ÖAW Austrian Academy of Sciences

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

JKU JOHANNES KEPLER UNIVERSITY LINZ

Computing smoothing rates of collective point smoothers for optimal control problems using symbolic computation

Veronika Pillwein Stefan Takacs

DK-Report No. 2010-09 10 2010

A–4040 LINZ, ALTENBERGERSTRASSE 69, AUSTRIA

Supported by

Austrian Science Fund (FWF)

FWF Der Wissenschaftsfonds.

Upper Austria

LAND OBERÖSTERREICH

DK sponsors:

- **Johannes Kepler University Linz** (JKU)
- **Austrian Science Fund** (FWF)
- **Upper Austria**

Computing smoothing rates of collective point smoothers for optimal control problems using symbolic computation

Veronika Pillwein, Stefan Takacs

Abstract The numerical treatment of systems of partial differential equations (PDEs) is of great interest as many problems from applications belong to that class. Also the optimality system of optimal control problems that is discussed in this work has such a structure. These problems are not elliptic and therefore both the construction of an efficient numerical solver and its analysis are hard.
Local Fourier analysis (or local mode analysis) is a widely-used tool to analyze numerical methods for solving discretized systems of PDEs. The rates that can be computed with local Fourier analysis are typically the supremum of some rational function. In several publications this supremum was merely approximated numerically by interpolation. We show that it can be resolved exactly using cylindrical algebraic decomposition which is a well established method in symbolic computation.

1 Introduction

Local Fourier analysis (or local mode analysis) is a commonly used approach for designing and analyzing convergence properties of multigrid methods. It 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. Other work on multigrid theory such as,

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

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

This work is supported by the Austrian Science Fund (FWF) under grants W1214/DK12 and DK6.

e.g., [5], [1] or [8], typically just shows the fact of convergence and does not give sharp or realistic bounds for convergence rates.

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 to more general problems, see, e.g. [2]. In this sense it can be viewed as heuristic approach for a wide class of applications.

Understanding local Fourier analysis as a machinery for analyzing a multigrid method, we apply it in this paper to a model problem and some specific solvers. Still, we keep in mind that this analysis can be carried over to a variety of other problems and solvers. This type of generalization has been carried out, e.g., for methods to solve optimal control problems that have been discussed in [12], [1] and [7]. The method itself is explained in detail, e.g, in [13]. In [13] moreover the local Fourier analysis software *LFA* is presented. This software can be configured using a graphical user interface and allows to approximate (numerically) smoothing and convergence rates based on local Fourier analysis approaches for various problems and multigrid approaches.

The main goal of this paper is to show that the analysis can be carried out in an entirely symbolic way and as such leads to sharp estimates on the smoothing rate for a collective Jacobi relaxation and collective Gauss-Seidel iteration scheme. For this purpose we restrict ourselves to the case of a one dimensional domain, i.e., an interval, and to piecewise linear ansatz functions (Courant elements). Aiming at an audience from both numerical and symbolic mathematics we try to stay at an elementary level and keep this note self-contained.

This paper is organized as follows. In subsection 1.1 we introduce a simple model problem and in subsection 1.2 we propose a multigrid approach to solve the discretized optimality system which is related to this model problem. The local Fourier analysis is introduced and carried out in sections 2 and 4, respectively. Section 3 gives a brief overview on quantifier elimination and cylindrical algebraic decomposition, i.e., on the symbolic methods applied in order to resolve smoothing rates symbolically.

1.1 Model problem

As a *model problem* we consider the following optimal control problem of tracking type: Minimize

$$J(y,u) := \frac{1}{2}\|y-y_D\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|u\|_{L^2(\Omega)},$$

subject to the elliptic boundary value problem (BVP)

$$-\Delta y = u \text{ in } \Omega \qquad \text{and} \qquad y = 0 \text{ on } \partial\Omega, \tag{1}$$

where $y \in H_0^1(\Omega)$ is the state variable and $u \in L^2(\Omega)$ is the control variable. The function $y_D \in L^2(\Omega)$ is given and $\alpha > 0$ is some fixed regularization or cost parameter. Here, Ω is a given domain with boundary $\partial\Omega$. The Banach space $L^2(\Omega)$ is the set of square integrable functions on Ω and the Sobolev space $H_0^1(\Omega)$ is the set of L^2-functions vanishing on the boundary with weak derivatives in $L^2(\Omega)$.

Note that for this setting the boundary value problem is (in weak sense) uniquely solvable in y for every given control u. At first we rewrite the BVP (1) in variational form: Find $y \in H_0^1(\Omega)$ such that

$$(y,p)_{H^1(\Omega)} = (u,p)_{L^2(\Omega)}$$

holds for all $p \in H_0^1(\Omega)$. Here $(f,g)_{L^2(\Omega)} = \int_\Omega f(x)\, g(x)\, \mathrm{d}x$ is the standard L^2-inner product and $(f,g)_{H^1(\Omega)} = (\nabla f, \nabla g)_{L^2(\Omega)}$ denotes the H^1-inner product.

Solving the model problem is equivalent to finding a saddle point of the Lagrange functional which leads to the first order optimality conditions (the Karush-Kuhn-Tucker system or, in short, *KKT system*), given by: Find $(y,u,p) \in H_0^1(\Omega) \times L^2(\Omega) \times H_0^1(\Omega)$ such that

$$\begin{aligned} (y,\widetilde{y})_{L^2(\Omega)} \qquad\qquad + (p,\widetilde{y})_{H^1(\Omega)} &= (y_D,\widetilde{y})_{L^2(\Omega)} \\ \alpha\,(u,\widetilde{u})_{L^2(\Omega)} - (p,\widetilde{u})_{L^2(\Omega)} &= 0 \\ (y,\widetilde{p})_{H^1(\Omega)} - (u,\widetilde{p})_{L^2(\Omega)} &= 0 \end{aligned}$$

holds for all $(\widetilde{y},\widetilde{u},\widetilde{p}) \in H_0^1(\Omega) \times L^2(\Omega) \times H_0^1(\Omega)$. The second equation immediately implies that $u = \alpha^{-1} p$, which allows a reduction to the following system (2×2 *formulation of the KKT system*): Find $(y,p) \in X := V \times V := H_0^1(\Omega) \times H_0^1(\Omega)$ such that

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

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

For finding the approximate solution to this problem we use finite element methods (FEM). Therefore we assume to have a sequence of grids partitioning the given domain Ω starting from an initial (coarse) grid on grid level $k = 0$. The grids on grid levels $k = 1,2,\ldots$ are constructed by refinement, i.e., the grid points of level $k-1$ are also grid points of level k. Using standard finite element techniques (Courant elements), we can construct finite dimensional subsets $V_k \subset V$, where the dimension N_k depends on the grid level k. By Galerkin principle, the finite element approximation $(y_k,p_k) \in X_k := V_k \times V_k$ fulfills

$$\begin{aligned} (y_k,\widetilde{y}_k)_{L^2(\Omega)} + (p_k,\widetilde{y}_k)_{H^1(\Omega)} &= (y_D,\widetilde{y}_k)_{L^2(\Omega)} \\ (y_k,\widetilde{p}_k)_{H^1(\Omega)} - \alpha^{-1}(y_k,\widetilde{p}_k)_{L^2(\Omega)} &= 0 \end{aligned} \tag{2}$$

for all $(\widetilde{y}_k,\widetilde{p}_k) \in X_k$.

Assuming to have a (nodal) basis $\Phi_k := (\varphi_{k,i})_{i=1}^{N_k}$ for V_k, we can rewrite the optimality system (2) in matrix-vector notation as follows:

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

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

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

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.

1.2 Multigrid methods and collective point smoothers

In this section we briefly introduce the multigrid framework that we analyze in this paper. Starting from an initial approximation $\underline{x}_k^{(0)}$ one step of the multigrid method with $\nu_1 + \nu_2$ smoothing steps for solving a discretized equation $\mathscr{A}_k \, \underline{x}_k = \underline{f}_k$ on grid level k is given by:

- Apply ν_1 (pre-)smoothing steps

$$\underline{x}_k^{(0,m)} := \underline{x}_k^{(0,m-1)} + \tau \hat{\mathscr{A}}_k^{-1}(\underline{f}_k - \mathscr{A}_k \, \underline{x}_k^{(0,m-1)}) \qquad \text{for } m = 1, \dots, \nu_1 \tag{3}$$

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

- Apply coarse-grid correction
  - Compute the defect and restrict to the coarser grid
  - Solve the problem on the coarser grid
  - Prolongate and add the result

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

$$\underline{x}_k^{(1,-\nu_2)} := \underline{x}_k^{(0,\nu_1)} + I_{k-1}^k \mathscr{A}_{k-1}^{-1} I_k^{k-1}(\underline{f}_k - \mathscr{A}_k \, \underline{x}_k^{(0,\nu_1)}).$$

- Apply ν_2 (post-)smoothing steps

$$\underline{x}_k^{(1,m)} := \underline{x}_k^{(1,m-1)} + \tau \hat{\mathscr{A}}_k^{-1}(\underline{f}_k - \mathscr{A}_k \, \underline{x}_k^{(1,m-1)}) \qquad \text{for } m = -\nu_2 + 1, \dots, 0$$

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

The smoothing steps are applied in order to reduce the high-frequency error, whereas the coarse-grid correction takes care of the low-frequent parts of the overall defect. In practice the problem on grid level $k-1$ is handled by applying one (V-cycle) or two (W-cycle) steps of the proposed method, recursively, and just on the coarse grid level $k=0$ the problem is solved exactly. The convergence of the two-grid method implies the convergence of the W-cycle multigrid method under mild assumptions, so we restrict ourselves to the analysis of the two-grid method only.

The intergrid-transfer operators I_{k-1}^{k} and I_{k}^{k-1} are chosen in a canonical way: we use the canonical embedding for the prolongation operator I_{k-1}^{k} and its adjoint as restriction operator I_{k}^{k-1}.

Next we need to specify the smoothing procedure (3). The preconditioning matrix $\hat{\mathscr{A}}_k$ is typically some easy-to-invert approximation of the matrix $\mathscr{A}_k$. In case of positive definite matrices (which may result from discretizing elliptic scalar BVPs), the preconditioning matrix can be composed in an either additive or multiplicative Schwarz manner based on local problems which live on patches, boxes or, which is the easiest case, just on points. The two main pointwise methods are Jacobi relaxation (additive Schwarz method) and Gauss-Seidel iteration (multiplicative Schwarz method).

We extend these methods to block-systems by combining (again in an additive or multiplicative Schwarz manner) local problems which involve the complete system of BVPs. Based on this idea, we can introduce the *collective Jacobi relaxation* with preconditioning matrix $\hat{\mathscr{A}}_k^{(jac)}$, given by

$$\hat{\mathscr{A}}_k^{(jac)} := \begin{pmatrix} \hat{M}_k^{(jac)} & \hat{K}_k^{(jac)} \\ \hat{K}_k^{(jac)} & -\alpha^{-1}\hat{M}_k^{(jac)} \end{pmatrix} := \begin{pmatrix} \operatorname{diag} M_k & \operatorname{diag} K_k \\ \operatorname{diag} K_k & -\alpha^{-1}\operatorname{diag} M_k \end{pmatrix},$$

where the damping parameter τ is chosen to be in $(0,1)$. Here, $\hat{M}_k^{(jac)}$ and $\hat{K}_k^{(jac)}$ are the preconditioning matrices that are used if Jacobi iteration is applied to linear systems with system matrices M_k and K_k, respectively.

The preconditioning matrix of the *collective Gauss-Seidel iteration* is given by

$$\hat{\mathscr{A}}_k^{(gs)} := \begin{pmatrix} \hat{M}_k^{(gs)} & \hat{K}_k^{(gs)} \\ \hat{K}_k^{(gs)} & -\alpha^{-1}\hat{M}_k^{(gs)} \end{pmatrix},$$

where $\hat{M}_k^{(gs)}$ and $\hat{K}_k^{(gs)}$ are the lower left triangular parts (including the diagonals) of M_k and K_k, respectively. Again with this choice, $\hat{M}_k^{(gs)}$ and $\hat{K}_k^{(gs)}$ are the preconditioning matrices that represent classical Gauss-Seidel iteration for linear systems with system matrices M_k and K_k, respectively. The damping parameter τ is chosen to be 1 in case of collective Gauss-Seidel iteration.

Numerical examples show good behavior of multigrid methods using such iterations as smoothing procedures and have been discussed in, e.g., [1] or [7].

We want to stress that in either case the multiplication with the preconditioning matrix $\hat{\mathscr{A}}_k$ can be realized efficiently if the iteration is implemented analogously to standard Jacobi or Gauss-Seidel iteration. For this in every step the corresponding components of $\underline{y}_k$ and $\underline{p}_k$ are collected to vectors in $\mathbb{R}^2$. Accordingly one has to collect the corresponding entries of the matrices A_k, B_k and C_k to 2×2 matrices. Executing the algorithm then only vectors in $\mathbb{R}^2$ need to be multiplied with 2×2 matrices and 2×2 linear systems need to be solved. For more detailed information how to implement collective iteration schemes see, e.g., [7].

2 Local Fourier analysis

Convergence properties of multigrid methods for the model problem have been investigated in a wide range of papers. In this paper we want to concentrate on an analysis, where symbolic computation can contribute significantly. For the time being, we complete the first step by analyzing the smoothing iteration. As mentioned earlier, we restrict the smoothing analysis to the case of a one dimensional domain Ω. While the proposed numerical method can be applied also to higher dimensions, the analysis of this case as well as the analysis of the full two-grid cycle is ongoing work.

For the analysis of the smoothing procedure introduced in the last section, we define the *iteration matrix* of the smoothing step by

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

which represents the modification of the error effected by the smoothing procedure, i.e.,

$$\underline{x}_k^{(0,m)} - \underline{x}_k^* = S_k(\underline{x}_k^{(0,m-1)} - \underline{x}_k^*),$$

where $\underline{x}_k^* := \mathscr{A}_k^{-1}\underline{f}_k$ denotes the exact solution to the system.

Certainly, if it can be shown that the spectral radius of the iteration matrix or, even better, its norm, are smaller than 1, then this yields convergence of the iterative scheme. At present time, we do not aim at proving convergence, but at showing that it is a *good smoother*. In other words, we want to show that it reduces the high frequency error terms which we do using local Fourier analysis that is introduced next.

2.1 Local Fourier analysis framework

Since for local Fourier analysis the boundary is neglected by assuming periodic boundary conditions that allow to extend a bounded domain Ω to the entire space $\mathbb{R}$, see [2], from now on we assume that $\Omega = \mathbb{R}$. Let us repeat that good conver-

gence and smoothing rates computed using local Fourier analysis for simple cases, typically also indicate good behavior of the analyzed methods in more general cases.

On this domain $\Omega = \mathbb{R}$, we assume to have on each grid level $k = 0, 1, 2, \ldots$ a uniform grid with nodes

$$x_{k,n} := n\, h_k \qquad \text{for } n \in \mathbb{Z},$$

where the uniform grid size is given by $h_k = 2^{-k}$. The functions in V_k are assumed to be continuous on the domain and to be linear between two nodes (Courant elements). This way the discretized function can be specified by prescribing the values on the nodes only.

The first step of local Fourier analysis consists of constructing *Fourier vectors* that diagonalize both mass and stiffness matrix. For every $\theta \in \Theta := [-\pi, \pi)$ and every grid level k, we can define a Fourier vector $\underline{\varphi_k}(\theta) \in \mathbb{R}^{\mathbb{Z}}$ as follows:

$$\underline{\varphi_k(\theta)} := (\mathrm{e}^{\mathrm{i}\theta x_{k,n}/h_k})_{n\in\mathbb{Z}}.$$

The Fourier vectors are the coordinate vectors of the *Fourier functions* with respect to the nodal basis Φ_k. So, the Fourier functions are those functions in V_k whose values on the grid points $x_{k,n}$ equal to the corresponding entries of the Fourier vector. Thus the Fourier function are the piecewise linear functions given by

$$\varphi_k(\theta)(x) = t\, \mathrm{e}^{\mathrm{i}\theta x_{k,n}/h_k} + (1-t)\, \mathrm{e}^{\mathrm{i}\theta x_{k,n+1}/h_k},$$

where n and $t \in [0, 1)$ are chosen such that $x = t\, x_{k,n} + (1-t)\, x_{k,n+1}$.

It is easy to see that every vector in $\mathbb{R}^{\mathbb{Z}}$ can be expressed as linear combination of countable infinitely many Fourier vectors. Therefore also every function in V_k can be expressed as linear combination of countable infinitely many Fourier functions. In case of a bounded domain, just finitely many Fourier vectors or functions would be necessary. Nonetheless for the analysis, all $\theta \in \Theta = [-\pi, \pi)$ are considered.

2.2 Operators in local Fourier space

The Fourier vectors defined in the preceding subsection diagonalize the blocks of our system matrix. Since we consider grids with uniform mesh-size h_k, the (infinitely large) mass matrix M_k and the (infinitely large) stiffness matrix K_k can be computed explicitly:

$$M_k = \frac{h_k}{6}\begin{pmatrix} \ddots & \ddots & \ddots & & \\ & 1 & 4 & 1 & \\ & & 1 & 4 & 1 \\ & & & \ddots & \ddots & \ddots \end{pmatrix} \quad \text{and} \quad K_k = \frac{1}{h_k}\begin{pmatrix} \ddots & \ddots & \ddots & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & \ddots & \ddots & \ddots \end{pmatrix}.$$

It is easy to see that the multiplication of one of these matrices with the vector $\underline{\varphi_k(\theta)}$ equals the multiplication of this vector with the symbol of the matrix, where the symbols are given by:

$$M_k\ \underline{\varphi_k(\theta)} = \underbrace{\frac{(4+e^{i\theta}+e^{-i\theta})h_k}{6}}_{\overline{M_k}(\theta):=}\ \underline{\varphi_k(\theta)} \quad \text{and} \quad K_k\ \underline{\varphi_k(\theta)} = \underbrace{\frac{2-e^{i\theta}-e^{-i\theta}}{h_k}}_{\overline{K_k}(\theta):=}\ \underline{\varphi_k(\theta)}.$$

Thus indeed the Fourier vectors are the eigenvectors with eigenvalues $\overline{M_k}(\theta)$ and $\overline{K_k(\theta)}$, respectively. The analogous holds for the preconditioning matrices. For the collective Jacobi relaxation, the preconditioning matrix itself is a diagonal matrix, therefore

$$\hat{M}_k^{(jac)}\ \underline{\varphi_k(\theta)} = \underbrace{\frac{2h_k}{3}}_{\overline{\hat{M}_k^{(jac)}}(\theta):=}\ \underline{\varphi_k(\theta)} \quad \text{and} \quad \hat{K}_k^{(jac)}\ \underline{\varphi_k(\theta)} = \underbrace{\frac{2}{h_k}}_{\overline{\hat{K}_k^{(jac)}}(\theta):=}\ \underline{\varphi_k(\theta)}$$

holds. As the block matrices $\mathscr{A}_k$ and $\hat{\mathscr{A}}_k$ are built from such matrices, we can conclude that for all $\theta \in \Theta$,

$$\operatorname{span}\left\{ \begin{pmatrix} \underline{\varphi_k(\theta)} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \underline{\varphi_k(\theta)} \end{pmatrix} \right\}$$

is invariant under the action of those block-matrices. Hence it suffices to consider only the symbol of the block matrix $\mathscr{A}_k$, given by

$$\overline{\mathscr{A}_k}(\theta) := \begin{pmatrix} \overline{M_k}(\theta) & \overline{K_k}(\theta) \\ \overline{K_k}(\theta) & -\alpha^{-1}\overline{M_k}(\theta) \end{pmatrix},$$

and the symbol $\overline{\hat{\mathscr{A}}_k^{(jac)}}(\theta)$ defined analogously.

As mentioned earlier, the smoothing iteration shall reduce the high-frequent parts of the error. To measure this phenomenon, we introduce the smoothing rate

$$q(\tau) := \sup_{\theta\in\Theta^{(high)}} \sup_{h_k>0} \sup_{\alpha>0} \sigma(\theta, h_k, \alpha, \tau), \tag{4}$$

where $\Theta^{(high)} := [-\pi,\pi)\backslash[-\frac{\pi}{2},\frac{\pi}{2})$ is the set of high frequencies and σ is defined as the spectral radius of the Fourier-transformed smoothing operator given by

$$\sigma(\theta, h_k, \alpha, \tau) := \rho\Big(\underbrace{I - \tau\left(\overline{\hat{\mathscr{A}}}(\theta)\right)^{-1}\overline{\mathscr{A}}(\theta)}_{\overline{S}_k(\theta):=}\Big). \tag{5}$$

In (4) the supremum is not only taken over all $\theta \in \Theta^{(high)}$, but also over all grid sizes $h_k > 0$ and choices of the parameter $\alpha > 0$. Therefore, we compute an upper bound

for the smoothing rate which is independent of h_k (which allows to show optimal convergence) and the parameter α (which allows to show robust convergence). For obvious reasons, the supremum is not taken with respect to the damping parameter τ, but is adjusted within the method such that the smoothing rate is optimal for Jacobi relaxation.

In principle it would be necessary to analyze the norm of the iteration matrix in (5) rather than analyzing the spectral radius. The spectral radius, however, equals the infimum over all matrix norms, which implies that for every $\varepsilon > 0$ there is a matrix norm such that

$$\|\overline{S}_k(\theta)\| \leq (1+\varepsilon)\,\rho(\overline{S}_k(\theta)),$$

see [6]. For the model problem and both proposed smoothing procedures (collective Jacobi relaxation and collective Gauss-Seidel iteration) straight-forward computations show that the spectral radius of the symbol of the smoothing operator $\rho(\overline{S}_k(\theta))$ is *equal* to its norm $\|\overline{S}_k(\theta)\|_{\hat{X}}$, if the matrix-norm is chosen as

$$\|\mathscr{M}\|_{\hat{X}} := \left\| \begin{pmatrix} \alpha^{1/2} & \\ & 1 \end{pmatrix} \mathscr{M} \begin{pmatrix} \alpha^{1/2} & \\ & 1 \end{pmatrix}^{-1} \right\|_{\ell^2}, \tag{6}$$

where $\|\cdot\|_{\ell^2}$ denotes the Euclidean norm. Observe that the scaling of the state y and the adjoined state p to each other in this norm equals to the scaling in the norm $\|\cdot\|_X$ in classical theory that can be found in [8].

An equivalent formulation for the definition of the smoothing rate (4) using quantifiers is: Determine λ such that

$$\forall\, \theta \in \Theta^{(high)}\, \forall\, h_k > 0\, \forall\, \alpha > 0 : \sigma^2(\theta, h_k, \alpha, \tau) \leq \lambda \tag{7}$$

holds. Then for every $\tau \in (0,1)$ the value of $q(\tau)$ is the smallest such λ.

The computation of $\sigma(\theta, h_k, \alpha, \tau)$ is straight forward, but the computation of $q(\tau)$ is non-trivial. This is where symbolic computation enters our analysis. In order to determine q (that is either a polynomial in τ or a constant) we invoke *quantifier elimination* using *cylindrical algebraic decomposition* (CAD) that is introduced in the next section. Note that for both preconditioners under consideration, (7) is a quantified formula on trigonometric polynomials (after clearing denominators). For the case of a collective Jacobi relaxation σ is given by

$$\sigma^2(\theta, h_k, \alpha, \tau) = \frac{h_k^4((\cos\theta + 2)\tau - 2)^2 + 36\alpha((\cos\theta - 1)\tau + 1)^2}{4\left(h_k^4 + 9\alpha\right)}.$$

CAD, as we detail in the next section, accepts as input only polynomial (or more general rational) inequalities over the reals. This is a complication that is easily resolved by replacing $\cos\theta$ by a real variable $c \in [-1,1]$ and, if necessary, $\sin\theta$ by a real variable $s \in [-1,1]$ together with Pythagoras' identity $s^2 + c^2 = 1$.

3 Quantifier elimination using cylindrical algebraic decomposition

So far we have reformulated the task of determining the smoothing rate for our multigrid methods to the problem of resolving a quantified polynomial inequality. That is, the given statement is of the form

$$Q_1\, x_1 \dots Q_n\, x_n : A(x_1, \dots, x_n, y_1, \dots, y_m),$$

where Q_i denote quantifiers (either $\forall$ or $\exists$) and $A(x_1, \dots, x_n, y_1, \dots, y_m)$ is a boolean combination of *polynomial* inequalities. The problem of finding an equivalent, quantifier free formula $B(y_1, \dots, y_m)$ consisting of a boolean combination of polynomial inequalities depending only on the free variables is called quantifier elimination. The first algorithm to solve this problem over the reals was given by A. Tarski [11] in the early 1950s. His method, however, was practically not efficient. Nowadays modern implementations [3, 9, 10] of G. Collins' cylindrical algebraic decomposition [4] make it possible to carry out nontrivial computations in a reasonable amount of time.

A simple example is given by: Determine a bound $B = B(z)$ for $0 < z < 1$ such that

$$\forall\, 0 < x < 1\, \forall\, 0 < y < 1 : \frac{x}{y+z} + \frac{y}{x+z} \leq B.$$

A CAD-computation quickly yields that $B(z) \geq \frac{1}{z}$. In cases where no free variables appear in the input, B is one of the logical constants True or False. Applied to a quantifier free formula the result of a CAD-computation is an equivalent formula that is normalized in a certain sense.

The formula for σ as stated above is a rational function in the given indeterminates. Adding the necessary constraints on the denominators (which is commonly handled internally by the implementations), this is still a valid input for a CAD-computation.

A major issue is the runtime complexity of CAD that depends heavily on the input parameters such as number of polynomial inequalities, polynomial degrees and 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. As we will see below, already for the one dimensional analysis suitable substitutions of the variables are applied in order to speed up the computations. These substitutions aim at reducing the number of variables on the one hand and lowering the polynomial degrees on the other hand.

For the forthcoming analysis of two (or even three) dimensional case, further simplifications will be necessary, since even though termination of the algorithm is proven, the actual computations might last longer than the expected life-time of the authors. Although it might seem a high price to pay, the gain is an *optimal* bound for the given formula that is determined by a *proving* procedure that is not approximate in any way.

4 Computing the smoothing rate

Now we are in the position to state the main results of this paper, the smoothing rates for collective Jacobi relaxation and collective Gauss-Seidel iteration.

4.1 Smoothing property - collective Jacobi relaxation

In the smoothing step we are concerned with the high-frequent parts of the error. Consequently, if we replace $\cos\theta$ by a real variable c, then the condition $\theta \in \Theta^{(high)}$ translates to $-1 \le c \le 0$.

With this substitution in the case of a collective Jacobi relaxation σ is given by $\sigma(\theta, h_k, \alpha, \tau) = \widetilde{\sigma}(\cos\theta, h_k, \alpha, \tau)$, where

$$\widetilde{\sigma}^2(c, h_k, \alpha, \tau) := \frac{h_k^4((c+2)\tau-2)^2 + 36\alpha((c-1)\tau+1)^2}{4\left(h_k^4+9\alpha\right)}.$$

With this rewriting the definition (7) has become a purely polynomial inequality and can invoke CAD to determine $q(\tau)$. For this purpose we used the CAD-implementation in *Mathematica*. The subscript 2 for σ below indicates that we are dealing with the square of the actual expression.

In[1]= $\sigma_2 = \dfrac{h^4((c+2)\tau-2)^2 + 36\alpha((c-1)\tau+1)^2}{4\left(h^4+9\alpha\right)}$;

In[2]= Resolve[ForAll[$c, -1 \le c \le 0$, ForAll[$h, h > 0$, ForAll[$\alpha, \alpha > 0, \sigma_2 \le \lambda$]]], $\{\tau, \lambda\}$, Reals]

Out[2]= $\left(\tau \le 0 \&\& \lambda \ge 4\tau^2 - 4\tau + 1\right) \| \left(0 < \tau \le \dfrac{4}{5} \&\& \lambda \ge \dfrac{1}{4}\left(\tau^2 - 4\tau + 4\right)\right) \| \left(\tau > \dfrac{4}{5} \&\& \lambda \ge 4\tau^2 - 4\tau + 1\right)$

The computation takes about one second and the result is a quantifier-free formula equivalent to the quantified formula given in (7). Note also that this is again a statement formulated in terms of polynomial inequalities. It is normalized in the sense that the parameter τ is assumed to be on the bottom level, which is indicated by the order of variables within the "Resolve"-command. Thus the output is sorted in a way that the conditions on τ are inequalities comparing to (algebraic) numbers, whereas the conditions on λ on the next higher level are formulated in terms of τ.

So for every τ the function value $q^2(\tau)$ is the smallest λ fulfilling Out[2]. For instance, if we plug $\tau = \frac{1}{2}$ into Out[2], the formula reduces to:

$$\text{False} \vee \left(\text{True} \wedge \lambda \ge \frac{3}{4}\right) \vee \text{False}.$$

As $q^2\left(\frac{1}{2}\right)$ is the smallest λ fulfilling the inequality, we have $q^2\left(\frac{1}{2}\right) = \frac{3}{4}$. Guided by this example we read off the general form for $q^2(\tau)$ which is a piecewise quadratic

function given by

$$q^2(\tau) = \begin{cases} 4\tau^2 - 4\tau + 1 & \text{for} \quad \tau \leq 0 \\ \frac{1}{4}\left(\tau^2 - 4\tau + 4\right) & \text{for} \quad 0 < \tau \leq \frac{4}{5} \\ 4\tau^2 - 4\tau + 1 & \text{for} \quad \frac{4}{5} < \tau \end{cases}. \tag{8}$$

Summarizing we have determined the supremum in (4) and therefore the smoothing rate $q(\tau)$. If we take the square root of (8) and restrict ourselves to the relevant range $\tau \in [0,1]$, we obtain the smoothing rates for the collective Jacobi relaxation:

$$q(\tau) = \begin{cases} \frac{1}{2}(2-\tau) & \text{for} \quad 0 \leq \tau \leq \frac{4}{5} \\ 2\tau - 1 & \text{for} \quad \frac{4}{5} < \tau \leq 1 \end{cases}.$$

Since our method gives an equivalent reformulation, we know that these bounds on the smoothing rate are sharp. The graph of the function q can be seen in figure 1. From this we see that $q(\tau)$ takes its minimum for $\tau = \frac{4}{5}$ with value $q\left(\frac{4}{5}\right) = \frac{3}{5}$. For the canonical choice $\tau = \frac{1}{2}$, we obtain $q\left(\frac{1}{2}\right) = \frac{3}{4}$.

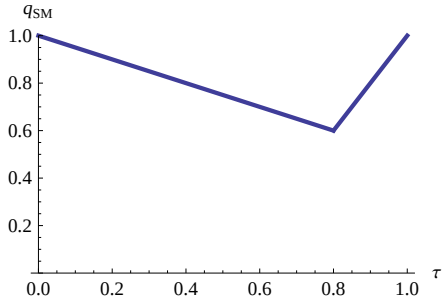

Fig. 1 Smoothing factor depending on damping parameter τ

4.2 Smoothing property - collective Gauss-Seidel iteration

In this subsection we carry over the smoothing analysis which we have done for the collective Jacobi relaxation to the collective Gauss-Seidel iteration. Again we can determine the symbol of the involved preconditioning matrices:

$$\hat{M}_k^{(gs)} \underline{\varphi_k(\theta)} = \underbrace{\frac{(4+e^{-i\theta})h_k}{6}}_{\overline{\hat{M}_k^{(gs)}}(\theta):=} \underline{\varphi_k(\theta)} \quad \text{and} \quad \hat{K}_k^{(gs)} \underline{\varphi_k(\theta)} = \underbrace{\frac{2-e^{-i\theta}}{h_k}}_{\overline{\hat{K}_k^{(gs)}}(\theta):=} \underline{\varphi_k(\theta)}.$$

The procedure for determining $\sigma(\theta,h,\alpha)$ (which now is independent of τ) is completely analogous to the previous case. Again, by our choice of matrix norm $\|\cdot\|_{\hat{X}}$, the same value for σ is obtained no matter whether we consider the spectral radius or the norm.

In order to have a purely polynomial input for the CAD computations, we simplify the formula for σ: the occurrences of $\cos\theta$ and $\sin\theta$ are replaced by c and s, respectively. Moreover we expand numerator and denominator and replace all occurrences of s^2 by $1-c^2$ thus arriving at numerator and denominator being linear in s. After these simple rewriting steps we obtain $\sigma(\theta,h_k,\alpha)=\widetilde{\sigma}(\sin\theta,\cos\theta,h_k,\alpha)$, where

$$\widetilde{\sigma}^2(s,c,h_k,\alpha) := \frac{\left(h_k^4+36\alpha\right)\left((17+8c)h_k^4+72h_k^2\alpha^{1/2}|s|+36(5-4c)\alpha\right)}{(17+8c)^2h_k^8+72(40c^2-28c+13)h_k^4\alpha+1296(5-4c)^2\alpha^2}.$$

The smoothing rate q is again the supremum over all high frequencies, grid sizes and choices of the parameter α and is given by

$$q^2 = \sup_{(s,c)\in D}\ \sup_{h_k>0}\ \sup_{\alpha>0} \widetilde{\sigma}^2(s,c,h_k,\alpha),$$

where $D := \{(s,c)\in\mathbb{R}^2\ :\ s^2+c^2=1,\ c\le 0\}$.

Note that in this definition above still a term $\alpha^{1/2}$ occurs. Before we can invoke CAD-computations, we have to rewrite $\widetilde{\sigma}$ as rational function. A first simplification is that as $\widetilde{\sigma}$ does not depend on the sign of s, we can restrict ourselves to assuming only non-negative values and thus replace $|s|$ by $s\ge 0$. To eliminate also $\sqrt{\alpha}$ in the numerator, we replace α by $\widetilde{\alpha}^2$, where $\widetilde{\alpha}>0$. Having completed these rewritings the final formula for q reads as

$$q^2 = \sup_{(s,c)\in\widetilde{D}}\ \sup_{h_k>0}\ \sup_{\widetilde{\alpha}>0} \frac{\left(h_k^4+36\widetilde{\alpha}^2\right)\left((17+8c)h_k^4+72h_k^2\widetilde{\alpha}s+36(5-4c)\widetilde{\alpha}^2\right)}{(17+8c)^2h_k^8+72(40c^2-28c+13)h_k^4\widetilde{\alpha}^2+1296(5-4c)^2\widetilde{\alpha}^4},$$

where $\widetilde{D} := \{(s,c)\in\mathbb{R}^2\ :\ s^2+c^2=1,\ c\le 0,\ s\ge 0\}$. We can again rewrite the supremum as quantified expression where the quantifiers can be eliminated with the help of a CAD computation. With *Mathematica's* quantifier elimination algorithm, we obtain the smoothing rate for the collective Gauss-Seidel iteration after about twenty minutes:

$$q = \tfrac{1}{7}(3+\sqrt{2}) \approx 0.63.$$

Even though twenty minutes are not a very long time to wait for a result that needs to be obtained only once, it still seems too long for such a simple formula. We can speed up the calculation significantly by reducing both the number of variables and the degrees of the polynomials by introducing a new variable $\eta := h_k^2/\widetilde{\alpha} > 0$. This substitution reduces the formula for q to

$$q^2 = \sup_{(s,c)\in\widetilde{D}}\ \sup_{\eta>0} \frac{\left(\eta^2+36\right)\left((17+8c)\eta^2+72\eta s+36(5-4c)\right)}{(17+8c)^2\eta^4+72(40c^2-28c+13)\eta^2+1296(5-4c)^2}.$$

Based on this representation *Mathematica's* quantifier elimination algorithm is able to derive q within about twenty seconds.

5 Concluding remarks

In this paper we have shown a strategy to compute the smoothing rate for a multigrid method using collective Jacobi relaxation or Gauss-Seidel iteration by means of symbolic computation in an entirely automatic manner. The proposed strategy strongly relies on the fact that local Fourier analysis is a systematic machinery which is applied to the problem and the given numerical method. Typically this approach leads to determining the supremum of an explicitly given term.

On the one hand, the smoothing rates we obtained this way may be viewed as an interesting result on their own. On the other hand, these rates will also enter a full two- or multigrid analysis which again can be done using local Fourier analysis.

Also for the full analysis, or the extension to higher dimensional cases, Fourier analysis leads to an expression that in the first step is a rational function in the mesh size h_k, the regularization parameter α, the damping parameter τ, and trigonometric expressions of the frequencies θ. This is in particular the case for the model problem described in this paper for the above mentioned generalizations.

Theoretical results guarantee that also these problems can be solved with the methods applied in this work. To obtain the full results in reasonable time, it is necessary to apply proper strategies to reduce the complexity of the problems in the formulation of the input which is ongoing work.

References

1. Borzi, A., Kunisch, K. and Kwak, D. Y. Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system. *SIAM J. Control Optim*, 41(5):1477 – 1497, 2003.
2. Brandt, A. Rigorous quantitative analysis of multigrid, i: Constant coefficients two-level cycle with l2-norm. *SIAM Journal on Numerical Analysis*, 31(6):1695 – 1730, 1994.
3. Brown, C.W. QEPCAD B – a program for computing with semi-algebraic sets. *Sigsam Bulletin*, 37(4):97–108, 2003.
4. Collins, G.E. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In *Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975)*, pages 134–183. Lecture Notes in Comput. Sci., Vol. 33. Springer, Berlin, 1975.
5. Hackbusch, W. *Multi-Grid Methods and Applications*. Springer, Berlin, 1985.
6. Holmes, R. B. A formula for the spectral radius of an operator. *The American Mathematical Monthly*, 75(2):163–166, 1968.
7. Lass, O., Vallejos, M., Borzi, A., and Douglas, C. C. Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems. *Computing*, 84(1-2):27–48, 2009.
8. Schöberl, J., Simon, R., and Zulehner, W. A Robust Multigrid Method for Elliptic Optimal Control Problems. *NuMa-Report*, 1, 2010.

9. Seidl, A., and Sturm, T. A generic projection operator for partial cylindrical algebraic decomposition. In *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation*, pages 240–247 (electronic), New York, 2003. ACM.
10. Strzeboński, A. Solving systems of strict polynomial inequalities. *J. Symbolic Comput.*, 29(3):471–480, 2000.
11. Tarski, A. *A decision method for elementary algebra and geometry*. University of California Press, Berkeley and Los Angeles, Calif., 1951. 2nd ed.
12. Trottenberg, U., Oosterlee, C., and Schüller, A. *Multigrid*. Academic Press, London, 2001.
13. Wienands, R., and Joppich, W. *Practical Fourier analysis for multigrid methods*. Chapman & Hall/CRC, 2005.

Technical Reports of the Doctoral Program "Computational Mathematics"

2010

2010-01 S. Radu, J. Sellers: *Parity Results for Broken k-diamond Partitions and (2k+1)-cores* March 2010. Eds.: P. Paule, V. Pillwein

2010-02 P.G. Gruber: *Adaptive Strategies for High Order FEM in Elastoplasticity* March 2010. Eds.: U. Langer, V. Pillwein

2010-03 Y. Huang, L.X.Châu Ngô: *Rational General Solutions of High Order Non-autonomous ODEs* June 2010. Eds.: F. Winkler, P. Paule

2010-04 S. Beuchler, V. Pillwein, S. Zaglmayr: *Sparsity optimized high order finite element functions for H(div) on simplices* September 2010. Eds.: U. Langer, P. Paule

2010-05 C. Hofreither, U. Langer, C. Pechstein: *Analysis of a non-standard finite element method based on boundary integral operators* September 2010. Eds.: B. Jüttler, J. Schicho

2010-06 M. Hodorog, J. Schicho: *A symbolic-numeric algorithm for genus computation* September 2010. Eds.: B. Jüttler, R. Ramlau

2010-07 M. Hodorog, J. Schicho: *Computational geometry and combinatorial algorithms for the genus computation problem* September 2010. Eds.: B. Jüttler, R. Ramlau

2010-08 C. Koukouvinos, V. Pillwein, D.E. Simos, Z. Zafeirakopoulos: *A Note on the Average Complexity Analysis of the Computation of Periodic and Aperiodic Ternary Complementary Pairs* October 2010. Eds.: P. Paule, J. Schicho

2010-09 V. Pillwein, S. Takacs: *Computing smoothing rates of collective point smoothers for optimal control problems using symbolic computation* October 2010. Eds.: U. Langer, P. Paule

The complete list since 2009 can be found at
`https://www.dk-compmath.jku.at/publications/`

Doctoral Program
"Computational Mathematics"

Director:

Prof. Dr. Peter Paule
Research Institute for Symbolic Computation

Deputy Director:

Prof. Dr. Bert Jüttler
Institute of Applied Geometry

Address:

Johannes Kepler University Linz
Doctoral Program "Computational Mathematics"
Altenbergerstr. 69
A-4040 Linz
Austria
Tel.: ++43 732-2468-7174

E-Mail:

office@dk-compmath.jku.at

Homepage:

http://www.dk-compmath.jku.at

Submissions to the DK-Report Series are sent to two members of the Editorial Board who communicate their decision to the Managing Editor.