k}}(\theta) \right) \overline{\overline{S_k}}(\theta)^{\nu_{pre}}.$$

Here, $\overline{\mathcal{A}_{k-1}}(2\theta)$ is a 2-by-2 matrix defined in (3.3), and $\overline{\overline{\mathcal{A}_k}}(\theta)$ is an 8-by-8 matrix, as introduced in (3.7). Also the symbol $\overline{\overline{S_k}}(\theta)$ is a block-diagonal 8-by-8 matrix, given analogously to (3.7). A similar analysis was done in [1], cf. Theorem 5.1 in their work. General remarks concerning local Fourier analysis can be found in [18], the application to the model problem, as done in this paper, is a straight-forward application of the techniques presented there.

3.5. **Estimating the convergence rate.** Based on the local Fourier analysis framework introduced in the last subsections, q_{TG}, the bound of the convergence

rate, introduced in (3.1), can be computed explicitly using the symbol. Using a proper choice of the norms we obtain

$$q_{TG} = \|TG_k^{k-1}\|_X = \sup_{\theta\in\Theta} \|\overline{TG_k^{k-1}}(\theta)\|_{\overline{X}}, \tag{3.8}$$

i.e., the quantity of our interest is the supremum of the norm of the symbol. In fact, for a given norm $\|\cdot\|_{\overline{X}}$ the identity (3.8) can be used as a definition for the norm $\|\cdot\|_X$, cf. Parseval's identity. In the case of our model problem, the norm of is a function of the frequency θ, the grid size h_k, the damping parameter τ and the parameter α, cf. the next two sections.

So far, such a supremum was typically approximated numerically. In the next two sections we show that the norm can also be computed using CAD which allows to represent the relation between the mentioned parameters and the convergence rate in a closed form.

4. An all-at-once analysis for the one dimensional case

We start the analysis with the one dimensional case, where we will omit some of the computational details and rather discuss the main steps of the procedure. We have summarized the full computations in a Mathematica notebook[1]. Note that the symbols of all involved matrices for the one dimensional case can be computed in a similar way as it was done in Section 3. The details can be found in [16].

As mentioned in the last section, the iteration matrix TG_k^{k-1} describes the convergence properties of the desired multigrid method. Here, we assume $\nu_{pre} = \nu_{post} = 1$ and are interested in computing the convergence rate $q_{TG}(\tau, h_k, \alpha) = \|TG_k^{k-1}\|_X$ of the two-grid iteration. For defining the norm $\|\cdot\|_X$ we use (3.8) and fix the norm $\|\cdot\|_{\overline{X}}$ in the Fourier space:

$$\|(\underline{y}_k, \underline{p}_k)\|_{\overline{X}}^2 := \|\underline{y}_k\|_{\ell^2}^2 + \alpha^{-1}\|\underline{p}_k\|_{\ell^2}^2.$$

$\|\cdot\|_{\overline{X}}$, applied to matrices, is the corresponding operator norm. This definition mainly measures the scaling between the involved components, y and p.

This choice guarantees that the norm $\|\cdot\|_X$ has a product structure:

$$\|(\underline{y}_k, \underline{p}_k)\|_X^2 = \|\underline{y}_k\|_V^2 + \alpha^{-1}\|\underline{p}_k\|_V^2,$$

where the norm $\|\cdot\|_V$ is independent of the choice of the parameters τ, h_k and α. The norm of TG_k^{k-1} still may depend on τ, h_k and α and as we are interested in a worst-case scenario, we take the supremum also with respect to h_k and α. For obvious reasons, the supremum is not taken with respect to the damping parameter τ, which shall be chosen in an optimal way.

The convergence rate $q_{TG}(\tau)$ is therefore given by

$$q_{TG}^2(\tau) = \sup_{h_k>0} \sup_{\alpha>0} \sup_{\theta\in\Theta} \|\overline{TG_k^{k-1}}(\theta)\|_{\overline{X}}^2, \tag{4.1}$$

where the norm $\|\overline{TG_k^{k-1}}(\theta)\|_{\overline{X}}$ depends on all of these variables, τ, h_k, α and θ. This norm can be computed in a straight-forward way. In order to make CAD applicable, we rewrite this quantity as a rational function. Therefore we replace $\cos(\theta)$ by a variable c. The expression can be simplified further using the substitutions $\eta :=$

[1]The document is available online at http://www.risc.jku.at/people/vpillwei/sLFA/

h_k^4/α and $\gamma := c^2$, which allows to eliminate h_k and α and to reduce the polynomial degree. We obtain

$$\|\overline{TG_k^{k-1}}(\theta)\|_{\overline{X}}^2 \quad = \quad \sigma\left(\tau, \frac{h_k^4}{\alpha}, \cos^2(\theta)\right),$$

where

$$\begin{aligned}
\sigma(\tau,\eta,\gamma) =& \frac{P_1(\tau,\eta,\gamma)P_2(\tau,\eta,\gamma)}{64(9+\eta)^2(9(-1+\gamma)^2+\eta(1+2\gamma)^2)}, \quad \text{with}\\
P_1(\tau,\eta,\gamma) =& \eta\left(\gamma^2\tau^2+\gamma\left(4-3\tau^2\right)+4(\tau-1)^2\right)\\
&+36\left(\gamma^2\tau^2+\gamma\left(6\tau^2-6\tau+1\right)+(\tau-1)^2\right)\\
P_2(\tau,\eta,\gamma) =& \eta^2\left(\gamma^3\tau^2+\gamma^2\left(4+16\tau-7\tau^2\right)+\gamma\left(8\tau^2-56\tau+52\right)+16(\tau-1)^2\right)\\
&+36\eta\left(2\gamma^3\tau^2+\gamma^2\left(28\tau^2-22\tau+5\right)+\gamma\left(34\tau^2-34\tau+5\right)+8(\tau-1)^2\right)\\
&+1296(\gamma-1)^2\left((\gamma+1)\tau^2-2\tau+1\right).
\end{aligned}$$

Now (4.1) can be rewritten using the function σ as follows:

$$q_{TG}^2(\tau) = \sup_{\eta>0}\sup_{0\le\gamma\le1}\sigma(\tau,\eta,\gamma). \tag{4.2}$$

In [12], the authors have discussed that such a problem can be resolved using quantifier elimination algorithms. As by definition the supremum is the least upper bound, $q_{TG}^2(\tau)$ is the smallest λ satisfying

$$\forall\ \eta>0\ :\ \forall\ 0\le\gamma\le1\ :\ \sigma(\tau,\eta,\gamma)\le\lambda. \tag{4.3}$$

The above is a quantified formula with rational content and we are interested in finding an equivalent quantifier-free condition on λ as a function in τ. This problem is known to be solvable using Cylindrical Algebraic Decomposition (CAD) [5, 6, 9]. Even though this method provably is correct and terminates on any given input fitting to the framework, the underlying computations are very involved. The computational complexity depends heavily on the input parameters such as the number of inequalities, the polynomial degrees (if the rational input is rephrased as a logical combination of polynomial inequalities) and the number of variables. In the worst case it is doubly exponential in the number of variables and this worst-case bound is not only met in theory, but often experienced in practice. There are several implementations of CAD available [4, 14, 15], for this work we use the Mathematica built-in commands "CylindricalDecomposition" and "Resolve". The quantifier-free formula allows to directly express q_{TG} as a piecewise rational function in τ as discussed in [12].

However, the problem (4.3) is too big to allow a direct application of Mathematica's CAD algorithm and it does not terminate within a reasonable amount of time (reasonable including a run-time of more than a week). Therefore strategies to accelerate the computations are necessary, besides the obvious reductions of the number of variables and degrees that was carried out already by substituting $\eta = h_k^4/\alpha$. Note that using CAD we are able to actually find (and thereby also prove) the desired bound – in theory. However, if we can come up with a good guess for the bound (again assisted by CAD), then CAD can be used to prove the guess rigorously. Commonly, proving is easier than finding, not only in the context of CAD. Hence below we first guess the convergence rate using the samples $\eta = 0$ and $\eta \to \infty$ and show that the guess is correct in the second step. This approach is

sufficient to deal with the all-at-once analysis in the one dimensional case. For the two dimensional case discussed in the next section the formula to be proven is still too large to be handled directly in this way. Hence the proof is split into smaller pieces that are reasonably fast, as is detailed below.

Returning to the one dimensional case following the outlined approach, by sampling we obtain

$$\begin{aligned}\sigma_0(\tau,\gamma) &:= \sigma(\tau,0,\gamma) = (1+\tau(-2+\tau+\gamma\tau))((\tau-1)^2+\gamma^2\tau^2+\gamma(1+6(\tau-1)\tau)),\\ \sigma_\infty(\tau,\gamma) &:= \lim_{\eta\to\infty}\sigma(\tau,\eta,\gamma)\\ &= \frac{1}{64(1+2\gamma)^2}(4(\tau-1)^2+\gamma^2\tau^2+\gamma(4-3\tau^2))(16(\tau-1)^2+\gamma^3\tau^2\\ &\quad +\gamma^2(4+16\tau-7\tau^2)+\gamma(52-56\tau+8\tau^2)).\end{aligned}$$

We compute the supremum using Mathematica's CAD algorithm for both cases separately and obtain

$$q_0^2(\tau) := \sup_{0\le\gamma\le1}\sigma_0(\tau,\gamma) \qquad \text{and} \qquad q_\infty^2(\tau) := \sup_{0\le\gamma\le1}\sigma_\infty(\tau,\gamma).$$

Here, both computations are finished after about ten seconds.[2] The results are rather big, which is why they are not displayed here. In particular the splitting points are algebraic numbers with minimal polynomials up to degree 18 or coefficients depending on τ, respectively. Hence we only present a simplified version of the Mathematica input and output for $\eta = 0$ here:

```
In[1]= Resolve[τ > 0 && ForAll[γ, 0 ≤ γ ≤ 1, σ0[τ, γ] ≤ λ], {τ, λ}, Reals]

Out[1]= (0 < τ ≤ τ0&&λ ≥ 2(2τ − 1)^2(2τ^2 − 2τ + 1)) || (τ0 < τ ≤ τ1&&λ ≥ (τ − 1)^4)
        || (τ > τ1&&λ ≥ 2(2τ − 1)^2(2τ^2 − 2τ + 1))
```

Above we use τ_0, τ_1 as abbreviations for the algebraic numbers $\tau_0 \approx 0.247067$ and $\tau_1 \approx 0.584927$ defined by the minimal polynomial $15z^4 - 28z^3 + 22z^2 - 8z + 1$.

As $q_0^2(\tau)$ is the smallest λ satisfying the above formula, the interpretation of the result is that $q_0^2(\tau)$ is a piecewise polynomial in τ:

$$q_0^2(\tau) = \begin{cases} 2(2\tau-1)^2(2\tau^2-2\tau+1), & 0<\tau\le\tau_0\\ (\tau-1)^4, & \tau_0<\tau\le\tau_1\\ 2(2\tau-1)^2(2\tau^2-2\tau+1), & \tau>\tau_1\end{cases}$$

In the same way, we compute $q_\infty^2(\tau)$. The guess for the uniform bound of the two-grid convergence rate defined in (4.2) is then given by the maximum of these two functions, i.e., we define

$$q_{GUESS}(\tau) := \max\{q_0(\tau), q_\infty(\tau)\}.$$

By construction, $q_{TG}(\tau) \ge q_{GUESS}(\tau)$ holds for all τ. In order to show that this guess really is an upper bound for all frequencies, we may again set up a quantified formula

$$\forall\, 0\le\tau\le1 \;:\; \forall\,\eta>0 \;:\; \forall\, 0\le\gamma\le1 \;:\; \sigma(\tau,\eta,\gamma) \le q_{GUESS}^2(\tau). \tag{4.4}$$

Since $q_{GUESS}(\tau)$ is a piecewise defined function, we may split (4.4) into the intervals used in the piecewise definition. Again we use Mathematica's CAD algorithm,

[2]All computations were done on a standard PC (4 cores with 2.67 GHz each). The times are given to indicate the order of magnitude.

which reduces these formulas to the logical constant true, which concludes the proof that $q_{GUESS}(\tau)$ is an upper bound. Each of these steps takes about five to ten seconds. By construction we also know that this upper bound is sharp.

By some straight-forward simplifications, we can rewrite $q_{TG}(\tau) = q_{GUESS}(\tau)$ in a nice closed form and we obtain the following theorem.

Theorem 4.1. *The convergence rate for the two-grid method in the framework stated in this section is given by*

$$q_{TG}(\tau) = \sup_{h_k>0} \sup_{\alpha>0} \|TG_k^{k-1}\|_X = \max\left\{|1-2\tau|\sqrt{2+4(\tau-1)\tau}, \frac{1}{4}(\tau-2)^2\right\}$$

for all $\tau \in [0,1]$*, which can be seen in Figure 1.*

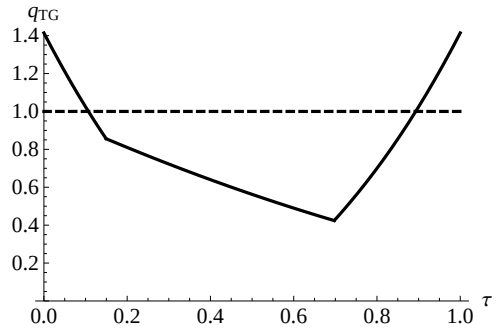

FIGURE 1. Two-grid convergence factor depending on damping parameter

This closed form allows to determine the choices of τ that lead to convergence or as well determine the optimal choice for τ, which is defined by an intersection point of the two contributing bounds. Thus it is again an algebraic number, whose minimal polynomial can be given. We keep in mind that the computed rate is the result of a sharp worst-case analysis of the convergence rate.

5. An analysis for the two dimensional case

In this section, we apply the sLFA as proposed in the last section to a two dimensional domain. The complete two-grid system, however, is too large for being analyzed in an all-at-once way, as this was done for the one dimensional case in the previous section.

The approach, we follow here, is to split the original problems into subproblems. A common approach is based on the natural splitting of the iteration matrix of the two-grid-operator into the iteration matrix of the smoothing steps and the iteration matrix of the coarse-grid-correction step. This can be done either in a multiplicative way using a pair of norms as introduced by Hackbusch, cf. [7], or based on splitting the domain of frequencies θ of the Fourier modes. The latter alternative is more suitable in terms of local Fourier analysis and therefore we stick to that approach. Again, we refer for the computational details to the Mathematica notebook[3] provided by the authors.

[3]The document is online available at http://www.risc.jku.at/people/vpillwei/sLFA/

First remember that the symbol of the two-grid operator was given by

$$\overline{TG_k^{k-1}}(\theta) = \overline{\overline{S_k}}(\theta)^{\nu_{post}}\, \overline{CG_k^{k-1}}(\theta)\, \overline{\overline{S_k}}(\theta)^{\nu_{pre}}.$$

We are interested in computing the supremum of $\|\overline{TG_k^{k-1}}(\theta)\|_{\overline{X}}$ for all $\theta \in \Theta^{(low)} := [-\pi/2, \pi/2)^2$, all grid sizes $h_k > 0$ and all $\alpha > 0$. For sake of simplicity, we restrict ourselves here to $\nu_{pre} = \nu_{post} = \nu/2 \in \mathbb{N}$.

Because of the semi-multiplicativity of any operator norm, we obtain

$$\left\|\overline{TG_k^{k-1}}(\theta)\right\|_{\overline{X}} \leq \left\|\mathcal{I}(q_1,q_2)^{\nu/2}\, \overline{CG_k^{k-1}}(\theta)\, \mathcal{I}(q_1,q_2)^{\nu/2}\right\|_{\overline{X}} \left\|\mathcal{I}(q_1,q_2)^{-1}\, \overline{\overline{S_k}}(\theta)\right\|_{\overline{X}}^{\nu},$$

where $\mathcal{I}(q_1,q_2)$ could be any matrix. We use the following ansatz:

$$\mathcal{I}(q_1,q_2) := \begin{pmatrix} q_1 & & & & & & & \\ & q_1 & & & & & & \\ & & q_2 & & & & & \\ & & & q_2 & & & & \\ & & & & q_2 & & & \\ & & & & & q_2 & & \\ & & & & & & q_2 & \\ & & & & & & & q_2 \end{pmatrix}, \tag{5.1}$$

where $q_1,\ q_2 > 0$. The fact that both, $\mathcal{I}(q_1,q_2)$ and $\overline{\overline{S_k}}(\theta)$, are block-diagonal guarantees

$$\begin{aligned}\left\|\mathcal{I}(q_1,q_2)^{-1}\, \overline{\overline{S_k}}(\theta)\right\|_{\overline{X}} \leq \max\{&q_1^{-1}\|\overline{S}_k(\theta_1,\theta_2)\|_{\overline{X}}, q_2^{-1}\|\overline{S}_k(\overline{\theta_1},\theta_2)\|_{\overline{X}},\\ &q_2^{-1}\|\overline{S}_k(\theta_1,\overline{\theta_2})\|_{\overline{X}}, q_2^{-1}\|\overline{S}_k(\overline{\theta_1},\overline{\theta_2})\|_{\overline{X}}\}.\end{aligned}$$

Because of $\theta := (\theta_1,\theta_2) \in \Theta^{(low)}$ and

$$\{(\overline{\theta_1},\theta_2), (\theta_1,\overline{\theta_2}), (\overline{\theta_1},\overline{\theta_2})\} \subset \Theta^{(high)} := [-\pi,\pi)^2 \backslash [-\pi/2,\pi/2)^2,$$

the ansatz

$$q_1 := \sup_{\alpha>0}\sup_{h_k>0}\sup_{\theta\in\Theta^{(low)}} \|\overline{S_k}(\theta)\|_{\overline{X}} \qquad \text{and} \qquad q_2 := \sup_{\alpha>0}\sup_{h_k>0}\sup_{\theta\in\Theta^{(high)}} \|\overline{S_k}(\theta)\|_{\overline{X}}$$

guarantees

$$\left\|\mathcal{I}(q_1,q_2)^{-1}\, \overline{\overline{S_k}}(\theta)\right\|_{\overline{X}} \leq 1.$$

Therefore

$$q_{TG} = \|\overline{TG_k^{k-1}}(\theta)\|_{\overline{X}} \leq \widetilde{q_{TG}}(q_1^{\nu/2}, q_2^{\nu/2}) := \|\mathcal{I}(q_1^{\nu/2}, q_2^{\nu/2})\, \overline{CG_k^{k-1}}(\theta)\, \mathcal{I}(q_2^{\nu/2}, q_2^{\nu/2})\|_{\overline{X}}$$

holds for all $\theta \in \Theta^{(low)}$, all $h_k > 0$ and all $\alpha > 0$. In Subsection 5.1 below we show that

$$q_1 \leq 1 \qquad \text{and} \qquad q_2 \leq q_{SM} < 1. \tag{5.2}$$

The quantity q_{SM} above is called the *smoothing rate*. The smoothing rate indicates the reduction of the amplitude of the high frequency error modes performed by the smoother. Due to the fact that typically $q_1 = 1$ holds, we write in a short notation $q_{TG}(q_2^{\nu/2}) := q_{TG}(1, q_2^{\nu/2})$ and $\mathcal{I}(q_2^{\nu/2}) := \mathcal{I}(1, q_2^{\nu/2})$. In Subsection 5.2 we show that, provided $q_1 = 1$ and $q_2 \leq q_{SM} < 1$, an optimal and robust convergence result can be shown, i.e., that $\widetilde{q_{TG}}(q_2^{\nu/2}) < 1$.

Analyzing the smoothing rate is standard, cf. [18, 1]. Typically such rates are not used to compute overall convergence rates as outlined above because the local

Fourier analysis is usually not done using symbolic computation. Commonly, the convergence rates are approximated numerically which is also possible directly for the whole two-grid operator.

Nonetheless, the separation of the analysis for smoother and coarse-grid approximation done in this section simplifies the comparison of different kinds of smoothers and the analysis for varying numbers of smoothing steps ν. Moreover, the separation decomposes the original problem to smaller subproblems which seems to be the key for extending the results presented here to more dimensions. (This is not an issue if convergence rates are approximated numerically because there the complexity of the analysis is not growing exponentially with the size of the problem.)

5.1. **Smoothing rate.** The main goal of this section is to compute q_{SM}, as introduced in (5.2). We have

$$q_{SM} = q_{SM}(\tau) := \sup_{\theta\in\Theta^{(high)}} \sup_{h_k>0} \sup_{\alpha>0} \| \underbrace{I - \tau \left(\overline{\hat{\mathcal{A}}_k}(\theta)\right)^{-1} \overline{\mathcal{A}}_k(\theta)}_{\overline{S}_k(\theta) =} \|_{\overline{X}}.$$

Analogously to the previous section we replace $\cos\theta_1$ and $\cos\theta_2$ by real variables c_1 and c_2, respectively. The condition $(\theta_1,\theta_2) \in \Theta^{(high)}$ is equivalent to $(c_1,c_2) \in [-1,1]^2\backslash(0,1]^2$. Because of symmetry in c_1 and c_2, it suffices without loss of generality to consider only $(c_1,c_2) \in D := [-1,0]\times[-1,1]$. As in the last section, we replace h_k^4/α by η to reduce the number of variables. Using these substitutions, we obtain

$$q_{SM}(\tau) = \sup_{(c_1,c_2)\in D} \sup_{\eta>0} \sigma(\tau,\eta,c_1,c_2),$$

where σ is given by

$$\sigma(\tau,\eta,c_1,c_2) := \frac{\eta((2+c_1)(2+c_2)\tau-4)^2 + 36(4+(c_1+c_2+2c_1c_2-4)\tau)^2}{16(36+\eta)}.$$

With this rewriting the function of our interest is a rational function and we can invoke CAD to determine $q(\tau)$. For this purpose we have used the CAD-implementation in Mathematica, which took about seven minutes, This shows the following result.

Theorem 5.1. *The smoothing rate for the collective Jacobi iteration in the framework stated in this section is given by*

$$q_{SM}(\tau) = \begin{cases} 1-\frac{\tau}{4} & \text{for} \quad 0 \le \tau \le \frac{8}{7} \\ \frac{3\tau}{2}-1 & \text{for} \quad \frac{8}{7} < \tau \le \frac{4}{3} \end{cases}.$$

for all $\tau \in [0,\frac{4}{3}]$.

Since CAD results in an equivalent reformulation, we know that these bounds on the smoothing rate are sharp. The graph of the function q_{SM} can be seen in Figure 2. We see that $q_{SM}(\tau)$ takes its minimum for $\tau = \frac{8}{7}$ with value $q_{SM}\left(\frac{8}{7}\right) = \frac{5}{7} \approx 0.714$. For the canonical choice $\tau = \frac{1}{2}$, we obtain $q_{SM}\left(\frac{1}{2}\right) = \frac{7}{8} \approx 0.875$. So we have shown, that there is a q_{SM} such that $q_2 \le q_{SM} < 1$. It remains to show that $q_1 \le 1$, i.e., that

$$\sup_{\theta\in\Theta^{(low)}} \sup_{h_k>0} \sup_{\alpha>0} \sigma(\theta,h_k,\alpha,\tau) \le 1$$

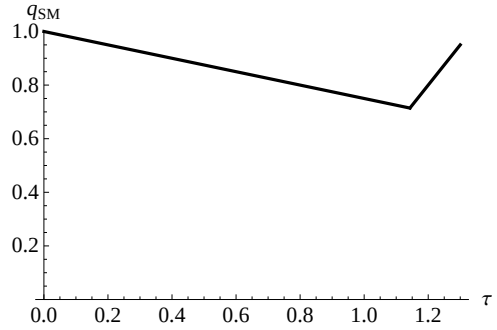

FIGURE 2. Smoothing factor depending on damping parameter τ

holds for all $\tau > 0$. Using the same reformulations as for $\theta \in \Theta^{(high)}$, we can again compute the supremum using Mathematica's CAD algorithm in about 22 minutes. We can observe that for $0 \leq \tau \leq \frac{8}{9}$ the supremum is equal to 1, which shows $q_1 \leq 1$.

As an alternative we could simply use CAD to reformulate the formula

$$\forall\, \theta \in \Theta^{(low)} \,:\, \forall\, h_k > 0 \,:\, \forall\, \alpha > 0 \,:\, \sigma(\theta, h_k, \alpha, \tau) \leq 1$$

in quantifier-free formulation, which leads to $0 \leq \tau \leq \frac{8}{9}$. This is done by Mathematica's CAD algorithm in less than one second to the logical constant true, which also shows $q_1 \leq 1$.

5.2. **Two-grid convergence rate.** As mentioned at the beginning of this section, here we have to analyze

$$\widetilde{q_{TG}}(q) := \sup_{\theta \in \Theta^{(low)}} \sup_{h_k > 0} \sup_{\alpha > 0} \|\mathcal{I}(q)\, \overline{CG_k^{k-1}}(\theta) \mathcal{I}(q)\|_{\overline{X}}.$$

Again, we compute

$$\sigma(\theta, h_k, \alpha, q) := \|\mathcal{I}(q)\, \overline{CG_k^{k-1}}(\theta) \mathcal{I}(q)\|_{\overline{X}}^2$$

first. As in the last section, we replace h_k^4/α by η to reduce the number of variables: the matrix depends on the real variables η, c_1, c_2 and q. Unfortunately, we were not able to compute the norm of that matrix by direct computation in an acceptable time. Certainly, we have

$$\begin{aligned} &\|\mathcal{I}(q)\, \overline{CG_k^{k-1}}(\theta) \mathcal{I}(q)\|_{\overline{X}}^2 \\ &\quad = \lambda_{\max}\Big(\underbrace{\mathcal{L}_k^{-1/2} \mathcal{I}(q)\, \overline{CG_k^{k-1}}(\theta) \mathcal{I}(q)\, \mathcal{L}_k \mathcal{I}(q)\, \overline{CG_k^{k-1}}(\theta) \mathcal{I}(q)\, \mathcal{L}_k^{-1/2}}_{\mathcal{N}_k :=} \Big), \end{aligned}$$

where the matrix $\mathcal{L}_k$ represents the norm $\|\cdot\|_{\overline{X}}$. Hence we need to compute the maximal eigenvalue of $\mathcal{N}_k$ which is a matrix with rational entries in all the parameters c_1, c_2, η and q. Here we face another computational challenge, since this task is too large to be handled directly by Mathematica's "Eigenvalues" command. Thus we use interpolation in q to first determine a guess for the eigenvalue and verify it in a separate step. This can be done, because the matrix has entries that are polynomial in q with rational coefficients depending on the other variables. If we multiply the matrix with the common denominator, the matrix entries are polynomials in all the variables with polynomial degree in q up to at most four. (For

the exact form see the Mathematica notebook.) For the polynomial interpolation to guess the eigenvalues of the matrix we used a degree bound that was observed for specific values of c_1, c_2, and η. It turns out that with multiplicity two each the eigenvalues are given by

$$0,\ q^4,\ \left(e(q)+\sqrt{d(q)}\right) \text{ and } \left(e(q)-\sqrt{d(q)}\right),$$

where e and d are rational functions in the unknowns c_1, c_2 and η that are too large to be displayed here. This guess was verified by comparing it to the characteristic polynomial of the matrix. Once we have the eigenvalues, it is readily verified that

$$\tilde{\sigma}(c_1, c_2, \eta, q) = e(q) + \sqrt{d(q)}$$

is the maximal eigenvalue and that it is symmetric in c_1 and c_2. Thus the domain that needs to be considered for taking the supremum is

$$\mathcal{D} := \{(c_1, c_2, \eta) \mid 0 \le c_1 \le c_2 < 1 \ \wedge \ \eta > 0\},$$

i.e.,

$$\widetilde{q_{TG}}(q)^2 = \sup_{(c_1,c_2,\eta)\in\mathcal{D}} \tilde{\sigma}(c_1, c_2, \eta, q).$$

As discussed in the previous section, we note that in theory quantifier elimination using CAD is capable of finding an upper bound for the maximal eigenvalue $\tilde{\sigma}(c_1, c_2, \eta, q)$, but in this particular case the memory and time requirements are too big. Hence, once more, we have to come up with a plausible guess for the bounds that we need to prove later. As before, since we are aiming at sharp bounds, it needs to be met at certain places. In order to obtain this conjectured bound we have a look at the extreme cases with respect to the frequencies c_1, c_2, i.e., for $(c_1, c_2) \in \mathcal{C} := \{(0,0), (0,1), (1,0), (1,1)\}$. In all these cases, $\tilde{\sigma}(c_1, c_2, \eta, q)$ is independent of η. Note that the function $\tilde{\sigma}$ is not defined for $(c_1, c_2, \eta) = (1, 1, 0)$. Thus we consider the limit

$$\lim_{c_2\to 1} \tilde{\sigma}(1, c_2, 0, q),$$

which gives another candidate for the upper bound. The maximum of all these samples is $q^2_{GUESS}(q)$ defined by

$$q_{GUESS}(q) := \begin{cases} \left(\frac{q^2+3}{4}\right), & 0 < q \le Q, \\ q\sqrt{q^2+1}, & Q < q < 1, \end{cases}$$

where $Q := \sqrt{\frac{1}{15}(4\sqrt{10}-5)}$ is the intersection point of the two bounds. This is what we use as our conjectured bound, i.e., we guess that $\tilde{\sigma}(c_1, c_2, \eta, q) \le q^2_{GUESS}(q)$ is true for all $(c_1, c_2, \eta, q) \in \mathcal{D} \times (0, 1)$. It remains to prove that this guess is correct, where we note again that, if we have shown that $q_{GUESS}(q)$ is an upper bound, we know by construction that it is sharp.

So, we have to prove that

$$\tilde{\sigma}(c_1, c_2, \eta, q) = e(q) + \sqrt{d(q)} \le q^2_{GUESS}(q)$$

is satisfied for all c_1, c_2, η and q in the domain of interest. First, observe that it is easily verified that the denominator D is positive in the given range (which is quickly confirmed by CAD). Therefore it suffices to show

$$D\,(e(q) + \sqrt{d(q)}) \le D\,q^2_{GUESS}(q) \tag{5.3}$$

Then, in order not having to deal with the square root, we use that

$$y \geq 0 \quad \wedge \quad x^2 \leq y^2 \quad \Rightarrow \quad x \leq y.$$

Hence for each of the intervals $q \in (0, Q]$ and $q \in (Q, 1)$ we need to show that

$$R(c_1, c_2, q, \eta) := D\,(q^2_{GUESS}(q) - e(q)) \geq 0 \quad , \qquad \text{and} \tag{5.4}$$

$$S(c_1, c_2, q, \eta) := D^2\,((q^2_{GUESS}(q) - e(q))^2 - d(q)) \geq 0 \tag{5.5}$$

to obtain (5.3). As mentioned earlier, this task is still too complicated to be handled by CAD directly. Hence we break down the computation into smaller pieces by deriving sufficient conditions in each step that we can verify easily using CAD, where easily means bounded by 100 seconds computation time for each substep.

Here, we only sketch the arguments for proving the bounds where we go into slightly more detail for the left bound, i.e., we assume that $0 < q \leq Q$. The proof of the right bound (i.e., for $Q < q < 1$) proceeds by exactly the same idea as the the proof of the left bound and is omitted here.

First note that $R(c_1, c_2, q, \eta)$ is a linear polynomial in η, say

$$R(c_1, c_2, q, \eta) = r_0(c_1, c_2, q^2) + r_1(c_1, c_2, q^2)\eta.$$

Here we use the fact that all the quantities depend on even powers of q only.

Since $\eta > 0$, a sufficient condition for (5.4) is that the coefficients of η are non-negative. So, a sufficient condition for $R \geq 0$ is if the coefficients r_0, r_1 are non-negative, and it turns out that indeed we run into this lucky case. This, however, cannot be shown directly so easily. Hence we iterate the simple idea of finding a necessary condition for the given coefficient polynomials r_0 and r_1. These coefficients are polynomials in c_1, c_2 and q_2 (representing q^2) with degrees $6, 6$ and 2, respectively. Hence we choose q_2 as the main variable. Then we derive necessary conditions using CAD on the coefficients a_i of a generic quadratic polynomial $p_2(x)$ such that

$$p_2(x) = a_0 + a_1 x + a_2 x^2 \geq 0, \qquad 0 < x < Q^2,$$

and finally, we verify that our specific coefficients satisfy these conditions using CAD. This breaking down into smaller subproblems that yield sufficient conditions for the bound to be proven is a very simple idea. Still, it turned out to be the key to obtain the proof in reasonable time with reasonable memory consumption. We simplify the amount of computations considerably, if we exploit the following basic logical rules

$$\begin{aligned} A \Rightarrow (B \wedge C) \quad &\Leftrightarrow \quad (A \Rightarrow B) \wedge (A \Rightarrow C), \quad \text{and} \\ A \Rightarrow (B \vee C) \quad &\Leftrightarrow \quad (A \Rightarrow B) \vee (A \Rightarrow C). \end{aligned} \tag{5.6}$$

In the second step we have to show (5.5). Here, the left hand side factors into

$$\tfrac{1}{16}(q^2 - 1)\, D\, S(c_1, c_2, q, \eta),$$

where S is a polynomial. Since $q^2 - 1 < 0$, it suffices to show that

$$S(c_1, c_2, q, \eta) \leq 0.$$

Again, we observe that S is linear in η, i.e., there are polynomials s_0 and s_1, both independent of η, such that

$$S(c_1, c_2, q, \eta) = s_0(c_1, c_2, q^2) + s_1(c_1, c_2, q^2)\eta.$$

As a necessary condition, we aim at proving that $s_i(c_1, c_2, q_2) \leq 0$ for $i = 0, 1$. Now the coefficients s_0, s_1 are cubic polynomials in q_2. We proceed as before and

determine necessary conditions on the coefficients a_i of a generic polynomial $p_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ to be non-negative in the given range using CAD that are verified for the particular coefficients using CAD and some manual simplifications by applying the rules (5.6). The procedure for proving the right bound, i.e., the case when $Q < q < 1$, is entirely analogous. Combining these results yields the following theorem.

Theorem 5.2. *We have the following upper bound for the convergence rate for every smoother satisfying the smoothing property* (5.2) *with smoothing rate* q_{SM} *and* $\nu/2$ *pre- and* $\nu/2$ *post-smoothing steps:* $\widetilde{q_{TG}}(q_{SM}^{\nu/2})$*, where the function* $\widetilde{q_{TG}}$ *is given by*

$$\widetilde{q_{TG}}(q) = \begin{cases} \frac{1}{4}(q^2+3) & \text{for} \quad 0 < q < Q \\ q\sqrt{q^2+1} & \text{for} \quad Q \le q < 1, \end{cases}$$

where $Q = \sqrt{\frac{1}{15}(4\sqrt{10}-5)}$.

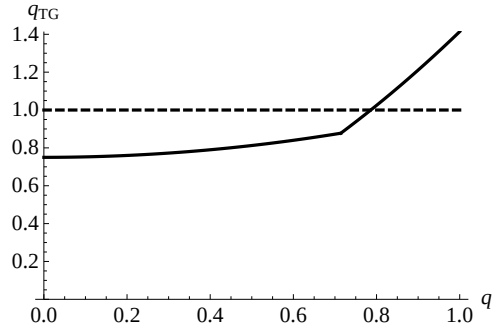

FIGURE 3. Two-grid convergence factor depending on smoothing rate

The combination of Theorems 5.1 and 5.2 gives an upper bound for the two-grid convergence factor, which can be seen for the cases of $\nu = \nu_{pre} + \nu_{post} = 2+2$ steps of the collective Jacobi iteration as smoother in Figure 4. Also this result can be written in closed form, but since the formulas are rather long we refer once more to the Mathematica file.

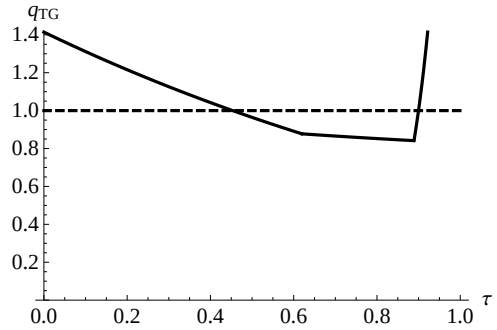

FIGURE 4. Two-grid convergence factor depending on τ for $\nu = \nu_{pre} + \nu_{post} = 2 + 2$ smoothing steps

The given result is an upper bound for the convergence rate. Contrary to the one dimensional case, discussed in Section 4, here the bound is not sharp. In fact, numerical experiments show much faster convergence, cf [17]. Nonetheless, the result gives quantitative results on both, the choice of the damping parameter τ and on ν, the number of smoothing steps that have to be applied. Such results are not obtained using a classical analysis based on smoothing and approximation property.

Acknowledgments. We thank Walter Zulehner for pointing out this interdisciplinary problem.

References

1. A. Borzi, K. Kunisch, and D.Y. Kwak, *Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system*, SIAM J. on Control and Optimization **41(5)** (2003), 1477 – 1497.
2. A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp. **31** (1977), 333 – 390.
3. ———, *Rigorous Quantitative Analysis of Multigrid, I: Constant Coefficients Two-Level Cycle with L_2-Norm*, SIAM J. on Numerical Analysis **31** (1994), no. 6, 1695 – 1730.
4. C.W. Brown, *QEPCAD B – a program for computing with semi-algebraic sets*, Sigsam Bulletin **37** (2003), no. 4, 97 – 108.
5. G.E. Collins, *Quantifier elimination for real closed fields by cylindrical algebraic decomposition*, Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975), Springer, Berlin, 1975, pp. 134 – 183. Lecture Notes in Comput. Sci., Vol. 33.
6. G.E. Collins and H. Hong, *Partial cylindrical algebraic decomposition for quantifier elimination*, Quantifier elimination and cylindrical algebraic decomposition (Linz, 1993), Texts Monogr. Symbol. Comput., Springer, Vienna, 1998, pp. 174–200.
7. W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.
8. H. Hong, R. Liska, and S. Steinberg, *Testing stability by quantifier elimination*, J. Symbolic Comput. **24** (1997), no. 2, 161–187, Applications of quantifier elimination (Albuquerque, NM, 1995).
9. M. Kauers, *How To Use Cylindrical Algebraic Decomposition*, Seminaire Lotharingien de Combinatoire **65** (2011), no. B65a, 1–16.
10. O. Lass, M. Vallejos, A. Borzi, and C.C. Douglas, *Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems*, Computing **84** (2009), no. 1 – 2, 27 – 48.
11. J.L. Lions, *Optimal control of systems governed by partial differential equations*, Berlin-Heidelberg-New York: Springer-Verlag, 1971.
12. V. Pillwein and S. Takacs, *Smoothing analysis of an all-at-once multigrid approach for optimal control problems using symbolic computation*, Numerical and Symbolic Scientific Computing: Progress and Prospects (U. Langer and P. Paule, eds.), Springer, Wien, 2011.
13. J. Schöberl, R. Simon, and W. Zulehner, *A Robust Multigrid Method for Elliptic Optimal Control Problems*, SIAM J. on Numerical Analysis **49** (2011), 1482 – 1503.
14. A. Seidl and T. Sturm, *A generic projection operator for partial cylindrical algebraic decomposition*, Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation (New York), ACM, 2003, pp. 240 – 247 (electronic).
15. A. Strzeboński, *Solving systems of strict polynomial inequalities*, J. Symbolic Comput. **29** (2000), no. 3, 471 – 480.
16. S. Takacs, *All-at-once Multigrid Methods for Optimality Systems Arizing from Optimal Control Problems*, Ph.D. thesis, Johannes Kepler University Linz, Doktoral Program Computational Mathematics, 2012, (in preperation).
17. S. Takacs and W. Zulehner, *Convergence Analysis of Multigrid Methods with Collective Point Smoothers for Optimal Control Problems*, Computing and Visualization in Science **14** (2011), no. 3, 131–141.
18. U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.
19. R. Wienands and W. Joppich, *Practical Fourier analysis for multigrid methods*, Chapman & Hall/CRC, 2005.

Research Institute for Symbolic Computation, Johannes Kepler University Linz, Austria
E-mail address: veronika.pillwein@risc.jku.at

Doctoral Program Computational Mathematics, Johannes Kepler University Linz, Austria
E-mail address: stefan.takacs@dk-compmath.jku.at

Latest Reports in this series

2009 - 2010

[..]

2011

[..]

2011-03	Michael Kolmbauer *Existance and Uniqueness of Eddy Current Problems in Bounded and Unbounded Domains*	May 2011
2011-04	Michael Kolmbauer and Ulrich Langer *A Robust Preconditioned-MinRes-Solver for Distributed Time-Periodic Eddy Current Optimal Control*	May 2011
2011-05	Michael Kolmbauer and Ulrich Langer *A Robust FEM-BEM Solver for Time-Harmonic Eddy Current Problems*	May 2011
2011-06	Markus Kollmann and Michael Kolmbauer *A Preconditioned MinRes Solver for Time-Periodic Parabolic Optimal Control Problems*	August 2011
2011-07	Nicole Spillane, Victorita Dolean, Patrice Hauret, Frédérik Nataf, Clemens Pechstein and Robert Scheichl *Abstract Robust Coarse Spaces for Systems of PDEs via Generalized Eigenproblems in the Overlaps*	November 2011
2011-08	Michael Kolmbauer *A Robust FEM-BEM MinRes Solver for Distributed Multiharmonic Eddy Current Optimal Control Problems in Unbounded Domains*	November 2011
2011-09	Michael Kolmbauer *Efficient Solvers for Multiharmonic Eddy Current Optimal Control Problems with Various Constraints*	November 2011
2011-10	Markus Kollmann, Michael Kolmbauer, Ulrich Langer, Monika Wolfmayr and Walter Zulehner *A Finite Element Solver for a Multiharmonic Parabolic Optimal Control Problem*	December 2011

2012

2012-01	Markus Kollmann and Walter Zulehner *A Robust Preconditioner for Distributed Optimal Control for Stokes Flow with Control Constraints*	January 2012
2012-02	Michael Kolmbauer and Ulrich Langer *A Robust Preconditioned MinRes-Solver for Time-Periodic Eddy Current Problems*	January 2012
2012-03	Wolfgang Krendl, Valeria Simoncini and Walter Zulehner *Stability Estimates and Structural Spectral Properties of Saddle Point Problems*	February 2012
2012-04	Helmut Gfrerer *On Directional Metric Regularity, Subregularity and Optimality Conditions for Nonsmooth Mathematical Programs*	February 2012
2012-05	Veronika Pillwein and Stefan Takacs *A Multigrid Fourier Analysis of a Multigrid Method using Symbolic Computation*	April 2012

From 1998 to 2008 reports were published by SFB013. Please see
`http://www.sfb013.uni-linz.ac.at/index.php?id=reports`
From 2004 on reports were also published by RICAM. Please see
`http://www.ricam.oeaw.ac.at/publications/list/`
For a complete list of NuMa reports see
`http://www.numa.uni-linz.ac.at/Publications/List/`