Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Chapter 5

Multigrid Methods

5.1 Introduction

From the point of view of the multigrid methodology, we can roughly distinguish between the direct multigrid approach where the optimization problem is implemented within the hierarchy of grid levels and the use of multigrid schemes as inner solvers within an outer optimization loop. The focus of this chapter is on multigrid techniques for optimization, and therefore we do not attempt to discuss other multilevel approaches to optimization such as domain decomposition methods [186], space-mapping and surrogate model techniques [119, 192], multigrid schemes designed to cope with the ill-posedness of inverse problems [42, 214, 221, 251], and other multigrid optimization schemes [256].

Before we address multigrid strategies for optimization problems, we provide an introduction to general multigrid techniques and related theoretical tools, with a focus on the optimization properties of the multigrid components.

5.2 Multigrid Methods for Linear Problems

We start introducing the basic components of a multigrid algorithm and discuss two standard iterative techniques: the Jacobi and the Gauss–Seidel schemes. These two classical iterative methods are characterized by poor convergence rates. However, for errors whose length scales are comparable to the mesh size, they provide rapid damping, leaving behind smooth, longer wavelength errors. These smooth components are responsible for the slow global convergence. A multigrid algorithm, employing grids of different mesh sizes, allows us to solve all wavelength components and provides rapid convergence rates. The multigrid strategy combines two complementary schemes. The high-frequency components of the error are reduced applying iterative methods like Jacobi or Gauss–Seidel schemes. For this reason these methods are called smoothers. On the other hand, low-frequency error components are effectively reduced by a coarse-grid correction scheme. Because the action of a smoothing iteration leaves only smooth error components, it is possible to represent these components as the solution of an appropriate coarser system. Once this coarser problem is solved, its solution is interpolated back to the fine grid to correct the fine-grid approximation for the low-frequency errors.

5.2.1 Iterative Methods and the Smoothing Property

Consider a large sparse linear system of equations $Au = f$, where A is a symmetric positive definite $n \times n$ matrix and u and f are n-dimensional vectors. Iterative methods for solving this problem are formulated as follows

$$u^{(\nu+1)} = Mu^{(\nu)} + Nf, \tag{5.1}$$

where M and N have to be constructed in such a way that given an arbitrary initial vector $u^{(0)}$, the sequence $u^{(\nu)}$, $\nu = 0, 1, \ldots$, converges to the solution $u = A^{-1}f$. Define the solution error at the sweep ν as $e^{(\nu)} = u - u^{(\nu)}$; then the iteration (5.1) is equivalent to $e^{(\nu+1)} = Me^{(\nu)}$. M is called the iteration matrix. We have the following convergence criterion based on the spectral radius $r(M)$ of the matrix; see, e.g., [367].

Theorem 5.1. *The method* (5.1) *converges for any initial iterate* $u^{(0)}$ *if and only if* $r(M) < 1$.

A general framework to define iterative schemes of type (5.1) is based on the concept of splitting [353]. Assume the splitting $A = B - C$, where B is nonsingular. By setting $Bu^{(\nu+1)} - Cu^{(\nu)} = f$ and solving with respect to $u^{(\nu+1)}$, one obtains

$$u^{(\nu+1)} = B^{-1}Cu^{(\nu)} + B^{-1}f.$$

Thus $M = B^{-1}C$ and $N = B^{-1}$. Typically, one considers the regular splitting $A = D - L - U$, where $D = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$ denotes the diagonal part of the matrix A, and $-L$ and $-U$ are the strictly lower and upper parts of A, respectively. Based on this splitting many choices for B and C are possible, leading to different iterative schemes. For example, the choice $B = \frac{1}{\omega}D$ and $C = \frac{1}{\omega}[(1-\omega)D + \omega(L+U)]$, with $0 < \omega \leq 1$, leads to the damped Jacobi iteration

$$u^{(\nu+1)} = (I - \omega D^{-1}A)u^{(\nu)} + \omega D^{-1}f. \tag{5.2}$$

Choosing $B = D - L$ and $C = U$, one obtains the Gauss–Seidel iteration

$$u^{(\nu+1)} = (D-L)^{-1}Uu^{(\nu)} + (D-L)^{-1}f. \tag{5.3}$$

Later on we denote the iteration matrices corresponding to (5.2) and (5.3) with $M_J(\omega)$ and M_{GS}, respectively.

It is useful to define and analyze the smoothing property of these iterations by introducing a simple model problem. Consider the finite difference approximation of a one-dimensional Dirichlet boundary value problem. We have

$$\begin{cases} -u''(x) = f(x) & \text{in} \quad \Omega = (0,1), \\ u(x) = g(x) & \text{on} \quad \{0,1\}. \end{cases} \tag{5.4}$$

Let Ω be represented by a grid Ω_h with grid size $h = \frac{1}{n+1}$ and grid points $x_j = jh$, $j = 0, 1, \ldots, n+1$. A discretization scheme for the second derivative at x_j is $h^{-2}[u(x_{j-1}) - 2u(x_j) + u(x_{j+1})] = u''(x_j) + O(h^2)$. Set $f_j^h = f(jh)$, and $u_j^h = u(jh)$. We obtain the following tridiagonal system of n equations

$$\begin{aligned} 2u_1^h - u_2^h &= h^2 f_1^h + g(0), \\ -u_{j-1}^h + 2u_j^h - u_{j+1}^h &= h^2 f_j^h, \qquad j = 2, \ldots, n-1, \\ -u_{n-1}^h + 2u_n^h &= h^2 f_n^h + g(1). \end{aligned} \tag{5.5}$$

Let us denote (5.5) (with all terms divided by h^2) by $A_h u_h = f_h$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

We discuss the solution to this problem by means of the damped Jacobi iteration with iteration matrix $M_J(\omega) = I - \omega D_h^{-1} A_h$. Consider the eigenvalue problem $M_J(\omega)v^k = \mu_k v^k$. The eigenvectors of $M_J(\omega)$ (and equivalently of A_h) are given by

$$v^k = \sqrt{2h}\,(\sin(k\pi hj))_{j=1,n}, \qquad k = 1,\ldots,n. \tag{5.6}$$

The eigenvalues of A_h are $\lambda_k = 4\sin^2(k\pi h/2)/h^2$ and the corresponding eigenvalues of $M_J(\omega)$ are

$$\mu_k(\omega) = 1 - \omega(1 - \cos(k\pi h)), \qquad k = 1,\ldots,n. \tag{5.7}$$

We have that $r(M_J(\omega)) < 1$ for $0 < \omega \le 1$, guaranteeing convergence. In particular, for the Jacobi iteration with $\omega = 1$ we have $r(M_J(1)) = 1 - \frac{1}{2}\pi h^2 + O(h^4)$, showing how the convergence of the Jacobi iteration deteriorates (i.e., r tends to 1) as $h \to 0$.

The purpose of an iteration in a multigrid algorithm is primarily to be a smoothing operator. In order to characterize this property, we need to distinguish between low- and high-frequency eigenvectors. We define

- *low-frequency* (LF) components: v^k with $1 \le k < \frac{n}{2}$;
- *high-frequency* (HF) components: v^k with $\frac{n}{2} \le k \le n$.

We now define the smoothing factor μ as the worst factor by which the amplitudes of HF components are damped per iteration. In the case of the Jacobi iteration we have

$$\begin{aligned}\mu &= \max\left\{|\mu_k|, \frac{n}{2} \le k \le n\right\} = \max\{1-\omega, \quad |1-\omega(1-\cos(\pi)|\} \\ &\le \max\{1-\omega, |1-2\omega|\}.\end{aligned}$$

Using this result we find that the optimal (smallest) smoothing factor $\mu = 1/3$ is obtained by choosing $\omega^* = 2/3$. This means that using $M_J(\omega^*)$ the HF error components are reduced by at least a factor of one-third after any sweep and this factor does not depend on the mesh size. Therefore if we use the expansion $e^{(\nu)} = \sum_k e_k^{(\nu)} v^k$, we have that a few sweeps of (5.2) give $|e_k^{(\nu)}| \ll |e_k^{(0)}|$ for HF error components. For this reason, although the global error decreases slowly by iteration, it is smoothed very quickly.

Most often, instead of a Jacobi method other iterations are used that suppress the HF components of the error more efficiently. This is the case of the Gauss–Seidel iteration (5.3). The smoothing property of this scheme is conveniently analyzed by using local Fourier analysis introduced by Brandt [83, 85]. This is an effective tool for analyzing the multigrid process even though it is based on certain idealized assumptions and simplifications: Boundary conditions are neglected, and the problem is considered on infinite grids $G^h = \{jh,\, j \in \mathbb{Z}\}$ and represented in terms of the (continuous) Fourier functions $\varphi(\theta, x) = e^{i\theta x/h}$ with $\theta \in (-\pi, \pi]$. Equivalently, we could refer to the Fourier functions on a unit interval with periodic boundary conditions. Notice that on G^h only the components $e^{i\theta x/h}$ with $\theta \in (-\pi, \pi]$ are visible; i.e., there is no other component with frequency $\theta_0 \in (-\pi, \pi]$ with $|\theta_0| < \theta$ such that $e^{i\theta_0 x/h} = e^{i\theta x/h}$, $x \in G^h$. The notion of LF and HF components on the grid G^h is related to a coarser grid denoted by G^H. In this way $e^{i\theta x/h}$ on G^h is said to be an HF component, with respect to the coarse grid G^H, if its restriction (projection) to G^H is not visible there. If $H = 2h$, then the high frequencies are those with $\frac{\pi}{2} \le |\theta| \le \pi$ and we have $e^{i\theta x/h} = e^{i(2\theta)x/H}$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

In this framework, in order to analyze a given iteration we represent solution errors in terms of their θ components $e^{(\nu)} = \sum_\theta \mathbb{E}_\theta^{(\nu)} e^{i\theta x/h}$ and $e^{(\nu+1)} = \sum_\theta \mathbb{E}_\theta^{(\nu+1)} e^{i\theta x/h}$ (with formal summation on θ), where $\mathbb{E}_\theta^{(\nu)}$ and $\mathbb{E}_\theta^{(\nu+1)}$ denote the error amplitudes of the θ component before and after smoothing, respectively. The action of the iteration matrix M is $e^{(\nu+1)} = M\, e^{(\nu)}$. In the Fourier space this action is represented by $\mathbb{E}_\theta^{(\nu+1)} = \hat{M}(\theta)\mathbb{E}_\theta^{(\nu)}$, and $\hat{M}(\theta)$ is the so-called Fourier symbol of M; see [340].

In the local Fourier analysis framework, the smoothing factor is then defined by

$$\mu = \max\left\{ \left| \frac{\mathbb{E}_\theta^{(\nu+1)}}{\mathbb{E}_\theta^{(\nu)}} \right|, \quad \frac{\pi}{2} \le |\theta| \le \pi \right\} = \max\left\{ |\hat{M}(\theta)|, \quad \frac{\pi}{2} \le |\theta| \le \pi \right\}. \tag{5.8}$$

Later, we consider the entire frequency domain spanned by the two sets of frequencies $\theta \in [-\pi/2, \pi/2)$ and $\overline{\theta} = \theta - sign(\theta)\pi$. Here θ represents LF components while $\overline{\theta}$ represents the HF components. This choice results in a representation with respect to two harmonics $e^{i\theta x/h}$ and $e^{i\overline{\theta} x/h}$. In this framework, a way to characterize the smoothing property of the smoothing operator M is to consider the action of M on both sets of frequencies,

$$\widehat{M}(\theta) = \begin{bmatrix} \hat{M}(\theta) & 0 \\ 0 & \hat{M}(\overline{\theta}) \end{bmatrix},$$

and to assume an ideal coarse-grid correction which annihilates the LF error components and leaves the HF error components unchanged. That is, one defines the projection operator $\widehat{Q}$ as follows

$$\widehat{Q}(\theta) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

In this framework the smoothing property of M is defined as follows

$$\mu = \max\{r(\widehat{Q}(\theta)\,\widehat{M}(\theta)) : \theta \in [-\pi/2, \pi/2)\}, \tag{5.9}$$

where r is the spectral radius.

For illustration, consider the Gauss–Seidel scheme applied to our discretized model problem. A smoothing sweep starting with an initial approximation $u^{(\nu)}$ produces a new approximation $u^{(\nu+1)}$ such that the corresponding error satisfies

$$B_h e^{(\nu+1)}(x) - C_h e^{(\nu)}(x) = 0, \qquad x \in G^h, \tag{5.10}$$

where $B_h = D_h - L_h$ and $C_h = U_h$. For a given θ, (5.10) at $x = jh$ becomes

$$\sum_\theta \left[(2 - e^{-i\theta})\mathbb{E}_\theta^{(\nu+1)} - e^{i\theta}\mathbb{E}_\theta^{(\nu)} \right] e^{i\theta j} = 0,$$

which must hold for all j; therefore we obtain $\hat{M}(\theta) = e^{i\theta}/(2 - e^{-i\theta})$. Hence we have

$$\mu = \max\left\{ \left| \frac{\mathbb{E}_\theta^{(\nu+1)}}{\mathbb{E}_\theta^{(\nu)}} \right|, \frac{\pi}{2} \le |\theta| \le \pi \right\} = \max\left\{ \left| \frac{e^{i\theta}}{2 - e^{-i\theta}} \right|, \frac{\pi}{2} \le |\theta| \le \pi \right\} = 0.45.$$

Similar values are obtained for the Gauss–Seidel iteration applied to the two- and three-dimensional version of our model problem. For a two-dimensional model Poisson problem the effect of smoothing can be seen in Figure 5.1.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

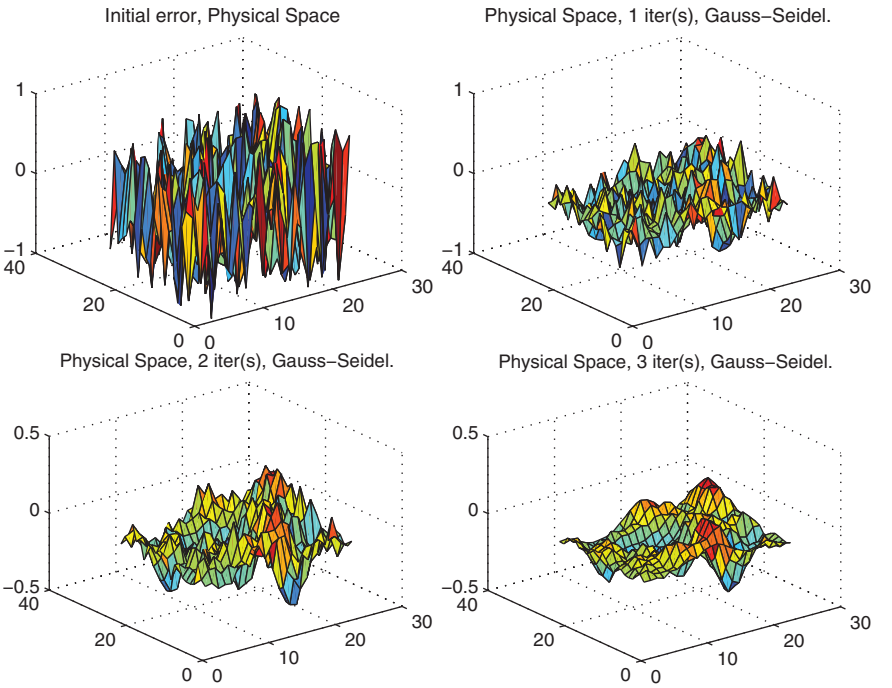

Figure 5.1. *Smoothing by Gauss–Seidel iteration.*

Another definition of smoothing property of an iterative scheme is due to Hackbusch [173]. Let M be the iteration matrix of a smoothing procedure and recall the relation $e^{(\nu)} = M^\nu e^{(0)}$. One can measure the smoothness of $e^{(\nu)}$ by a norm involving differences of the value of this error on different grid points. A natural choice is to take the second-order difference matrix A_h above. Then the following smoothing factor is defined

$$\mu(\nu) = \|A_h M^\nu\| / \|A_h\|.$$

The iteration defined by M is said to possess the smoothing property if there exists a function $\eta(\nu)$ such that, for sufficiently large ν, we have

$$\|A_h M^\nu\| \le \eta(\nu) h^{-\alpha}, \tag{5.11}$$

where $\alpha > 0$ and $\eta(\nu) \to 0$ as $\nu \to \infty$. This is the case for our model problem where A_h is the discretization of the minus Laplacian and using the damped Jacobi iteration, $M = I_h - \omega h^2 A_h$, $\omega \in (0, 1/2)$. To show this fact, recall the following lemma [173].

Lemma 5.2. *Let B be real symmetric semipositive definite matrix such that $0 \le B \le I$, $0 < \gamma \le 1$, and ν is a positive integer. Then*

$$\|B(I - \gamma B)^\nu\| \le \hat{\eta}(\nu), \qquad \hat{\eta}(\nu) = \frac{\nu^\nu}{\gamma(\nu+1)^{\nu+1}}.$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Proof. Notice that the spectrum $\sigma(B) \in (0,1)$ and that $\|f(B)\| = \max\{|f(\lambda)|, \lambda \in \sigma(B)\}$. Find the maximum of the function $f(x) = x(1-\gamma x)^{\nu}$. □

Now, we have that there exists a constant C such that $\|h^2 A_h\| \leq C$; therefore the matrix $B = \frac{h^2}{C} A_h$ satisfies the conditions of Lemma 5.2. Hence, the smoothing property is given for $\omega < 1/C$ and with $\alpha = 2$ and $\eta(\nu) = (\frac{1}{\omega})\nu^{\nu}/(\nu+1)^{(\nu+1)}$. For the Gauss–Seidel iteration one can prove that the smoothing property holds with $\alpha = 2$ and $\eta(\nu) \approx 1/\nu$.

5.2.2 Iterative Methods as Minimization Schemes

The discussion on iterative schemes given above is typical within the classical multigrid framework where multigrid operators are characterized by their properties on the Fourier space. In our context, however, we are particularly interested in the optimization properties of the various multigrid components. In this section, we consider iterative schemes from this point of view. We use the equivalence between solving the problem $Au = f$, where A is symmetric and positive definite, and minimizing the functional

$$J(u) = \frac{1}{2} u^{\top} A u - u^{\top} f. \tag{5.12}$$

Recall that many iterative methods like Jacobi and Gauss–Seidel schemes can be written in terms of a nonsingular matrix R as follows

$$u^{(\nu+1)} = (I - RA)u^{(\nu)} + Rf = u^{(\nu)} + R(f - Au^{(\nu)}) = u^{(\nu)} + R r^{(\nu)}, \tag{5.13}$$

where $r^{(\nu)} = f - Au^{(\nu)} = -\nabla J(u^{(\nu)})$ is the residual for the $u^{(\nu)}$ approximation. With $R^{-1} = D/\omega$ we have the damped Jacobi iteration; choosing $R^{-1} = D - L$ (resp., $R^{-1} = D + U$), the forward (resp., backward) Gauss–Seidel scheme is obtained.

Using (5.13) in (5.12), we have

$$\begin{aligned} J(u^{(\nu+1)}) &= \frac{1}{2}(u^{(\nu+1)})^{\top} A u^{(\nu+1)} - (u^{(\nu+1)})^{\top} f \\ &= \frac{1}{2}(u^{(\nu)} + R r^{(\nu)})^{\top} A (u^{(\nu)} + R r^{(\nu)}) - (u^{(\nu)} + R r^{(\nu)})^{\top} f \\ &= J(u^{(\nu)}) + \frac{1}{2}(R r^{(\nu)})^{\top} A R r^{(\nu)} + (R r^{(\nu)})^{\top}(A u^{(\nu)} - f) \\ &= J(u^{(\nu)}) + \frac{1}{2}(R r^{(\nu)})^{\top} A R r^{(\nu)} - (R r^{(\nu)})^{\top} r^{(\nu)}. \end{aligned}$$

We obtain the following

$$J(u^{(\nu+1)}) = J(u^{(\nu)}) - (R r^{(\nu)})^{\top} \left[\left(R^{-1} - \frac{1}{2} A \right) R r^{(\nu)} \right].$$

In the case of the Gauss–Seidel iteration we have

$$R^{-1} - \frac{1}{2} A = \frac{1}{2} D - \frac{1}{2}(L - U)$$

and therefore

$$(R r^{(\nu)})^{\top} \left[\left(R^{-1} - \frac{1}{2} A \right) R r^{(\nu)} \right] = \frac{1}{2}(R r^{(\nu)})^{\top} [D R r^{(\nu)}] \geq c \|\nabla J(u^{(\nu)})\|^2 > 0,$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

where we used the fact that $(R r^{(\nu)})^\top [(L-U) R r^{(\nu)}] = 0$ because $L-U$ is antisymmetric and we set $c = \lambda_{min}(R^\top D R)/2$. Hence, we find that the Gauss–Seidel scheme is a minimizer in the sense that

$$J(u^{(\nu)}) - J(u^{(\nu+1)}) \geq c \, \|\nabla J(u^{(\nu)})\|^2.$$

Summing this inequality over ν results in the following

$$J(u^{(0)}) - J(u^{(k+1)}) \geq c \sum_{\nu=0}^{k} \|\nabla J(u^{(\nu)})\|^2.$$

Then taking the limit as ν goes to $+\infty$, we obtain

$$\lim_{\nu \to +\infty} \nabla J(u^{(\nu)}) = 0,$$

as $J(u)$ is bounded below.

Next, consider the case of the damped Jacobi iteration where

$$R^{-1} - \frac{1}{2} A = \frac{1}{\omega} \left(D - \frac{\omega}{2} A \right).$$

We have the following lemma [367].

Lemma 5.3. *Let A be real symmetric with $a_{ii} > 0$, and let $\omega > 0$. The matrix $2\omega^{-1} D - A$, where $D =$ diag A, is positive definite if and only if ω satisfies*

$$0 < \omega \leq \frac{2}{1 - \mu_{min}},$$

where $\mu_{min} \leq 0$ is the minimum eigenvalue of $I - D^{-1} A$.

Proof. Let $B = I - D^{-1} A$. The matrix $2\omega^{-1} D - A$ is positive definite if and only if

$$2\omega^{-1} I - D^{-1/2} A D^{-1/2} = (2\omega^{-1} - 1) I + D^{1/2} B D^{-1/2} = H$$

is positive definite. The eigenvalues of H are $2\omega^{-1} - 1 + \mu_i$, where μ_i are the eigenvalues of B. Since $Tr\, B = 0$ and the μ_i are real, it follows that $\mu_{min} \leq 0$. Therefore H is positive definite if $2\omega^{-1} - 1 + \mu_i > 0$, that is, if $0 < \omega \leq \frac{2}{1-\mu_{min}} \leq \omega \leq \frac{2}{1-\mu_i}$. ∎

Therefore $(D - \frac{\omega}{2} A) \geq 0$ for $\omega \in (0, 2/(1 - \mu_{min}))$ and hence

$$J(u^{(\nu)}) - J(u^{(\nu+1)}) \geq c \, \|\nabla J(u^{(\nu)})\|^2,$$

where $c = \lambda_{min}(R^\top (2\omega^{-1} D - A) R)/2$. It follows that the damped Jacobi iteration provides a minimizing sequence such that $\lim_{\nu \to +\infty} \nabla J(u^{(\nu)}) = 0$.

In a classical multigrid context, the criteria for choosing an iteration scheme is its ability to smooth errors. In an optimization context, it is required that the iterative scheme be a minimizer. Thus many other well-known iterative methods can be chosen like, for example, the steepest descent (gradient) method given by

$$R = R^{(\nu)} = \left(\frac{{r^{(\nu)}}^\top r^{(\nu)}}{{r^{(\nu)}}^\top A r^{(\nu)}} \right) I = \alpha_\nu I.$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Also notice that $r^{(\nu)} = -J'(u^{(\nu)})$. Therefore we can write $u^{(\nu+1)} = u^{(\nu)} + \alpha_\nu r^{(\nu)}$. It follows that $J(u^{(\nu+1)}) = J(u^{(\nu)}) - \frac{\alpha_\nu}{2}\|\nabla J(u^{(\nu)})\|^2$.

The iterative schemes discussed above can be interpreted as the process of minimizing the functional J by optimizing successively with respect to each unknown variable (Gauss–Seidel scheme) or in parallel by updating all unknown variables at the same time (Jacobi scheme, steepest descent). In this sense these methods belong to the class of successive or parallel subspace correction (SSC or PSC) methods [334] and coordinate descent methods [341].

Convergence rates for SSC and PSC iterations applied to a convex functional $J(u)$ are proved in [334] assuming that $J : V \to \mathbb{R}$ is Gâteaux differentiable and that there exist constants $K, L > 0$, $p \geq q > 1$, such that

$$\langle J'(u) - J'(v), w - v\rangle \geq K\,\|u - v\|_V^p, \tag{5.14}$$

$$\|J'(u) - J'(v)\|_{V'} \leq L\,\|u - v\|_V^{q-1}, \tag{5.15}$$

for all $u, v \in V$, and $\langle\cdot,\cdot\rangle$ is the duality pairing between V and its dual space V'.

5.2.3 The Twogrid Scheme and the Approximation Property

After the application of ν_1 smoothing sweeps to the problem $A_h u_h = f_h$, we obtain an approximation $\tilde{u}_h$ whose error $\tilde{e}_h = u_h - \tilde{u}_h$ is smooth. Then $\tilde{e}_h$ can be approximated on a coarser space. We need to interpret this smooth error as the solution of a coarse problem whose matrix A_H and right-hand side have to be defined. For this purpose notice that in our model problem A_h is the second-order difference operator, approximating the one-dimensional minus Laplacian, and the *residual* $r_h = f_h - A_h\tilde{u}_h$ is a smooth function if $\tilde{e}_h$ is smooth. Obviously, because of linearity the original equation $A_h u_h = f_h$ and the residual equation $A_h\tilde{e}_h = r_h$ are equivalent. The difference is that $\tilde{e}_h$ and r_h are smooth; therefore we can think of representing them on a coarser grid with mesh size $H = 2h$. We define r_H as the restriction of the fine-grid residual to the coarse grid, that is, $r_H = I_h^H r_h$, where I_h^H is a suitable restriction operator (e.g., straight injection). This defines the right-hand side of the coarse problem. Since $\tilde{e}_h$ is the solution of a difference operator which can be represented analogously on the coarse discretization level, we define the following coarse problem

$$A_H\tilde{e}_H = r_H. \tag{5.16}$$

Here A_H represents the same discrete operator but relative to the grid with coarse mesh size H. Reasonably, one expects $\tilde{e}_H$ to be an approximation to $\tilde{e}_h$ on the coarse grid. Because of its smoothness, then we can apply a *prolongation* operator I_H^h to transfer $\tilde{e}_H$ to the fine grid. Therefore, since by definition $u_h = \tilde{u}_h + \tilde{e}_h$, we update the function $\tilde{u}_h$ applying the following coarse-grid correction step

$$\tilde{u}_h^{new} = \tilde{u}_h + I_H^h\tilde{e}_H.$$

Notice that $\tilde{e}_h$ was a smooth function and the last step has amended $\tilde{u}_h$ by its smooth error. In practice, also the interpolation procedure may introduce HF errors on the fine grid. Therefore it is convenient to complete the twogrid (TG) process by applying ν_2 postsmoothing sweeps after the coarse-grid correction.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

We summarize the TG procedure with the following algorithm. To emphasize that the iteration $u_h^{(l)} = Mu_h^{(l-1)} + Nf_h$ is a smoothing procedure, we denote it by $u_h^{(l)} = S_h(u_h^{(l-1)}, f_h)$. When no confusion may arise, we also use S to denote the iteration matrix (in place of) M. Let u_h^0 be the starting approximation. A TG scheme is as follows.

ALGORITHM 5.1. TG scheme.

- Input: initial approx. $u_h^{(0)}$, V-cycle index $n = 0$, maximum n_{max}, tolerance tol.

1. While ($n < n_{max}$ && $\|r_h\| > tol$) do
2. Presmoothing steps: $u_h^{(l)} = S(u_h^{(l-1)}, f_h), l = 1, \ldots, \nu_1$;
3. Computation of the residual: $r_h = f_h - A_h u_h^{(\nu_1)}$;
4. Restriction of the residual: $r_H = I_h^H r_h$;
5. Solution of the coarse-grid problem $e_H = (A_H)^{-1} r_H$;
6. Coarse-grid correction: $u_h^{(\nu_1+1)} = u^{(\nu_1)} + I_H^h e_H$;
7. Postsmoothing steps: $u_h^{(l)} = S(u_h^{(l-1)}, f_h), l = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1$;
8. Set $n = n + 1$;
9. End while

A TG scheme starts at the fine level with presmoothing, performs a coarse-grid correction solving a coarse-grid auxiliary problem, and ends with postsmoothing. This procedure defines one cycle. A pictorial representation of this process where "fine" is a high level and "coarse" is a low level looks like a "V" workflow. This is called the V-cycle. To solve the problem to a given tolerance, we have to apply the TG V-cycle iteratively. In fact, in Algorithm 5.1 a while loop is implemented that applies the TG cycle at most n_{max} times or until a given tolerance on the residual is achieved. Later, we shall define only the cycle and omit the repeated call of the cycle. The TG iterative scheme can be written in the form (5.1) as stated by the following lemma.

Lemma 5.4. *The TG iteration matrix is given by*

$$M_{TG} = S_h^{\nu_2}(I_h - I_H^h (A_H)^{-1} I_h^H A_h) S_h^{\nu_1}, \tag{5.17}$$

where I_h is the identity and S_h is the smoothing iteration matrix.

Proof. For the proof, notice that the coarse-grid correction gives $e_h^{(\nu_1+1)} = e^{(\nu_1)} - I_H^h e_H$. □

For the model problem considered here, it is possible to estimate the spectral radius of M_{TG}. Consider the damped Jacobi smoother with $\omega = 1/2$, assume that I_H^h is the piecewise

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

linear interpolation given by

$$I_H^h = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{pmatrix},$$

and let I_h^H be restriction by weighting such that $r_H(x_j) = (r_h(x_{j-1}) + 2r_h(x_j) + r_h(x_{j+1}))/4$, $j = 2,4,\ldots,n-1$. In stencil form we have

$$I_H^h = \frac{1}{4}\begin{pmatrix} 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \end{pmatrix}.$$

With this setting the following theorem is proved [173] using discrete Fourier analysis.

Theorem 5.5. *Let the TG scheme* 5.1 *with* $\nu = \nu_1 + \nu_2 \geq 1$. *The spectral radius of the iteration matrix* M_{TG} *given by* (5.17) *is bounded by*

$$r(M_{TG}) \leq \max\{\chi(1-\chi)^\nu + (1-\chi)\chi^\nu \quad : \quad 0 \leq \chi \leq 1/2\} =: r_\nu < 1$$

uniformly with respect to the mesh size h. *Hence* (5.17) *is a convergent iteration.*

In the framework of local Fourier analysis, a simple and effective way to predict the convergence factor of the TG scheme, for usually moderate values of ν, is to assume that the coarse-grid correction step solves "exactly" the LF error components, and there is no interaction between HF and LF components. This can be considered an "ideal" situation. Then the reduction of a suitable norm of the error (e.g., discrete L^2-norm) by one V-cycle of the TG method is determined by the reduction of the HF components on the fine grid. For this reason the convergence factor, denoted by ρ, can be roughly estimated by

$$\rho_I = \mu^{\nu_1+\nu_2}. \tag{5.18}$$

A sharper bound can be computed by TG Fourier analysis [85]. For this purpose we need to construct the Fourier symbol of the TG coarse-grid correction operator

$$CG_h^H = [I_h - I_H^h (A_H)^{-1} I_h^H A_h].$$

We denote the corresponding symbol by

$$\widehat{CG}_h^H(\theta) = [\hat{I}_h - \hat{I}_H^h(\theta)(\widehat{A}_H(2\theta))^{-1} \hat{I}_h^H(\theta)\widehat{A}_h(\theta)].$$

(Recall that $e^{i\theta x/h} = e^{i(2\theta)x/H}$.) The symbol of the coarse-grid operator is

$$\widehat{A}_H(2\theta) = -\frac{2\cos(2\theta) - 2}{H^2},$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

and similarly one constructs $\widehat{A}_h(\theta)$ corresponding to the two harmonics, that is,

$$\widehat{A}_h(\theta) = \begin{bmatrix} -\frac{2\cos(\theta)-2}{h^2} & 0 \\ 0 & \frac{2\cos(\theta)+2}{h^2} \end{bmatrix}.$$

The symbol of the restriction operator is

$$\hat{I}_h^H(\theta) = \begin{bmatrix} (1+\cos(\theta))/2 & (1-\cos(\theta))/2 \end{bmatrix},$$

whereas for the injection operator we have $\hat{I}_k^{k-1}(\theta) = 1$. For the linear prolongation operator we have $\hat{I}_{k-1}^k(\theta) = \hat{I}_k^{k-1}(\theta)^\top$.

The symbol of the TG method is given by

$$\widehat{TG}_h^H(\theta) = \hat{S}_h(\theta)^{\nu_2}\, \widehat{CG}_h^H(\theta)\, \hat{S}_h(\theta)^{\nu_1}.$$

This is a 2×2 matrix corresponding to the two frequency components. In this framework the convergence factor is defined as follows

$$\rho(TG_h^H) = \sup\left\{ r(\widehat{TG}_h^H(\theta)) : \theta \in [-\pi/2, \pi/2) \right\}.$$

In Table 5.1, we report estimates of the convergence factor ρ_I as given by (5.18) and the estimates r_{TG} resulting from the TG convergence analysis. These are compared with the estimates of ρ_ν by Theorem 5.5. The estimated ρ_I approximates well the bound ρ_ν provided that $\nu_1 + \nu_2$ is small. For large ν, ρ_I has an exponential decay behavior, whereas ρ_ν and ρ_{TG} have a slower decay as observed in numerical experiments.

Table 5.1. *Comparison of error reduction factors.*

ν	ρ_I	ρ_{TG}	ρ_ν
1	0.5	0.4	0.5
2	0.25	0.19	0.25
3	0.12	0.12	0.12
4	0.06	0.08	0.08

Notice that measuring ρ requires the knowledge of the exact solution. Because this is usually not available, ρ is measured as the asymptotic value of the reduction of a suitable norm (usually the discrete L^2-norm) of the residuals after consecutive TG cycles.

Another theoretical approach to multigrid convergence analysis, related to the smoothing property (5.11), introduces the approximation property to measure how good the coarse-grid solution approximates the fine-grid solution. This is expressed by the following estimate:

$$\| A_h^{-1} - I_H^h\, A_H^{-1}\, I_h^H \| \le c_A\, h^\beta. \tag{5.19}$$

This estimate actually measures the accuracy between $u_h = A_h^{-1}\, f_h$ and $I_H^h u_H$, where $u_H = A_H^{-1} I_h^H\, f_h$. Standard accuracy estimates for our model problem result in $\beta = 2$. This is due to the second-order accuracy of the 3-point Laplacian in one dimension and the fact that the error in interpolation and restriction is of second order.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Using the estimates of the smoothing and approximation properties, one can prove convergence of the TG scheme as follows. Consider, for simplicity, $\nu_2 = 0$; i.e., only presmoothing is applied. Then for our model problem, we have

$$\begin{aligned}\|M_{TG}\| &= \|(I_h - I_H^h (A_H)^{-1} I_h^H A_h) S_h^\nu\| \\ &= \|(A_h^{-1} - I_H^h (A_H)^{-1} I_h^H) A_h S_h^\nu\| \\ &\le \|A_h^{-1} - I_H^h (A_H)^{-1} I_h^H\| \|A_h S_h^\nu\| \\ &\le c_A \, \eta(\nu), \end{aligned} \tag{5.20}$$

where $c_A \, \eta(\nu) < 1$ for sufficiently large ν. Notice that the coarse-grid correction without pre- and postsmoothing is not a convergent iteration, in general. In fact, I_h^H maps from a fine- to a coarse-dimensional space and $I_H^h (A_H)^{-1} I_h^H$ is not invertible. We may have $r(I_h - I_H^h (A_H)^{-1} I_h^H A_h) \ge 1$.

Next, we discuss the optimization properties of the coarse-grid correction step. In fact, we show that $u_h^{(\nu+1)} = u^{(\nu)} + I_H^h e_H$ with $e_H = (A_H)^{-1} r_H$ and $r_H = I_h^H r_h$ provides an update in the descent direction in the sense that

$$\nabla J_h (u_h^{(\nu)})^\top (I_H^h e_H) < 0,$$

unless $e_H = 0$, occurring at convergence. This means that the TG scheme results in an optimization iteration by choosing a smoothing scheme with minimization properties and by performing a globalization step along the coarse-grid correction.

We assume that

$$I_h^H = c_I \, (I_H^h)^\top \text{ for a constant } c_I > 0.$$

This assumption holds, for example, for I_h^H being full-weighting restriction and I_H^h bilinear interpolation. In this case we have $c_I = (h/H)^d$, with d the space dimension. It follows that

$$\begin{aligned}\nabla J_h (u_h^{(\nu)})^\top (I_H^h e_H) &= -r_h{}^\top (I_H^h e_H) = -\frac{1}{c_I} (I_h^H r_h)^\top e_H \\ &= -\frac{1}{c_I} r_H^\top e_H = -\frac{1}{c_I} (A e_H)^\top e_H < 0. \end{aligned}$$

In general, updating along a descent direction with a unitary steplength does not guarantee a reduction of the value of the objective. For this purpose a line search or an a priori choice of steplength α is required (globalization) such that

$$J(u^{(\nu)} + \alpha \, I_H^h e_H) < J(u^{(\nu)}).$$

This issue is discussed in detail in Section 5.4. In the case of our strictly convex quadratic minimization problem, we obtain

$$\alpha^* = \operatorname*{argmin}_\alpha J(u + \alpha \, I_H^h e_H) = 1.$$

That is, a steplength of one along the coarse-grid correction direction is optimal.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

5.2.4 The Multigrid Scheme

In the TG scheme the size of the coarse problem may still be too large to be solved exactly. Therefore it is convenient to use recursively the TG method to solve the coarse problem (5.16) by introducing a further coarse-grid problem. This process can be repeated until a coarsest grid is reached where the corresponding residual equation is inexpensive to solve, e.g., by direct methods. This is a rough qualitative description of the multigrid method.

For a more detailed description, let us introduce a sequence of grids with mesh size $h_1 > h_2 > \cdots > h_L > 0$. Here $k = 1, 2, \ldots, L$ is called the level number. With Ω_{h_k} we denote the set of grid points with grid spacing h_k. The number of interior grid points will be n_k. With V_k we denote the space of functions defined on Ω_{h_k}, which are vectors of $\mathbb{R}^{n_k}$. On each level k we define the problem $A_k u_k = f_k$. Here A_k is an $n_k \times n_k$ symmetric positive definite matrix, and u_k and f_k are vectors of size n_k. The transfer among levels is performed by two linear mappings: the restriction I_k^{k-1} and the prolongation I_{k-1}^k operators. With $u_k^{(l)} = S_k(u_k^{(l-1)}, f_k)$ we denote a smoothing iteration.

For variables defined on V_k we introduce the inner product $(\cdot,\cdot)_k$ with associated norm $\|u\|_k = (u,u)_k^{1/2}$. We denote with $|u|_k = (A_k u, u)^{1/2}$ the norm induced by A_k.

The following defines one cycle of the multigrid algorithm.

ALGORITHM 5.2. Multigrid scheme.

- Input: starting approx. $u_k^{(0)}$.
  1. If $k = 1$ solve $A_k u_k = f_k$ exactly.
  2. Presmoothing steps: $u_k^{(l)} = S(u_k^{(l-1)}, f_k), l = 1, \ldots, \nu_1$;
  3. Computation of the residual: $r_k = f_k - A_k u_k^{(\nu_1)}$;
  4. Restriction of the residual: $r_{k-1} = I_k^{k-1} r_k$;
  5. Set $u_{k-1} = 0$;
  6. Call γ times the multigrid scheme to solve $A_{k-1} u_{k-1} = r_{k-1}$;
  7. Coarse-grid correction: $u_k^{(\nu_1+1)} = u_k^{(\nu_1)} + I_{k-1}^k u_{k-1}$;
  8. Postsmoothing steps: $u_k^{(l)} = S(u_k^{(l-1)}, f_k), l = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1$;
  9. End.

The multigrid algorithm involves a new parameter, called the cycle index, γ, which is the number of times the multigrid procedure is applied to the coarse-level problem. Since this procedure converges fast, $\gamma = 1$ or $\gamma = 2$ is the typical value used. For $\gamma = 1$ we have a V-cycle, whereas $\gamma = 2$ is called the W-cycle. It turns out that with a reasonable γ, the coarse problem is solved almost exactly. Therefore in this case the convergence factor of a multigrid cycle equals that of the corresponding TG method, i.e., approximately $\rho = \mu^{\nu_1+\nu_2}$. Actually in many problems $\gamma = 2$ or even $\gamma = 1$ is sufficient to retain the TG convergence. A picture of the multigrid workflow is given in Figure 5.2.

Also the multigrid scheme can be expressed in the form (5.1) as stated by the following lemma.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

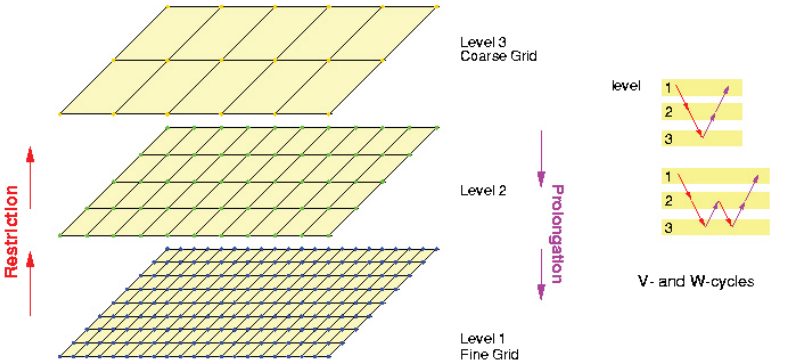

Figure 5.2. *Multigrid setting.*

Lemma 5.6. *The multigrid iteration matrix is given in recursive form by the following. For $k = 1$, let $M_1 = 0$. For $k = 2, \ldots, L$,*

$$M_k = S_k^{\nu_2}(I_k - I_{k-1}^k(I_{k-1} - M_{k-1}^{\gamma})A_{k-1}^{-1} I_k^{k-1} A_k) S_k^{\nu_1}, \tag{5.21}$$

where I_k is the identity, S_k is the smoothing iteration matrix, and M_k is the multigrid iteration matrix for the level k.

Proof. To derive (5.21) consider an initial error $e_k^{(0)}$. The action of ν_1 presmoothing steps gives $e_k = S_k^{\nu_1} e_k^{(0)}$ and the corresponding residual $r_k = A_k e_k$. On the coarse grid, this error is given by $e_{k-1} = A_{k-1}^{-1} I_k^{k-1} r_k$. However, in the multigrid algorithm we do not invert A_{k-1} (unless on the coarsest grid) and we apply γ multigrid cycles instead. That is, we denote with $v_{k-1}^{(\gamma)}$ the approximation to e_{k-1} obtained after γ application of M_{k-1}, and we have for the error (of the error) $\eta_{k-1}^{(\gamma)} = M_{k-1}^{\gamma} \eta_{k-1}^{(0)}$. That is,

$$e_{k-1} - v_{k-1}^{(\gamma)} = M_{k-1}^{\gamma}(e_{k-1} - v_{k-1}^{(0)}).$$

Following the multigrid Algorithm 5.2, we set $v_{k-1}^{(0)} = 0$ (step 5). Therefore we have $e_{k-1} - v_{k-1}^{(\gamma)} = M_{k-1}^{(\gamma)} e_{k-1}$, which can be rewritten as $v_{k-1}^{(\gamma)} = (I_{k-1} - M_{k-1}^{\gamma}) e_{k-1}$. It follows that

$$\begin{aligned} v_{k-1}^{(\gamma)} &= (I_{k-1} - M_{k-1}^{\gamma}) e_{k-1} = (I_{k-1} - M_{k-1}^{\gamma}) A_{k-1}^{-1} I_k^{k-1} r_k \\ &= (I_{k-1} - M_{k-1}^{\gamma}) A_{k-1}^{-1} I_k^{k-1} A_k e_k. \end{aligned}$$

Correspondingly, the coarse-grid correction is $u_k^{(\nu_1+1)} = u_k^{(\nu_1)} + I_{k-1}^k v_{k-1}^{(\gamma)}$. In terms of error functions this means that

$$e_k^{(\nu_1+1)} = e_k - I_{k-1}^k v_{k-1}^{(\gamma)}.$$

Substituting the expression for $v_{k-1}^{(\gamma)}$ given above, we obtain

$$e_k^{(\nu_1+1)} = [I_k - I_{k-1}^k (I_{k-1} - M_{k-1}^{\gamma}) A_{k-1}^{-1} I_k^{k-1} A_k] e_k^{(\nu_1)}.$$

Finally, consideration of the pre- and postsmoothing sweeps proves the lemma. ∎

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

We now illustrate a multigrid convergence theory due to Bramble, Pasciak, and Xu [82]. For this purpose, we consider a Poisson model problem and rewrite the multigrid iteration matrix above in the form of a classical iteration as

$$M_k = I_k - B_k A_k,$$

where I_k denotes the identity on V_k. Let $R_k : V_k \to V_k$ be an iteration operator such that $S_k = I_k - R_k A_k$ for $k > 1$. Consider

$$\begin{aligned} -\Delta u &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{5.22}$$

The matrix form of this problem is

$$A_k u_k = f_k \qquad \text{in } V_k. \tag{5.23}$$

We introduce the following operators. We interpret $I_k^{k-1} : V_k \to V_{k-1}$ as the L_k^2 projection defined by

$$(I_k^{k-1} u, v)_{k-1} = (u, I_{k-1}^k v)_k$$

for all $u \in V_k$ and $v \in V_{k-1}$. Similarly, let $P_{k-1} : V_k \to V_{k-1}$ be the A_k projection defined by

$$(A_{k-1} P_{k-1} u, v)_{k-1} = (A_k u, I_{k-1}^k v)_k$$

for all $u \in V_k$ and $v \in V_{k-1}$.

The V-cycle multigrid algorithm to solve (5.23) in recursive form is given as follows.

ALGORITHM 5.3. Multigrid scheme: recursive form.

- Set $B_1 = A_1^{-1}$. For $k \geq 2$ define $B_k : V_k \to V_k$ in terms of B_{k-1} as follows. Let $f_k \in V_k$.

  1. Define $u^{(l)}$ for $l = 1, \ldots, \nu_1$ by

  $$u^{(l)} = u^{(l-1)} + R_k(f_k - A_k u^{(l-1)}).$$

  2. Set $u^{(\nu_1+1)} = u^{(\nu_1)} + I_{k-1}^k q$, where

  $$q = B_{k-1} I_k^{k-1}(f_k - A_k u^{(\nu_1)}).$$

  3. Set $B_k f_k = u^{(\nu_1+\nu_2+1)}$, where $u^{(\ell)}$ for $\ell = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1$ is given by step 2 (with $R_k^\top$ instead of R_k for a symmetric multigrid scheme).

To simplify the analysis of this scheme, we choose $\nu_1 = 1$ and $\nu_2 = 0$, and take $u^{(0)} = 0$. From the definition of P_{k-1}, we see that

$$I_k^{k-1} A_k = A_{k-1} P_{k-1}. \tag{5.24}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Let $S_k u = S_k (u - u^{(0)}) = u - u^{(1)}$. Now for $u \in V_k$, $k = 2, \ldots, L$, we have

$$\begin{aligned}(I_k - B_k A_k) u &= u - u^{(1)} - I_{k-1}^k q \\ &= S_k u - I_{k-1}^k B_{k-1} A_{k-1} P_{k-1} S_k u \\ &= [I_k - I_{k-1}^k B_{k-1} A_{k-1} P_{k-1}] S_k u \\ &= [(I_k - I_{k-1}^k P_{k-1}) + I_{k-1}^k (I_{k-1} - B_{k-1} A_{k-1}) P_{k-1}] S_k u.\end{aligned} \tag{5.25}$$

It is immediate to see that this recurrence relation, including postsmoothing, can be written as

$$M_k = S_k [(I_k - I_{k-1}^k P_{k-1}) + I_{k-1}^k M_{k-1} P_{k-1}] S_k.$$

Clearly, it is equivalent to (5.21) with $\gamma = 1$. Starting from this recurrence relation, in [82] the following BPX multigrid convergence theorem is proved.

Theorem 5.7. *Let R_k satisfy* (5.26) *and* (5.27) *for $k > 1$. Then there exist positive constants $\delta_k < 1$ such that*

$$(A_k M_k u, u)_k \le \delta_k (A_k u, u)_k \quad \forall u \in V_k.$$

In the following, we sketch the proof of this theorem. The two conditions required in Theorem 5.7 are given below. The first condition concerns the smoothing operator R_k as follows. For simplicity, let R_k be symmetric and positive definite and S_k be nonnegative. We need $A_k S_k = S_k A_k$. There exist constants $C_R > 0$ and $c > 0$ independent of u and k such that

$$C_R \frac{\|u\|_k^2}{\lambda_k} \le (Ru, u)_k \le c (A_k^{-1} u, u)_k \quad \forall u \in V_k, \tag{5.26}$$

where λ_k denotes the largest eigenvalue of A_k. In general, notice that if the spectrum $\sigma(S_k) = \sigma(I_k - R_k A_k) \in (-1, 1)$, then there exist positive constants a_0 and a_1 smaller than one such that

$$-a_0 (A_k u, u)_k \le (A_k (I_k - R_k A_k) u, u)_k \le a_1 (A_k u, u)_k.$$

This is the same as $(1 - a_1)(A_k^{-1} u, u)_k \le (Ru, u)_k \le (1 + a_0)(A_k^{-1} u, u)_k$; see [75]. Notice that (5.26) corresponds to the smoothing property (5.11).

The second assumption is a regularity and approximation assumption. There exist $0 < \alpha \le 1$ and a constant $C_\alpha > 0$ independent of k such that

$$(A_k (I_k - I_{k-1}^k P_{k-1}) u, u)_k \le C_\alpha \left(\frac{\|A_k u\|_k^2}{\lambda_k} \right)^\alpha (A_k u, u)_k^{1-\alpha} \quad \forall u \in V_k. \tag{5.27}$$

The case $\alpha = 1$ corresponds to full elliptic regularity, i.e., $\|u\|_{H^2} \le c \|f\|_{L^2}$. Notice that (5.27) corresponds to the approximation property (5.19). In fact, we have

$$I_k - I_{k-1}^k P_{k-1} = \left(A_k^{-1} - I_{k-1}^k A_{k-1}^{-1} I_k^{k-1} \right) A_k,$$

and if $A_k u_k = f_k$, then $A_{k-1}(P_{k-1} u_k) = I_k^{k-1} f_k$.

The proof of Theorem 5.7 is now by induction. For $k = 1$ we have $M_1 = I_k - B_1 A_1 = I_k - A_1^{-1} A_1 = 0$ and the claim of the theorem is true. Now assume it is true for $k - 1$. We

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

have

$$\begin{aligned}
(A_k M_k u,u)_k &= (A_k S_k (I_k - I_{k-1}^k P_{k-1}) S_k u,u)_k + (A_k S_k I_{k-1}^k M_{k-1} P_{k-1} S_k u,u)_k \\
&= (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + (A_k I_{k-1}^k M_{k-1} P_{k-1} z,z)_k \\
&= (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + (M_{k-1} P_{k-1} z, I_k^{k-1} A_k z)_{k-1} \\
&= (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + (M_{k-1} P_{k-1} z, A_{k-1} P_{k-1} z)_{k-1} \\
&= (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + (M_{k-1} v, A_{k-1} v)_{k-1} \\
&\le (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + \delta_{k-1} (A_{k-1} v, v)_{k-1} \\
&= (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + \delta_{k-1} (A_{k-1} P_{k-1} z, P_{k-1} z)_{k-1} \\
&= (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + \delta_{k-1} (A_k z, I_{k-1}^k P_{k-1} z)_k \\
&= (1-\delta_{k-1}) (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + \delta_{k-1} (A_k z,z)_k,
\end{aligned}$$

where we let $z = S_k u$ (the case with ν pre- and postsmoothing sweeps requires $z = S_k^\nu u$) and $v = P_{k-1} z$. To complete the proof of the theorem, one considers the resulting inequality

$$(A_k M_k u,u)_k \le (1-\delta_{k-1}) (A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k + \delta_{k-1} (A_k z,z)_k. \tag{5.28}$$

Using (5.27) with $\alpha = 1$ we have

$$(A_k (I_k - I_{k-1}^k P_{k-1}) z,z)_k \le C_1 \frac{\|A_k z\|_k^2}{\lambda_k}.$$

Next, we find a δ independent of k such that Theorem 5.7 holds for all k. We need the following lemma.

Lemma 5.8. *The following estimate holds*

$$(A_k (I_k - I_{k-1}^k P_{k-1}) S_k^\nu u, S_k^\nu u)_k \le \frac{C_1}{2\nu C_R} \left(|u|_k^2 - |S_k^\nu u|_k^2 \right).$$

Proof.

$$\begin{aligned}
(A_k (I_k - I_{k-1}^k P_{k-1}) S_k^\nu u, S_k^\nu u)_k &\le C_1 \lambda_k^{-1} \|A_k S_k^\nu u\|_k^2 \\
&= (C_1/C_R) (R_k A_k S_k^\nu u, A_k S_k^\nu u)_k \\
&= (C_1/C_R) ((I - S_k) S_k^{2\nu} u, A_k u)_k \\
&\le \frac{C_1}{2\nu C_R} \left(|u|_k^2 - |S_k^\nu u|_k^2 \right).
\end{aligned}$$

The last inequality follows from

$$((I - S_k) S_k^{2\nu} u, A_k u)_k \le \frac{1}{2\nu} \sum_{j=0}^{2\nu-1} ((I - S_k) S_k^j u, A_k u)_k = \frac{1}{2\nu} \left(|u|_k^2 - |S_k^\nu u|_k^2 \right)$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

resulting from

$$(1-x)x^{2m} \le \frac{1}{2m}(1-x)\sum_{j=0}^{2m-1} x^j = \frac{1}{2m}(1-x^{2m}) \text{ for } 0 \le x \le 1. \quad \square$$

With this lemma and (5.28) (with $\delta_{k-1} = \delta$) we obtain

$$(A_k M_k u, u)_k \le (1-\delta)\frac{C_1}{2\nu C_R}\left(|u|_k^2 - |S_k^\nu u|_k^2\right) + \delta\,|S_k^\nu u|_k^2.$$

Now, choosing $\delta_k = \delta = C_1/(C_1 + 2\nu C_R)$ for all k, we have

$$(A_k M_k u, u)_k \le \delta\,(A_k u, u)_k,$$

where $0 < \delta < 1$ for $\nu \ge 1$ and Theorem 5.7 is proved with δ independent of k.

Notice that since M_k is symmetric with respect to $(A_k\cdot,\cdot)_k$, it follows that $(A_k M_k^2 u, u)_k \le \delta^2 (A_k u, u)_k$. This fact and the additional condition

$$(A_k I_{k-1}^k u, I_{k-1}^k u)_k \le 2(A_{k-1}u, u)_{k-1} \quad \forall u \in V_k,$$

which characterizes the case of nested spaces, allow us to extend the theorem above to the case of W-cycles ($\gamma = 2$), and the same estimate of δ results [75].

5.2.5 The Algebraic Multigrid Method

Since the pioneering works in [86, 291, 324] there has been extensive research for the development of multigrid methods which resemble the geometric multigrid process and utilize only information contained in the algebraic equations resulting from the discretization of the differential problem at hand. The totality of these schemes defines the class of algebraic multigrid (AMG) methods; see [325] for an exhaustive review of AMG methods available.

A step towards the theoretical development of AMG involves the definition of smooth and rough components of the error in an algebraic rather than a geometrical sense. Given a classical iterative method, a smooth error is an error which is slowly reduced by such a method. If S is the smoothing scheme, then the smooth error e is characterized by $|Se|_k \approx |e|_k$. For the Gauss–Seidel method this condition roughly means that the error is such that the residual $|r_i| \ll |a_{ii}\,e_i|$, that is,

$$\left| a_{ii}e_i + \sum_{j\in\tilde{N}_i} a_{ij}e_j \right| \ll |a_{ii}e_i|, \tag{5.29}$$

where $A = (a_{ij})$ and $\tilde{N}_i = \{j \ne i : a_{ij} \ne 0\}$.

The next step in the development of AMG is the definition of a prolongation operator I_{k-1}^k, which has the smooth error vectors in its range. Notice that based on (5.29) it is possible to approximate the smooth error component e_i as a linear combination of its neighboring error components e_j, that is,

$$e_i \approx -\sum_{j\in\tilde{N}_i} \frac{a_{ij}}{a_{ii}} e_j. \tag{5.30}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

Therefore, we can think of partitioning the whole set of unknown variables (also called points) in two subsets, C and F. The subset C has been selected as the subset of linearly independent components for the prolongation operator. It represents a coarse level of variables. The remaining variables belonging to the complement F of C are assumed to be expressed by means of an approximate version of (5.30). We have the following interpolation formula for the error function at the coarse level $k-1$ to the fine level k

$$(I_{k-1}^k e)_i = \begin{cases} \sum_{j \in P_i} w_{ij}\, e_j, & i \in F, \\ e_i, & i \in C, \end{cases} \tag{5.31}$$

where $P_i \subset \tilde{N}_i \cap C$ and $w_{ij} = -\alpha_i a_{ij}/a_{ii}$. The w_{ij}'s measure the coupling between variables. This formula is known as direct interpolation [325]. The leading coefficient α_i is introduced to take into account that, generally, $P_i \neq \tilde{N}_i$. It is selected by assuming that, for the limit case of zero row sum matrices, constants are interpolated exactly.

In order to have effective interpolation formulas, the set P_i should contain those indices j for which the absolute value of w_{ij} is larger, or at least some of them. In other words, the variables in P_i should have a certain amount of coupling for variable i. To this end the notion of strong coupling between variables is introduced. Assume, for simplicity, that $a_{ij} < 0$ whenever $i \neq j$ and that $a_{ii} > 0$. A point i is said to have strongly negatively coupling (or be n-coupled) to another point j if

$$-a_{ij} > \varepsilon_{str} \max_{a_{ik}<0} |a_{ik}| \quad \text{with } 0 < \varepsilon_{str} < 1. \tag{5.32}$$

Usually $\varepsilon_{str} = 0.2$, but this value is not critical. The set of points to which i is strongly n-coupled is

$$S_i = \left\{ j \in \tilde{N}_i \,:\, i \,\text{is strongly n-coupled to}\, j \right\}. \tag{5.33}$$

Clearly the variables in S_i should be in P_i to have an effective interpolation formula. Consequently a point that has a large amount of points strongly n-coupled to it should be put in C. This amount is the number of elements in the set $S_i^T = \left\{ j \,:\, i \in S_j \right\}$. Note that $S_i^T \neq S_i$ even for symmetric matrices, since the relation of strong n-coupling is not symmetric. Once the point i has been put in C, the points in S_i^T that are not already in C are put in F. One could implement the following procedure to make the C/F splitting of points: Let U be the set of still undecided points. Then the next C point is the undecided variable for which the quantity $\lambda_i = \left| S_i^T \cap F \right|$ is higher (given a set X, $|X|$ denotes the number of elements it contains). Note that λ_i cannot be greater than the number of nonzero off-diagonal elements of the ith column of A.

This algorithm may have two drawbacks. First, if two or more unconnected points have a high value for $\left| S_i^T \right|$, then each of these points will be the root of unrelated C/F splittings whose merging can give rise to a nonuniform distribution of C and F variables, as shown in Figure 5.3. Second, it requires a computational burden which can be too high. In fact, a vector $\Lambda = (\lambda_1 \ldots \lambda_N)$ must be initialized and ordered. Then, each time a point becomes an F point, Λ needs to be updated and reordered. The ordering is needed in order to avoid a sequential search for the highest λ_i.

Therefore it is preferable to implement a different algorithm redefining the measure of importance as $\lambda_i = \left| S_i^T \cap F \right|$. With this modification the problem of unrelated splittings no longer arises, and the vector of nonzero λ_i is of a manageable size. The size $\left| S_i^T \cap U \right|$ is taken into account only when all λ_i are zero (e.g., at the start of the splitting). But instead

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

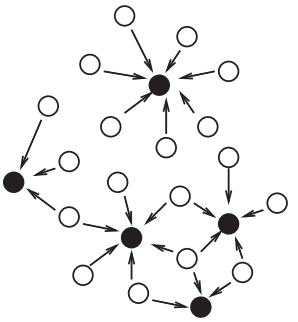

Figure 5.3. *Filled circles denote C points, empty circles denote F points, and the arrows denote strong n-coupling. Note that there is a double layer of F variables around the C point on the right. This figure first appeared in A. Borzì and G. Borzì, An algebraic multigrid method for a class of elliptic differential systems, SIAM J. Sci. Comput.,* 25(1) (2003), 302–323.

of searching for the maximal size $\left|S_i^T\right|$ among all variables, the search for the maximum is limited to a small set of randomly selected points. This does not cause much harm since $\left|S_i^T\right|$ is bounded by the number of nonzero elements in a column. These choices have been introduced in the algorithm to retain low complexity. It may happen that some points remain undecided, since they have strong n-connections only with F points. These undecided points are promoted to F points and interpolated from the neighboring F points. So, the interpolation process becomes a multipass interpolation (in this case a two-pass) [325].

Now, assume the sets C and F have been determined. The prolongation operator I_{k-1}^k is given by (5.31) with $P_i = S_i \cap C$. The restriction operator is given by the transpose of the prolongation operator, multiplied by the normalization coefficient as $I_k^{k-1} = \frac{N_{k-1}}{N_k}\left(I_{k-1}^k\right)^T$, where N_k is the number of variables at level k. The normalization coefficient is such that a restriction operator maps a constant to an almost constant vector with the same mean value.

The coarse matrix of coefficient A_{k-1} is defined by the Galerkin formula

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k.$$

The coarsening strategy defined above is called *standard coarsening*. When standard coarsening is applied to matrices with small stencils, it may result in large memory complexity, since the reduction of the number of variables is not very high and the coarse matrix has bigger stencils. In order to maintain low memory complexity or to further reduce it, *aggressive coarsening* is introduced. This corresponds to a generalization of the notion of connectivity between points: the point i has a (p,l) long-range strong n-connection with the point j if there are at least p paths of length l, each given by a sequence of points $i_0, i_1, \ldots, i_l$ with $i_0 = i$ and $i_l = j$ such that $i_{q+1} \in S_{i_q}$ for $q = 0, 1, \ldots, l-1$ (see [325]). Aggressive coarsening is made by substituting the set $S_i^{p,l} = \{j \in \Omega : i \text{ strongly n-connected to } j \text{ w.r.t. } (p,l)\}$ to S_i in the algorithm previously described.

An alternative formulation of AMG schemes is based on smoothed aggregation [350]. In this approach, the prolongator is defined by a disaggregation followed by a smoothing.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Thus the coarsening process is determined by the selection of aggregates (disjoint sets), as opposed to the selection of C-points in classical AMG. The prolongator smoother usually has the form $\tilde{S}_k = I_k - \tilde{M}_k^{-1} A_k$.

5.3 Multigrid Methods for Nonlinear Problems

Two multigrid approaches to the solution of nonlinear problems are usually considered. The first is based on a generalization of the multigrid scheme described above, which is called the full approximation storage (FAS) scheme [83]. The second approach is based on the Newton method and uses the multigrid scheme as inner solver of the linearized equations defining the Jacobian in the Newton step.

An interesting comparison between the FAS scheme and the Newton-multigrid scheme is presented in [94]. First, it is shown that in terms of computing time, the exact Newton approach is not a viable method. Further, it is demonstrated that the inexact-Newton-multigrid scheme may provide efficiency similar to that of the FAS scheme. However, it remains an open issue how many inner multigrid cycles are possibly needed in the Newton-multigrid method to match the FAS efficiency. In this sense the FAS scheme is more robust, and we discuss this method in the following. We remark that the FAS scheme is a special instance of the class of nonlinear multigrid methods discussed in [173].

5.3.1 The FAS Multigrid Method

To illustrate the FAS method, consider the discrete nonlinear problem

$$A_k(u_k) = f_k, \tag{5.34}$$

where $A_k(\cdot)$ represents a nonlinear discrete operator on Ω_{h_k}.

The starting point for the FAS scheme is again to define a suitable smoothing process denoted by $u = S(u, f)$. This can be a Picard iteration or a local Newton–Gauss–Seidel iteration; see, e.g., [57]. Suppose now one applies this iterative method a few times to (5.34), obtaining some approximate solution $\tilde{u}_k$. The desired exact correction e_k is defined by $A_k(\tilde{u}_k + e_k) = f_k$. Here the coarse residual equation makes no sense because of the nonlinearity. Nevertheless the "correction" equation can instead be written in the form

$$A_k(\tilde{u}_k + e_k) - A_k(\tilde{u}_k) = r_k, \tag{5.35}$$

where $r_k = f_k - A_k(\tilde{u}_k)$. Now assume to represent $\tilde{u}_k + e_k$ on the coarse grid in terms of the coarse-grid variable

$$u_{k-1} := \hat{I}_k^{k-1} \tilde{u}_k + e_{k-1}. \tag{5.36}$$

Since $\hat{I}_k^{k-1} \tilde{u}_k$ and $\tilde{u}_k$ represent the same function but on different grids, the standard choice of the fine-to-coarse linear operator $\hat{I}_k^{k-1}$ is straight injection. The formulation of (5.35) on the coarse level is obtained by replacing $A_k(\cdot)$ by $A_{k-1}(\cdot)$, $\tilde{u}_k$ by $\hat{I}_k^{k-1} \tilde{u}_k$, and r_k by $I_k^{k-1} r_k = I_k^{k-1}(f_k - A_k(\tilde{u}_k))$; thus we get the FAS equation

$$A_{k-1}(u_{k-1}) = I_k^{k-1}(f_k - A_k(\tilde{u}_k)) + A_{k-1}(\hat{I}_k^{k-1} \tilde{u}_k). \tag{5.37}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

This equation can also be written in the form

$$A_{k-1}(u_{k-1}) = I_k^{k-1} f_k + \tau_k^{k-1}, \tag{5.38}$$

where

$$\tau_k^{k-1} = A_{k-1}(\hat{I}_k^{k-1}\tilde{u}_k) - I_k^{k-1} A_k(\tilde{u}_k).$$

Observe that (5.38) without the τ_k^{k-1} term is the original equation represented on the coarse grid. At convergence we have $u_{k-1} = \hat{I}_k^{k-1} u_k$, because $f_k - A_k(u_k) = 0$ and $A_{k-1}(u_{k-1}) = A_{k-1}(\hat{I}_k^{k-1} u_k)$. The term τ_k^{k-1} is the fine-to-coarse *defect or residual correction*, that is, the correction to (5.38) such that its solution coincides with the fine-grid solution. This fact allows us to reverse the point of view of the multigrid approach [84]. Instead of regarding the coarse level as a device for accelerating convergence on the fine grid, one can view the fine grid as a device for calculating the correction τ_k^{k-1} to the coarse FAS equation which is computationally less expensive.

The direct use of u_{k-1} on fine grids, that is, the direct interpolation of this function by $I_{k-1}^k u_{k-1}$, cannot be used, since it introduces the interpolation errors of the full (possibly oscillatory) solution, instead of the interpolation errors of only the correction e_{k-1}, which is assumed smooth. For this reason the following coarse-grid correction is used

$$u_k = \tilde{u}_k + I_{k-1}^k (u_{k-1} - \hat{I}_k^{k-1}\tilde{u}_k). \tag{5.39}$$

A complete γ-cycle of the FAS scheme is summarized below.

ALGORITHM 5.4. FAS scheme.

- Input: starting approx. $u_k^{(0)}$.
  1. If $k = 1$ solve $A_k(u_k) = f_k$ exactly.
  2. Presmoothing steps: $u_k^{(l)} = S(u_k^{(l-1)}, f_k)$, $l = 1, \ldots, \nu_1$;
  3. Computation of the residual: $r_k = f_k - A_k(u_k^{(\nu_1)})$;
  4. Restriction of the residual: $r_{k-1} = I_k^{k-1} r_k$;
  5. Set $u_{k-1} = \hat{I}_k^{k-1} u_k^{(\nu_1)}$;
  6. Set $f_{k-1} = r_{k-1} + A_{k-1}(u_{k-1})$
  7. Call γ times the FAS scheme to solve $A_{k-1}(u_{k-1}) = f_{k-1}$;
  8. Coarse-grid correction: $u_k^{(\nu_1+1)} = u_k^{(\nu_1)} + I_{k-1}^k (u_{k-1} - \hat{I}_k^{k-1} u_k^{(\nu_1)})$;
  9. Postsmoothing steps: $u_k^{(l)} = S(u_k^{(l-1)}, f_k)$, $l = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1$;
  10. End.

The action of one FAS scheme can be also expressed in terms of a (nonlinear) multigrid iteration operator B_k. Starting with an initial approximation $u_k^{(0)}$ the result of one FAS cycle is then denoted by $u_k = B_k(u_k^{(0)})\, f_k$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

ALGORITHM 5.5. FAS V-cycle scheme: recursive form.

- Set $B_1(u_1^{(0)}) \approx A_1^{-1}$ (e.g., iterating with S_1 starting with $u_1^{(0)}$). For $k = 2, \ldots, L$ define B_k in terms of B_{k-1} as follows:
  1. Set the starting approximation $u_k^{(0)}$.
  2. Presmoothing. Define $u_k^{(l)}$ for $l = 1, \ldots, \nu_1$ by
  $$u_k^{(l)} = S_k(u_k^{(l-1)}, f_k).$$
  3. Coarse-grid correction. Set
  $$u_k^{(\nu_1+1)} = u_k^{(\nu_1)} + I_{k-1}^k(u_{k-1} - \hat{I}_k^{k-1} u_k^{(\nu_1)}),$$
  where u_{k-1} is given by
  $$u_{k-1} = B_{k-1}(u_{k-1}^{(0)})\Big[I_k^{k-1}(f_k - A_k(u_k^{(\nu_1)})) + A_{k-1}(u_{k-1}^{(0)})\Big],$$
  and $u_{k-1}^{(0)} = \hat{I}_k^{k-1} u_k^{(\nu_1)}$.
  4. Postsmoothing. Define $u_k^{(l)}$ for $l = \nu_1 + 2, \ldots, \nu_1 + \nu_2 + 1$ by
  $$u_k^{(l)} = S_k(u_k^{(l-1)}, f_k).$$
  5. Set $B_k(u_k^{(0)})\, f_k = u_k^{(\nu_1+\nu_2+1)}$.

Next, we discuss a multigrid approach to optimization which closely relates to the FAS scheme and provides a framework to investigate convergence of multigrid schemes for nonlinear problems.

5.3.2 The Full Multigrid Method

When dealing with nonlinear problems it may be essential to start the iterative procedure from a good initial approximation. The multigrid setting suggests a natural way to define this approximation. Suppose we start the solution process from a coarse working level $\ell = K < M$. At this level we solve the problem by multigrid iteration and then we interpolate the solution to the next working level to serve as initial approximation for the multigrid iterative process at this level; that is, we have

$$u_{\ell+1} = \tilde{I}_\ell^{\ell+1} u_\ell. \tag{5.40}$$

The approach of using a coarse-grid approximation as a first guess for the solution process on a finer grid is known as nested or cascadic iteration. The algorithm obtained by combining the multigrid scheme with nested iteration is called the full multigrid (FMG) method; see Figure 5.4. The interpolation operator (5.40) used in the FMG scheme is called the FMG interpolator. Because of the improvement on the initial solution at each starting level, the FMG scheme results in being more efficient than the iterative application of the multigrid cycle without FMG initialization. For Poisson model problems the com-

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

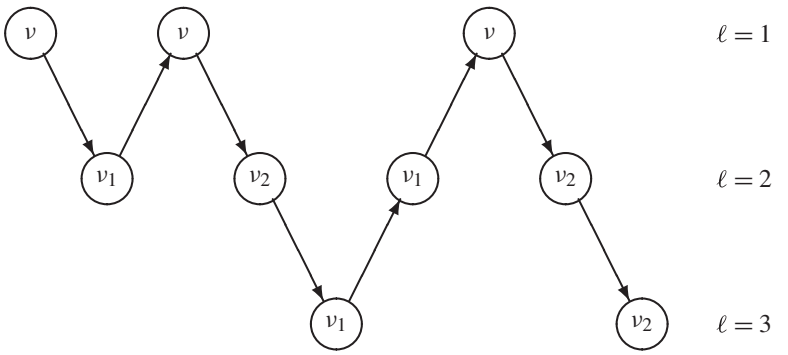

Figure 5.4. *The FMG scheme.*

putational complexity of the plain multigrid scheme is of order $\mathcal{O}(n \log n)$ while the FMG implementation provides $\mathcal{O}(n)$.

ALGORITHM 5.6. *N*-FMG scheme.

- Input: set $\ell = K$ and the initial approximation $u_\ell^{(0)}$.

  1. Compute u_ℓ applying N-cycles of the multigrid scheme with $u_\ell^{(0)}$ as initial guess.
  2. If $k = L$ then stop.
  3. Else if $\ell < L$ then by interpolation set $u_{\ell+1}^{(0)} = \tilde{I}_\ell^{\ell+1} u_\ell$.
  4. Set $\ell = \ell + 1$, goto step 2.
  5. End

Within the N-FMG algorithm an estimate of the degree of accuracy of solutions can be obtained by comparing solutions at different levels. The norm of the solution error on level ℓ can be defined as

$$E_\ell = \|u_\ell - \hat{I}_{\ell+1}^{\ell} u_{\ell+1}\|, \tag{5.41}$$

where $\hat{I}_{\ell+1}^{\ell}$ is, e.g., the injection operator.

To discuss the efficiency of the FMG approach, consider the following three-dimensional Poisson problem with Dirichlet boundary conditions

$$\begin{cases} -(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}) = 3\sin(x+y+z) & \text{in} \quad \Omega = (0,2)^3, \\ u(x,y,z) = \sin(x+y+z) & \text{for} \quad (x,y,z) \in \partial\Omega. \end{cases} \tag{5.42}$$

On each level ℓ, we choose a uniform mesh with grid size $h_\ell = \frac{2}{n_\ell + 1}$, $n_\ell = 2^\ell - 1$, and finite difference discretization gives

$$-\left(\frac{u_{i+1jk}^\ell - 2u_{ijk}^\ell + u_{i-1jk}^\ell}{h_\ell^2} + \frac{u_{ij+1k}^\ell - 2u_{ijk}^\ell + u_{ij-1k}^\ell}{h_\ell^2} + \frac{u_{ijk+1}^\ell - 2u_{ijk}^\ell + u_{ijk-1}^\ell}{h_\ell^2}\right)$$
$$= 3\sin(ih_\ell, jh_\ell, kh_\ell), \quad i,j,k = 1,\ldots,n_\ell. \tag{5.43}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Thus we use a 7-point stencil approximation which is $O(h^2)$ accurate. To solve this problem we employ the FAS scheme with $\gamma = 1$ and $L = 7$ with Gauss–Seidel smoothing, $\nu_1 = 2$ and $\nu_2 = 1$. The smoothing factor estimated by local mode analysis is $\mu = 0.567$. Hence, by this analysis, the expected reduction factor is $\rho \approx \mu^{\nu_1+\nu_2} = 0.18$. The observed convergence factor is ≈ 0.20.

Here $\hat{I}_k^{k-1}$ is simple injection and I_k^{k-1} and I_{k-1}^k are the full weighting and (tri-)linear interpolation, respectively. Finally, the FMG operator $\tilde{I}_\ell^{\ell+1}$ is cubic interpolation [84, 173]. Now let us discuss the optimality of the FMG algorithm resulting with these components.

To show that the FMG scheme is able to solve the given discrete problem at a minimal cost, we compute E_ℓ (maximum norm) and show that it behaves like h_ℓ^2, demonstrating convergence. This means that the ratio $E_\ell/E_{\ell+1}$ at convergence should be a factor of $h_\ell^2/h_{\ell+1}^2 = 4$. Results are reported in Table 5.2. We find similar results with $N = 10$, $N = 3$, and $N = 1$ FMG schemes. Therefore the choice $N = 1$ in a FMG cycle is suitable to solve the problem to second-order accuracy.

Table 5.2. *The estimated solution error for various N-FMG cycles.*

Level	10-FMG	3-FMG	1-FMG
3	$6.75\,10^{-4}$	$6.79\ \ 10^{-4}$	9.44×10^{-4}
4	$1.73\,10^{-4}$	$1.75\ \ 10^{-4}$	2.34×10^{-4}
5	$4.36\,10^{-5}$	$4.40\ \ 10^{-5}$	5.92×10^{-5}
6	$1.09\,10^{-5}$	$1.10\ \ 10^{-5}$	1.48×10^{-5}

To estimate the amount of work invested in the FMG method, let us define the work unit (WU) [83], i.e., the computational work of one smoothing sweep at the finest level. On the level $\ell \le M$ the work involved is $(\frac{1}{2})^{3(M-\ell)}WU$, where the factor $\frac{1}{2}$ is given by the mesh size ratio $h_{\ell+1}/h_\ell$ and the exponent 3 is the number of spatial dimensions. Thus a multigrid cycle that uses $\nu = \nu_1 + \nu_2$ relaxation sweeps on each level requires

$$W_{cycle} = \nu \sum_{k=1}^{L} \left(\frac{1}{2}\right)^{3(L-k)} WU < \frac{8}{7}\nu WU,$$

ignoring transfer operations. Hence the computational work employed in an N-FMG method is roughly

$$W_{FMG} = N \sum_{\ell=K}^{L} \left(\frac{1}{2}\right)^{3(L-\ell)} W_{cycle},$$

ignoring the FMG interpolation and work on the coarsest grid. This means that, using the 1-FMG method, we solve the discrete three-dimensional Poisson problem to second-order accuracy with a number of computer operations which is proportional to the number of unknowns on the finest grid. In the present case we have an FMG work of approximately $4WU$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

5.4 The Multigrid Optimization Scheme

As optimization problems become more large-sized, the need of fast optimization schemes increases. In fact, most of these problems have complicated structure that make it difficult to implement sophisticated solution strategies, thus making the use of simple gradient-based optimization schemes the only viable choice. However, in the last few years, a multigrid optimization (MGOPT) strategy has been proposed [232, 260] that can be used to accelerate classical optimization schemes. The MGOPT multigrid approach to optimization problems resembles the FAS scheme [83] and the nonlinear multigrid methods discussed in [173]. The MGOPT scheme is designed for unconstrained optimization and constructs a hierarchy of objective functionals to be minimized. However, it is formally equivalent to the FAS scheme with damping of the coarse-grid correction step [178] applied to the first-order optimality condition equations. Further contributions to this topic can be found in, e.g., [45, 152, 267].

Some successful applications to bilinear optimal control problems [344, 345], to image reconstruction problems [267], and to hyperbolic problems [232] suggest that the MGOPT framework could provide the strategy of choice to construct fast optimization schemes. However, the successful application of MGOPT schemes requires the appropriate choice of the MGOPT components that are formally similar to the classical multigrid components. In particular, a hierarchy of optimization problems on a hierarchy of optimization spaces ought to be defined. Then interspace transfer operators between coarser and finer spaces are needed. A fast optimization scheme is obtained when the optimization step obtained with the coarsening-refining process is able to complement the classical optimization scheme.

In [45, 232, 260, 345], it is emphasized that under appropriate assumptions, the multigrid coarse-grid correction provides a descent direction and, therefore, combining this fact with a linesearch procedure and a minimizing classical iteration, a globally convergent algorithm is obtained. Numerical experiments, e.g., [260, 345], demonstrate that MGOPT greatly improves the efficiency of the underlying optimization scheme used as a "smoother," suggesting that the MGOPT scheme may be beneficial in combination with well-known optimization algorithms. This claim appears to be true as long as a line search along the coarse-level correction is performed and the problem has an underlying geometrical structure for which the construction of interspace transfer operators becomes evident.

Consider the following (locally) convex optimization problem

$$\min_{u_k} J_k(u_k), \tag{5.44}$$

where $k = 1, 2, \ldots, L$ is the resolution or discretization parameter, L denotes the finest resolution, and u_k is the (unconstrained) optimization variable in the space V_k. For variables defined on V_k, we introduce the inner product $(\cdot,\cdot)_k$ with associated norm $\|x\|_k = (x,x)_k^{1/2}$. Among spaces V_k, restriction operators $I_k^{k-1} : V_k \to V_{k-1}$ and prolongation operators $I_{k-1}^k : V_{k-1} \to V_k$ are defined. We require that $(I_k^{k-1}u, v)_{k-1} = (u, I_{k-1}^k v)_k$ for all $u \in V_k$ and $v \in V_{k-1}$, that is, $I_k^{k-1} = c_I\,(I_{k-1}^k)^\top$ for a constant $c_I > 0$. Notice that for optimization problems with PDEs, the definition of the hierarchy of spaces V_k and of the intergrid transfer operators follows the guidelines of geometrical/algebraical multigrid techniques. However, in principle the MGOPT framework is not restricted to PDE-based optimization

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

problems, and therefore it could be applied to minimization problems without a geometric context. In this case, given J_L and V_L at the finest resolution it is an open issue of how to choose the hierarchy of J_k and V_k. We remark that the MGOPT scheme is an iterative gradient-based optimization method. Therefore it must be formulated in the same space where the gradient is defined. In the following, we assume an L^2 formulation of the gradient.

We denote with O_k an optimization algorithm (for example, the truncated Newton scheme used in [260]) and require that given an initial approximation $u_k^{(0)}$ to the solution of (5.44), the application of O_k results in sufficient reduction as follows

$$J_k(O_k(u_k^{(0)})) \leq J_k(u_k^{(0)}) - \eta \|\nabla J_k(u_k^{(0)})\|^2 \qquad \text{for some } \eta \in (0,1).$$

To define one cycle of the MGOPT method, it is convenient to consider the minimization problem $\min_{u_k}(J_k(u_k) - (f_k, u_k)_k)$, where $f_L = 0$. Let $u_k^{(0)}$ be the starting approximation at resolution k.

ALGORITHM 5.7. MGOPT scheme.

- Input: starting approx. $u_k^{(0)}$
  1. If $k = 1$ solve $\min_{u_k}(J_k(u_k) - (f_k, u_k)_k)$ exactly, i.e., solve $\nabla J_k(u_k) = f_k$.
  2. Preoptimization. Define $u_k^{(1)} = O_k(u_k^{(0)})$.
  3. Set up and solve a coarse-grid minimization problem. Define $u_{k-1}^{(1)} = I_k^{k-1} u_k^{(1)}$. Compute the fine-to-coarse gradient correction

     $$\tau_{k-1} = \nabla J_{k-1}(u_{k-1}^{(1)}) - I_k^{k-1} \nabla J_k(u_k^{(1)}), \qquad f_{k-1} = I_k^{k-1} f_k + \tau_{k-1}.$$

     The coarse-grid minimization problem is given by

     $$\min_{u_{k-1}} (J_{k-1}(u_{k-1}) - (f_{k-1}, u_{k-1})_{k-1}). \tag{5.45}$$

     Solve (5.45) with MGOPT to obtain u_{k-1}.
  4. Line search and coarse-grid correction. Perform a line search in the $I_{k-1}^k(u_{k-1} - I_k^{k-1} u_k^{(1)})$ direction to obtain α_k that minimizes J_k. The coarse-grid correction is given by

     $$u_k^{(2)} = u_k^{(1)} + \alpha_k I_{k-1}^k (u_{k-1} - I_k^{k-1} u_k^{(1)}).$$

  5. Postoptimization. Define $u_k^{(3)} = O_k(u_k^{(2)})$.
  6. End.

Roughly speaking, the essential guideline for constructing J_k on coarse levels is that it must sufficiently well approximate the convexity properties of the functional at finest resolution. This property and the following remark give an insight into the fact that the

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

coarse-grid correction provides a descending direction; recall the discussion at the end of Section 5.2.3 and see also Lemma 5.13 below.

Remark 5.9. We have that

$$\nabla (J_{k-1}(u_{k-1}) - (f_{k-1}, u_{k-1})_{k-1})|_{u_{k-1}^{(1)}} = I_k^{k-1} \left(\nabla J_k(u_k^{(1)}) - f_k \right).$$

That is, the gradient of the coarse-grid functional at the coarse approximation $u_{k-1}^{(1)} = I_k^{k-1} u_k^{(1)}$ equals the restriction of the gradient of the fine-grid functional at corresponding fine approximation $u_k^{(1)}$.

As remarked in [261], if the coarse optimization problem approximates the corresponding fine problem, the reduction obtained in the coarse problem should approximate the reduction in the fine problem obtained after the line search. In classical optimization terms, the coarse reduction represents the predicted reduction while the actual one is that obtained in the fine grid. We have the following estimates for the predicted and actual reductions of the value of the objective [261]. Denote $\hat{J}_k(u_k) = J_k(u_k) - (f_k, u_k)_k$ and $e_{k-1} = u_{k-1} - u_{k-1}^{(1)}$. The predicted reduction, up to $O(\|e_{k-1}\|^3)$ terms, is given as follows

$$\begin{aligned} R_p &= \hat{J}_{k-1}(u_{k-1}^{(1)}) - \hat{J}_{k-1}(u_{k-1}) \\ &= -(e_{k-1}, I_k^{k-1} \nabla J_k(u_k^{(1)}))_{k-1} - \frac{1}{2} (e_{k-1}, \nabla^2 \hat{J}_{k-1}(u_{k-1}^{(1)}) e_{k-1})_{k-1}. \end{aligned}$$

The actual reduction, up to $O(\|e_{k-1}\|^3)$ terms, is given by the following

$$\begin{aligned} R_a &= \hat{J}_k(u_k^{(1)}) - \hat{J}_k(u_k^{(2)}) \\ &= -(e_{k-1}, I_k^{k-1} \nabla J_k(u_k^{(1)}))_{k-1} - \frac{1}{2} (e_{k-1}, I_k^{k-1} \nabla^2 \hat{J}_k(u_k^{(1)}) I_{k-1}^k e_{k-1})_{k-1}. \end{aligned}$$

Therefore, we can state the following.

Remark 5.10. In the MGOPT process, the difference between the fine-grid (actual) reduction of the objective value and the corresponding coarse (predicted) reduction is estimated with

$$R_a - R_p = \frac{1}{2} \left(e_{k-1}, \left[\nabla^2 \hat{J}_{k-1}(u_{k-1}^{(1)}) - I_k^{k-1} \nabla^2 \hat{J}_k(u_k^{(1)}) I_{k-1}^k \right] e_{k-1} \right)_{k-1} + O(\|e_{k-1}\|^3).$$

That is, this difference provides a measure of the discrepancy between the coarse Hessian and the projected fine Hessian.

5.4.1 Convergence of the MGOPT Method

Assume that for each k, J_k is twice Fréchet differentiable and $\nabla^2 J_k$ is strictly positive definite and satisfies the condition $(\nabla^2 J_k(u)v, v)_k \geq \beta \|v\|_k^2$ together with $\|\nabla^2 J_k(u) - \nabla^2 J_k(v)\|_k$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

$\leq \lambda \|u - v\|_k$ uniformly for some positive constants β and λ. We use the expansion

$$J_k(u+z) = J_k(u) + (\nabla J_k(u), z)_k + \frac{1}{2}\int_0^1 (\nabla^2 J_k(u+tz)z, z)_k \, dt. \tag{5.46}$$

The main tool for our discussion is the following lemma [178].

Lemma 5.11. *For $u, v \in V_k$ assume $(\nabla J_k(u), v)_k \leq 0$ and let γ be such that*

$$0 < \gamma \leq -2\delta(\nabla J_k(u), v)_k \left[\int_0^1 (\nabla^2 J_k(u + t\gamma v)\, v, v)_k dt\right]^{-1}$$

for some $\delta \in (0, 1]$. Then

$$-(1-\delta)\gamma(\nabla J_k(u), v)_k \leq J_k(u) - J_k(u+\gamma v) \leq -\gamma(\nabla J_k(u), v)_k. \tag{5.47}$$

Proof. Set $z = \gamma v$ in (5.46). The first inequality follows from the restriction to γ. The second inequality follows from the positivity of $\nabla^2 J_k$. ∎

The next lemma provides an explicit estimate for the steplength α_k for the coarse-grid gradient correction in step 4 of Algorithm 5.7.

Lemma 5.12. *For $u, v \in V_k$ assume $(\nabla J_k(u), v)_k \leq 0$ and let*

$$\alpha(u, v) = \min\left\{2, \frac{-(\nabla J_k(u), v)_k}{(\nabla^2 J_k(u)v, v)_k + \lambda \|v\|_k^3}\right\}. \tag{5.48}$$

Then

$$0 \leq -\frac{1}{2}\alpha(u, v)(\nabla J_k(u), v)_k \leq J_k(u) - J_k(u + \alpha(u, v)v). \tag{5.49}$$

Proof. For the proof it is enough to verify that Lemma 5.11 may be applied with $\gamma = \alpha(u, v)$ and $\delta = 1/2$. Notice that

$$\int_0^1 (\nabla^2 J_k(u + t\alpha v)v, v)_k \, dt \leq (\nabla^2 J_k(u)v, v)_k + \lambda \|v\|_k^3.$$

Therefore we have

$$\alpha(u, v) \leq \frac{-(\nabla J_k(u), v)_k}{(\nabla^2 J_k(u)v, v)_k + \lambda \|v\|_k^3} \leq \frac{-(\nabla J_k(u), v)_k}{\int_0^1 (\nabla^2 J_k(u + t\alpha v)v, v)_k dt}.$$

Hence α satisfies the condition of Lemma 5.11 with $\delta = 1/2$. ∎

The following lemma states that the MGOPT coarse-grid correction with steplength $0 < \alpha \leq 2$ given by Lemma 5.12 is a minimizing step. Notice that the above lemmas are formulated for a functional $J_k(u_k)$ and its gradient $\nabla J_k(u_k)$. They hold true considering $J_k(u_k) - (f_k, u_k)_k$ and $\nabla J_k(u_k) - f_k$.

Lemma 5.13. *Take $u_k^{(1)} \in V_k$ and define $u_{k-1}^{(1)} = I_k^{k-1} u_k^{(1)} \in V_{k-1}$. Denote $\hat{J}_{k-1}(u_{k-1}) = J_{k-1}(u_{k-1}) - (f_{k-1}, u_{k-1})_{k-1}$, where $f_{k-1} = I_k^{k-1} f_k + \tau_{k-1}$ and $\tau_{k-1} = \nabla J_{k-1}(u_{k-1}^{(1)}) -$*

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

$I_k^{k-1}\nabla J_k(u_k^{(1)})$. *Let* $u_{k-1} \in V_{k-1}$ *be such that* $\hat{J}_{k-1}(u_{k-1}) \le \hat{J}_{k-1}(u_{k-1}^{(1)})$ *and define* $q = I_{k-1}^k(u_{k-1} - u_{k-1}^{(1)})$. *Then*

$$\hat{J}_k(u_k^{(1)} + \alpha(u_k^{(1)}, q)q) - \hat{J}_k(u_k^{(1)}) \le \frac{1}{2}\alpha(u_k^{(1)}, q)(\nabla \hat{J}_k(u_k^{(1)}), q)_k, \tag{5.50}$$

where $\alpha(u_k^{(1)}, q)$ *is defined in Lemma* 5.12 *(strict inequality holds if* $\hat{J}_{k-1}(u_{k-1}) < \hat{J}_{k-1}(u_{k-1}^{(1)})$*).*

Proof. The proof follows from Lemma 5.12 after showing that $(\nabla \hat{J}_k(u_k^{(1)}), q)_k \le 0$. From (5.46) we obtain

$$(\nabla \hat{J}_{k-1}(u_{k-1}^{(1)}), u_{k-1} - u_{k-1}^{(1)})_k \le \hat{J}_{k-1}(u_{k-1}) - \hat{J}_{k-1}(u_{k-1}^{(1)}) \le 0.$$

Now we have

$$\begin{aligned}(\nabla \hat{J}_k(u_k^{(1)}), q)_k &= (\nabla \hat{J}_k(u_k^{(1)}), I_{k-1}^k(u_{k-1} - u_{k-1}^{(1)}))_k \\ &= (I_k^{k-1}(\nabla \hat{J}_k(u_k^{(1)})), u_{k-1} - u_{k-1}^{(1)})_{k-1} \\ &= (\nabla \hat{J}_{k-1}(u_{k-1}^{(1)}), u_{k-1} - u_{k-1}^{(1)})_{k-1} \le 0. \end{aligned} \tag{5.51}$$

For the last equality recall Remark 5.9 and the discussion at the end of Section 5.2.3. ∎

Notice that in Lemma 5.13 it is not required to solve exactly the coarse minimization problem: find $u \in V_{k-1}$ such that $\hat{J}_{k-1}(u) = \min_{u_{k-1}} \hat{J}_{k-1}(u_{k-1})$. This is (formally) required only on the coarsest grid. The following theorem states convergence of the MGOPT method.

Theorem 5.14. *The MGOPT method described above provides a minimizing iteration, and if* J *is strictly convex, then (the index* L *of the finest level is omitted)*

$$\lim_{i\to\infty} \|u^{(i)} - u\| = 0,$$

where $J(u) = \min_v J(v)$ *and* (i) *is the MGOPT cycle index.*

Proof. The proof of the first part is by induction. For $k = 2$ we have u, where $\hat{J}_1(u) = \min_{u_1} \hat{J}_1(u_1)$, and from Lemma 5.13 it follows that

$$\begin{aligned}\hat{J}_2(u_2^3) = \hat{J}_2(O_2(u_2^2)) \le \hat{J}_2(u_2^2) &= \hat{J}_2(u_2^1 + \alpha\, I_{k-1}^k(u - I_k^{k-1}u_2^1)) \\ \le \hat{J}_2(u_2^1) = \hat{J}_2(O_2(u_2^0)) &\le \hat{J}_2(u_2^0). \end{aligned} \tag{5.52}$$

If $\hat{J}_2(u_2^0) > \min_{u_2} \hat{J}_2(u_2)$, then (5.52) holds with strict inequality.

For $k > 2$, due to the induction hypothesis and because of Lemma 5.13 the theorem holds.

The sequence $\{u_L^{(i)}\}_{i\ge1}$ is in the compact set $A = \{v \in V_L : J_L(v) \le J_L(u_L^0)\}$ and $\{J_L(u_L^{(i)})\}_{i\ge1}$ is a nonincreasing sequence in the compact set $V = \{J_L(v) : v \in A\}$, so this sequence converges. We can write $\lim_{i\to\infty}(J_L(u_L^{(i)}) - J_L(u_L)) = 0$. Strict convexity and (5.46) give that $\lim_{i\to\infty} \|u_L^{(i)} - u_L\|_L = 0$. ∎

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Linear multigrid schemes including AMG methods [51, 291, 325] can be interpreted as MGOPT schemes for the quadratic functional

$$J_k(u) = \frac{1}{2}(u, A_k u)_k - (u, b_k)_k, \quad u \in V_k,$$

where $V_k = \mathbb{R}^{n_k}$ and A_k is an $n_k \times n_k$ symmetric positive definite matrix. Consider $n_{k-1} < n_k$ and take I_{k-1}^k to be full rank. Then, with the Galerkin formula $A_{k-1} = I_k^{k-1} A_k I_{k-1}^k$ and $b_{k-1} = I_k^{k-1} b_k$, one obtains a suitable coarse functional $J_{k-1}(u) = \frac{1}{2}(u, A_{k-1}u)_{k-1} - (u, b_{k-1})_{k-1}$. A TG analysis of the MGOPT scheme applied to this problem reveals that we have convergence for $\alpha = 1$. In fact, considering (5.48) with $v = I_H^h e_H$, we have

$$\alpha = -\frac{(\nabla J_h, I_H^h e_H)_h}{(A_h I_H^h e_H, I_H^h e_H)_h} = \frac{(I_h^H r_h, e_H)_H}{(A_h I_H^h e_H, I_H^h e_H)_h} = \frac{(A_H e_H, e_H)_H}{(I_h^H A_h I_H^h e_H, e_H)_H} = 1.$$

(Notice that in the linear case $\lambda = 0$.)

The MGOPT convergence theory given above applies also to analysis of the FAS scheme. For this purpose we assume the existence of a functional J_k such that $\nabla J_k(u_k) = A_k(u_k) - f_k$. Then, subject to the conditions given above, mesh-independent convergence of the FAS scheme is proved if one proves that α given by (5.48) is always greater than or equal to one.

5.4.2 The Construction of the MGOPT Components

Experience suggests that for optimization problems with an underlying geometrical and/or differential structure, strategies similar to that of geometrical multigrid methods applied to PDE problems can be followed for the construction of a hierarchy of objectives. Thus the objective J_k at level k is the approximation of the functional J at the k level. Also AMG methods provide guidelines for the implementation of a hierarchy of functionals

$$J_k(x) = \frac{1}{2}(x, A_k x)_k - (x, b_k)_k, \quad x \in V_k,$$

where $V_k = \mathbb{R}^{n_k}$ with Euclidean inner product $(\cdot,\cdot)_k$ normalized by n_k, and A_k is an $n_k \times n_k$ symmetric positive definite matrix. Consider $n_{k-1} < n_k$ and take $I_{k-1}^k : V_{k-1} \to V_k$ to be full rank. Let $I_k^{k-1} = \left(\frac{n_{k-1}}{n_k}\right)\left(I_{k-1}^k\right)^T$. Then, with the choice $A_{k-1} = I_k^{k-1} A_k I_{k-1}^k$ and $b_{k-1} = I_k^{k-1} b_k$, one obtains a suitable coarse functional

$$J_{k-1}(y) = \frac{1}{2}(y, A_{k-1} y)_{k-1} - (y, b_{k-1})_{k-1}, \quad y \in V_{k-1}.$$

Now, we discuss the construction of the MGOPT components in the general case where only the function $J : \mathbb{R}^n \to R$ is available. We focus on the case where J is twice Fréchet differentiable and $\nabla^2 J$ is strictly positive definite. We denote $n_k = n$ and $V_k = \mathbb{R}^{n_k}$.

Consider the following quadratic model of the objective. We have

$$J_k(x) = J_k(x^0) + (\nabla J_k(x^0), x - x^0) + \frac{1}{2}(x - x^0, \nabla^2 J_k(x^0)(x - x^0)), \tag{5.53}$$

where $x^0 \in V_k$ is sufficiently close to x^*, the J-minimizer.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Associated to the Hessian, we have the following eigenvalue problem

$$\nabla^2 J_k(x^0)\, v_j = \mu_j v_j, \qquad j = 1,\ldots,n,$$

where the eigenvalues μ_j result in being real and positive and the eigenvectors can be taken orthonormal. We assume that the eigenvalues have increasing value such that $\mu_i < \mu_j$ if $i < j$. Therefore, the set of vectors $(v_j)_{j=1,n_k}$ provides a basis for V_k, and we have the following

$$x^* - x^0 = c_1 v_1 + \cdots + c_n v_n.$$

Next, we assume that the iterative optimization scheme denoted by O is more effective in minimizing along some of the eigenvectors and less effective with respect to the remaining ones. Specifically, we assume that the optimization scheme is fast in solving the components in the subspace $\text{span}\{v_{m+1},\ldots,v_n\}$, $1 < m < n_k$. Hence, after the optimization step, $x^1 = O(x^0)$, we have

$$x^* - x^1 \approx c_1 v_1 + \cdots + c_m v_m. \tag{5.54}$$

Notice that (5.54) can also be written as follows

$$x^* - x^1 \approx [v_1, \ldots, v_m] \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix}.$$

This result suggests that the vectors v_j, $j = 1,\ldots,m$, provide the columns of a matrix which can play the role of the prolongation operator. Therefore, we write

$$I_{k-1}^k = [v_1, \ldots, v_m]. \tag{5.55}$$

By doing this, we identify the coarse space as the space of the coefficients $c = (c_1,\ldots,c_m)^T$. (This corresponds to the space of the error in the coarse grids of a linear multigrid method.) We set $n_{k-1} = m$ and $V_{k-1} = \mathbb{R}^{n_{k-1}}$.

In this space, an appropriate coarse function is given by the following quadratic model

$$J_{k-1}(c) = (I_k^{k-1} \nabla J_k(x^1), c) + \frac{1}{2}(c, I_k^{k-1} \nabla^2 J_k(x^1)\, I_{k-1}^k c). \tag{5.56}$$

This model has a minimum at $c^* \in V_{k-1}$ given by the solution of the following

$$\left(I_k^{k-1} \nabla^2 J_k(x^1)\, I_{k-1}^k\right) c = -I_k^{k-1} \nabla J_k(x^1),$$

which corresponds to a Newton step defined on the coarse space with a Hessian obtained by a Galerkin projection. Therefore, the solution to the coarse minimization problem defined by (5.56) provides a coarse-grid correction $I_{k-1}^k c^*$ that updates x^1 towards the minimizer x^*, that is,

$$x^* \approx x^2 = x^1 + I_{k-1}^k c^*.$$

Notice that no gradient correction has been introduced. In fact, in the coarse model (5.56) the variable c represents corrections for the minimizer, that is, differences between a given approximation and the minimizer sought. This corresponds to the classical linear multigrid approach to PDE problems. In a nonlinear approach, the coarse model is a

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

function of a coarse representation of x, which we denote by x^c, in the sense that

$$x^* - x^1 \approx I_{k-1}^k (x^c - I_k^{k-1} x^1).$$

This results in the coarse quadratic model

$$F(x^c) = J_{k-1}(x^c) - (I_k^{k-1} \nabla^2 J(x^1)\, I_{k-1}^k\, I_k^{k-1} x^1, x^c), \tag{5.57}$$

obtained neglecting terms depending only on x^1. Further, from (5.56) we have

$$\left(I_k^{k-1} \nabla^2 J_k(x^1)\, I_{k-1}^k \right) x^c = \nabla J_{k-1}(x^c) - I_k^{k-1} \nabla J_k(x^1),$$

which we use in (5.57) to obtain the following

$$F(x^c) = J_{k-1}(x^c) - (\nabla J_{k-1}(I_k^{k-1} x^1) - I_k^{k-1} \nabla J_k(x^1), x^c). \tag{5.58}$$

Thus we obtain the classical MGOPT coarse-space minimization problem; compare with (5.45). We remark that the coarse quadratic model is defined based on x^1 while the classical optimization step was analyzed based on the eigenvectors of the Hessian defined on x^0. This inconsistency becomes relevant for highly nonlinear optimization problems for which the quadratic model starts losing its validity. This occurrence could be controlled in part with the help of robust globalization procedures.

However, the open challenge is how to construct an approximation to the set of eigenvectors defining the prolongation operator without having to solve the eigenvalue problem for the Hessian. Indeed, we can easily construct a set of orthonormal vectors spanning the space where the optimization procedure effectively acts, e.g., applying the Gram–Schmidt procedure to the set of approximation increments. But what we need is the subspace orthogonal to this set.

Another, less explored possibility to construct appropriate transfer operators is to pursue the approach of AMG smoothed aggregation starting from simple disaggregation operators and using the classical optimization scheme for smoothing.

5.5 Multigrid and Reduced SQP for Parameter Optimization

Optimization problems of the form

$$\min J(y,u), \tag{5.59}$$

$$c(y,u) = 0 \tag{5.60}$$

are called parameter optimization problems if the space U for the influence variables is finite-dimensional, i.e., $u \in U = \mathbb{R}^{n_u}$, and there is no further multigrid structure conceived within U. However, the constraint (5.60) is thought of as a PDE. This situation happens frequently in simulation-based optimization problems such as, e.g., parameter identification problems or shape optimization with an a priori defined shape parametrization. Typically, the number of parameters is low—often below 100. In this framework, the method of choice is a reduced SQP approach as demonstrated in [310, 309, 118]. Some details have to be discussed if the PDE in (5.60) is solved by a multigrid method. A straightforward implementation of a *reduced SQP method* within the separability framework applied to problem (5.59)–(5.60) iterates over the following steps.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

ALGORITHM 5.8. Reduced SQP method.

- Initialize $\ell = 0, y_0, u_0$.
  1. Solve with multigrid the adjoint problem $c_y^*(y_\ell, u_\ell)p_\ell = -J_y(y_\ell, u_\ell)$ and build the reduced gradient $\nabla_u J_\ell = J_u^* + c_u^*(y_\ell, u_\ell)p_\ell$
  2. Build some approximation $B_\ell \approx H(y_\ell, u_\ell, p_\ell)$ of the reduced Hessian, e.g., by quasi-Newton update formula
  3. Solve
$$\Delta u = \arg \min_{u \in \mathcal{L}U(u_\ell)} \frac{1}{2}(u, B_\ell u) + (\nabla_u J_\ell, u),$$
where $\mathcal{L}U(u_\ell)$ denotes the linearization of U in u_ℓ
  4. Solve with multigrid the linear problem
$$c_y(y_\ell, u_\ell)\Delta y = -(c_u(y_\ell, u_\ell)\Delta u + c(y_\ell, u_\ell))$$
  5. Update $y_{\ell+1} = y_\ell + \tau\, \Delta y$ and $u_{\ell+1} = u_\ell + \tau\, \Delta u$, where τ is some linesearch updating factor in the early iterations.

The linesearch factor τ in step 5 can be determined by the use of classical merit functions as discussed in [265]. Alternatively, efficient trust region approaches are discussed in [187]. In steps 1 and 4, linear systems have to be solved in our context by application of multigrid methods. Since these systems are adjoint to each other, the natural question arises of whether this fact should be reflected in the respective multigrid solvers, as well. This fact is particularly important if the linear systems are solved up to some tolerance. Classical reduced SQP convergence theory can guarantee convergence only if we know that the reduced gradient γ can be interpreted as a derivative, i.e., we need the consistency condition

$$\nabla_u J_\ell^* = \frac{\partial}{\partial u} J(y_\ell - A c_u(y_\ell, u_\ell)u, u_\ell + u), \tag{5.61}$$

where A is an approximation to $c_y(y_\ell, u_\ell)^{-1}$ defined by the multigrid algorithm for the linearized forward problem. If we do not satisfy this consistency condition, the necessary condition $\gamma_\ell \approx 0$ for optimality cannot be used as an indication for the optimal solution and therefore as a stopping rule for the iterations if we want to use only a comparatively coarse accuracy in the forward solver. Then, this inconsistent reduced gradient might even give a direction which is not a descent direction.

In [309] a proof is given for the fact that the condition (5.61) leads to the following requirements for the construction of grid transfer operators and the smoothing operators

$${}_{\mathcal{A}}\mathcal{I}_k^{k-1} = \left({}_{\mathcal{F}}\mathcal{I}_{k-1}^k\right)^*, \quad {}_{\mathcal{A}}\mathcal{S} = ({}_{\mathcal{F}}\mathcal{S})^*, \quad {}_{\mathcal{A}}\mathcal{I}_{k-1}^k = \left({}_{\mathcal{F}}\mathcal{I}_k^{k-1}\right)^*,$$

where ${}_{\mathcal{A}}\mathcal{I}$ and ${}_{\mathcal{F}}\mathcal{I}$ with indices mean the transfer operators for the adjoint and forward problems, respectively. Here ${}_{\mathcal{A}}\mathcal{S}$ and ${}_{\mathcal{F}}\mathcal{S}$ represent the respective smoothing operators. For efficiency reasons, the accuracy in the respective linear systems can be adapted in the style of inexact reduced SQP methods as in [187], where the accuracy of the linear subproblems is continuously increased, when zooming in to the solution. This is not required when using approximate reduced SQP methods, as in [310], where the optimization problem is reformulated so that a stagnation point of the resulting approximate algorithm is always an

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

optimal solution, regardless of whether the accuracy of the linear subproblems is increased during the nonlinear iterations or not.

Often, additional constraints are to be satisfied, either formulated in the description of the set U or as a finite number of state constraints. These additional constraints can be efficiently taken care of within the setup above in so-called partially reduced SQP methods, as introduced in [310].

5.6 Schur-Complement-Based Multigrid Smoothers

The earliest multigrid optimization approaches and many later ones have been based on a smoothing concept which can be interpreted as a Schur-complement splitting of the KKT matrix. Considering a Newton iteration for the necessary conditions to problem (5.59)–(5.60), we obtain an incremental iteration where the increments in all variables

$$\mathbf{w} = \begin{pmatrix} \Delta y \\ \Delta u \\ \Delta p \end{pmatrix}$$

are the solution to the equation

$$\mathcal{A}\mathbf{w} = \begin{pmatrix} -\nabla_y L(y,u,p) \\ -\nabla_u L(y,u,p) \\ -c(y,u) \end{pmatrix} =: \mathbf{f}. \tag{5.62}$$

The function $L(y,u,p)$ is the Lagrangian of the optimization problem, and the operator matrix

$$\mathcal{A} = \begin{bmatrix} L_{yy} & L_{yu} & c_y^* \\ L_{uy} & L_{uu} & c_u^* \\ c_y & c_u & 0 \end{bmatrix} \tag{5.63}$$

is the KKT matrix, i.e., the matrix of second-order derivatives of the Lagrangian of the optimization problem. All variants of SQP methods for nonlinear problems play with variable approximations of the matrix $\mathcal{A}$ above, since the system (5.62) can be viewed as a linear-quadratic optimization problem.

For the linear-quadratic problem of Example 2.16(c), equation (2.8), the operator matrix is constant,

$$\mathcal{A} = \begin{bmatrix} I & 0 & -\Delta \\ 0 & \nu I & -I \\ -\Delta & -I & 0 \end{bmatrix},$$

where I is the identity operator in the interior of the domain Ω, and Δ represents the Laplacian with homogeneous Dirichlet boundary conditions.

Schur-complement smoothing approaches that can still be written in the form

$$\mathbf{w}^{(l)} = \mathbf{w}^{(l-1)} + \mathcal{R}(\mathbf{f} - \mathcal{A}\mathbf{w}^{(l-1)})$$

aim at maintaining a high degree of modularity in the implementation of a multigrid optimization method. If, for example, one has a fast Poisson solver for inverting $-\Delta$, one aims at iterative methods which use this fast solver and thus at a modular method. This is the starting point of the early multigrid optimization methods in [168]. Before going into more detail, we briefly refer to the basic Schur-complement approach.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

A Schur decomposition of a general 2×2-block matrix

$$K = \begin{bmatrix} A & B^\top \\ B & D \end{bmatrix},$$

with symmetric blocks A and D, and A invertible, is an explicit reformulation of a block Gauss decomposition, i.e.,

$$K \begin{bmatrix} I & -A^{-1}B^\top \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ B & S \end{bmatrix},$$

where $S = D - BA^{-1}B^\top$ is the so-called Schur complement. Obviously, in Schur-complement approaches, one needs the inverses of the blocks A and S or at least approximations of them, thus defining iterative methods rather than factorization methods. Iterative Schur-complement solvers are based on the scheme

$$\mathbf{w}^{(l)} = \mathbf{w}^{(l-1)} + \begin{bmatrix} I & -\tilde{A}^{-1}B^\top \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{A} & 0 \\ B & \tilde{S} \end{bmatrix}^{-1} (\mathbf{f} - K\,\mathbf{w}^{(l-1)}), \tag{5.64}$$

where $\tilde{A}$ and $\tilde{S}$ are approximations to A and S.

If we want to employ this technique, we first have to match the blocks in (5.63) with the blocks in the matrix K. A possible approach is the identification

$$A = \begin{bmatrix} L_{yy} & L_{yu} \\ L_{uy} & L_{uu} \end{bmatrix},$$

and B and D are chosen accordingly. The factorization is a so-called range space factorization. In many cases, the A-block thus defined may not be invertible, which is a limiting factor for the method. Therefore, this arrangement is not well suited for PDE-constrained optimization problems, unlike for variational problems like Stokes or Navier–Stokes problems [74, 362].

Interchanging the second and third rows and columns in the matrix $\mathcal{A}$ and identifying

$$A = \begin{bmatrix} L_{yy} & c_y^* \\ c_y & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} L_{uy} & c_u^* \end{bmatrix}, \qquad D = L_{uu}$$

leads to a so-called nullspace decomposition. In the iterative version of this approach, the A and S blocks are inverted only approximatively. With this decomposition the Schur complement reads as

$$S = L_{uu} - L_{uy}\, c_y^{-1} c_u - c_u^* c_y^{*-1} L_{yu} + c_u^* c_y^{*-1} L_{yy}\, c_y^{-1} c_u,$$

which is also the otherwise-called reduced Hessian that characterizes the optimization problem. Recall that coercivity of the reduced Hessian guarantees well-posedness of the overall optimization problem. For the purpose of illustration, we elaborate on the above expression in the case of Example 2.16(c). The reduced Hessian is

$$S = \nu\, I - 0\, \Delta^{-1}\, I - I\, \Delta^{-1}\, 0 + I\, \Delta^{-1}\, I\, \Delta^{-1}\, I = \nu\, I + (\Delta^{-1})^2,$$

which is the compact operator $(\Delta^{-1})^2$ perturbed by $\nu \cdot I$. In [168], Hackbusch uses this insight in proposing a multigrid smoother for integral Fredholm operators of the second

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

kind, operating on the controls u only:

$$u^{(l)} = \frac{1}{\nu}(\gamma - (\Delta^{-1})^2 u^{(l-1)}) = u^{(l-1)} + \frac{1}{\nu}\left(\gamma - S u^{(l-1)}\right),$$

where $\gamma = \Delta^{-1} z$ is the reduced gradient at zero. In the nullspace Schur-complement setting, this corresponds to choosing

$$\tilde{A} = \begin{bmatrix} 0 & c_y^* \\ c_y & 0 \end{bmatrix}, \qquad \tilde{S} = \nu I.$$

Notice that because in Example 2.16(c) we have $L_{uy} = 0$ and $L_{yu} = 0$, certain terms in the iteration (5.64) vanish, so that only one exact solution with c_y and one exact solution with c_y^* are to be performed. For this purpose a fast Poisson solver is used. But still, the forward system and the adjoint system are solved exactly. Similar ideas lead to the generalization to parabolic optimal control problems in [171, 2].

If one wants to save effort and so performs a full solution of the forward and adjoint systems not in each smoothing step but rather in successive smoothing steps for the forward and adjoint systems, the resulting iteration (5.64) is no longer a smoothing step of a multigrid method of the second kind. In [311, 240, 314, 118] iteration (5.64) is interpreted as a transforming smoothing iteration and successfully applied in various practical problems. The choices for the algorithmic blocks are

$$\tilde{A} = \begin{bmatrix} L_{yy} & \tilde{c}_y^* \\ \tilde{c}_y & 0 \end{bmatrix} \qquad \text{and} \qquad \tilde{S} = \nu I,$$

where $\tilde{c}_y$ is some approximation to c_y useful for smoothing, e.g., just the diagonal of c_y in Jacobi smoothing. The block $\tilde{S}$ mostly consists of the regularizing part, but in numerical experiments it has been shown that a deterioration of the algorithmic performance for $\nu \to 0$ can be avoided by a small number of conjugate gradient iterations for the Schur-complement system. The same iteration is used as an iterative solver in [200] and in [163] as a preconditioner for Krylov methods for the optimality conditions.

Each smoothing step of the approximate nullspace iterations for the solution of system (5.62) runs through the following steps:

(1) compute defects $\begin{pmatrix} d_a \\ d_d \\ d_c \end{pmatrix} := \mathcal{A}\mathbf{w}^{(l)} - \mathbf{f}$

(2) $\tilde{d}_c := \tilde{c}_y^{-1} d_c$

(3) $\tilde{d}_a := \tilde{c}_y^{-*}(d_a - L_{yy}\tilde{d}_c)$

(4) $\Delta u := -\tilde{S}^{-1}(d_d + L_{uy}^*\tilde{d}_c + c_u^*\tilde{d}_a)$

(5) $\Delta y := \tilde{d}_c + \tilde{c}_y^{-1} c_u \Delta u$

(6) $\Delta\lambda := \tilde{d}_a + \tilde{c}_y^{-*}(L_{yu}\Delta u - L_{yy}\Delta y)$

(7) $\mathbf{w}^{(l+1)} = \mathbf{w}^{(l)} + \begin{pmatrix} \Delta y \\ \Delta u \\ \Delta\lambda \end{pmatrix}$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

This iteration shows good smoothing properties in practical applications as demonstrated in [311, 240, 314, 118]. The convergence theory is based not on Fourier analysis as later discussed for other smoothing concepts but rather on transforming smoothers and is discussed in [315]. The Schur-complement-based smoothing approaches discussed above decouple the smoothing of the forward and adjoint equations from the smoothing of the design equation or Schur-complement equation. In this way, the smoothing algorithm for the overall optimization system is still a highly modular algorithm. The price for this high degree of modularity is the necessity to deal with the nullspace Schur complement. Typically, only the easily accessible part stemming from regularization is used as an approximation, i.e., $\tilde{S} = L_{uu}$. If the regularization parameter ν tends towards zero, this strategy runs into trouble. In [2], this problem is resolved by a more refined analysis of the reduced Hessian for inverse problems [117]. In Example 2.16(c), additional accuracy with respect to the Schur-complement system

$$\bar{S}u^l = -\gamma, \qquad \text{where} \qquad \bar{S} = \nu I + c_u^*(\tilde{c}_y^*)^{-1}\tilde{c}_y^{-1}c_u,$$

can be achieved by a small number of conjugate gradient steps as demonstrated in [315]. It should be noted that this approximate Schur complement is formed with the approximations $\tilde{c}_y \approx c_y$ which are cheaply inverted because they are used for smoothing the forward and the adjoint systems.

A variation of the nullspace Schur-complement iteration is presented in [218, 184], where the $D = L_{uu}$-block is used as a pivot instead of the A-block. The resulting Schur complement is then

$$S = \begin{bmatrix} L_{yy} & c_y^* \\ c_y & 0 \end{bmatrix} - \begin{bmatrix} L_{yu}L_{uu}^{-1}L_{uy} & L_{yu}L_{uu}^{-1}c_u^* \\ c_u L_{uu}^{-1}L_{uy} & c_u L_{uu}^{-1}c_u^* \end{bmatrix}.$$

In particular in cases similar to Example 2.16(c), where $L_{yu} = 0$ and $L_{uu} = \nu \cdot I$, we see that

$$S = \begin{bmatrix} L_{yy} & c_y^* \\ c_y & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{\nu}c_u c_u^* \end{bmatrix}.$$

This means that

$$\begin{bmatrix} L_{yy} & c_y^* \\ c_y & 0 \end{bmatrix}^{-1} S = I - \text{compact operator},$$

which makes the above-mentioned multigrid methods of the second kind applicable. So far, this has been used only in the form of two-level cascadic methods.

Schur-complement multigrid approaches for parabolic problems have not been widely used. They can be found, e.g., in [172], where the forward and the adjoint systems are solved exactly. Schur-complement approaches are advantageous in hyperbolic optimization problems because of the nondiffusive nature of the forward problem, where an exact solver can be provided. This is shown in more detail in the next section in the form of an application to optical flow problems.

A Multigrid Solver and a Hyperbolic Optimal Control Problem

Multigrid methods for optimal control problems governed by hyperbolic systems are a much less investigated field of research, partly because multigrid methods are not considered as a natural choice for the solution of hyperbolic equations, even though there are many examples of successful application of these methods to hyperbolic problems. Therefore, it

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

is more natural to use multigrid schemes in a decoupled form as discussed in the previous section on Schur-complement approaches. In fact, in many cases an elliptic nullspace Schur complement results, making the use of multigrid methods obvious.

In this context, an interesting application problem results from the optimal control formulation of the optical flow problem as proposed in [55, 56]. An optical flow is the field of apparent velocities in a sequence of images; see [24, 201, 359]. From the knowledge of the optical flow, information about the spatial arrangement of objects and the rate of change of this arrangement ought to be obtained.

The forward model is based on the assumptions that the image brightness of an object point remains constant in the images when the object moves. That is, the total time derivative of the brightness at each point (x_1, x_2) at time t is zero. This approach leads to the following *optical flow constraint* (OFC) equation

$$\frac{\partial y}{\partial t} + u\frac{\partial y}{\partial x_1} + v\frac{\partial y}{\partial x_2} = 0, \tag{5.65}$$

where $y = y(x_1, x_2, t)$ denotes the image brightness at (x_1, x_2) and t, and $\vec{w} = (u, v)$ represents the optical flow vector.

Now consider a sequence of image frames $\{Y_k\}_{k=0,N}$ on Ω sampled at increasing time steps, $t_k \in [0, T]$, $k = 0, 2, \ldots, N$, where $t_0 = 0$ and $t_N = T$. In the optimal control formulation we require estimation of $\vec{w}$ such that the resulting $y(\cdot, t_k, \vec{w})$ approximates Y_k at the sampling times. This means solving

$$\begin{cases} y_t + \vec{w} \cdot \nabla y = 0 & \text{in } Q = \Omega \times (0, T], \\ y(\cdot, 0) = Y_0 \end{cases} \tag{5.66}$$

and minimizing the cost functional

$$\begin{aligned} J(y, \vec{w}) = \frac{1}{2}\int_\Omega \sum_{k=1}^{N} |y(x_1, x_2, t_k) - Y_k|^2 d\Omega \\ + \frac{\alpha}{2}\int_Q \Phi\left(\left|\frac{\partial \vec{w}}{\partial t}\right|^2\right) dq + \frac{\beta}{2}\int_Q \Psi(|\nabla u|^2 + |\nabla v|^2) dq + \frac{\gamma}{2}\int_Q |\nabla \cdot \vec{w}|^2 dq. \end{aligned} \tag{5.67}$$

Here, α, β, and γ are predefined nonnegative weights. The term with Φ provides bounded variation type regularization across edges and corners of images, where $\nabla \vec{w}$ is large; see, e.g., [209]. The last term in (5.67) improves the filling-in properties of the optimal control solution; see [55, 56] for details.

The optimal solution is characterized by the following optimality system

$$\begin{aligned} y_t + \vec{w} \cdot \nabla y = 0, \quad \text{with } y(\cdot, 0) = Y_0, \\ p_t + \nabla \cdot (\vec{w} p) = \sum_{k=1}^{N-1} \left[\delta(t - t_k)(y(\cdot, t_k) - Y_k)\right], \quad \text{with } p(\cdot, T) = -(y(\cdot, T) - Y_N), \\ \alpha\frac{\partial^2 u}{\partial t^2} + \beta\nabla \cdot [\Psi'(|\nabla u|^2 + |\nabla v|^2)\nabla u] + \gamma\frac{\partial}{\partial x_1}(\nabla \cdot \vec{w}) = p\frac{\partial y}{\partial x_1}, \\ \alpha\frac{\partial^2 v}{\partial t^2} + \beta\nabla \cdot [\Psi'(|\nabla u|^2 + |\nabla v|^2)\nabla v] + \gamma\frac{\partial}{\partial x_2}(\nabla \cdot \vec{w}) = p\frac{\partial y}{\partial x_2}, \end{aligned} \tag{5.68}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

where δ denotes the Dirac δ-function. The interpretation of the second equation in (5.68) is

$$p_t + \nabla \cdot (\vec{w}\, p) = 0 \text{ on } t \in (t_{k-1}, t_k) \quad \text{for } k = 1, \ldots, N, \tag{5.69}$$

$$p(\cdot, t_k^+) - p(\cdot, t_k^-) = y(\cdot, t_k) - Y_k \quad \text{for } k = 1, \ldots, N-1. \tag{5.70}$$

The last two equations are nonlinear elliptic equations representing the optimality condition. As boundary conditions for $\vec{w}$ one can choose homogeneous Dirichlet boundary conditions on the spatial boundary and natural boundary conditions at the temporal boundaries of Q.

To solve (5.68), an explicit time-marching second-order TVD scheme for the forward-backward hyperbolic subsystem and a FAS multigrid method for the elliptic control equations is proposed in [55, 56]. The proposed method is summarized as follows.

ALGORITHM 5.9. Loop for solving the optical flow problem.

1. Apply the Horn and Schunck method [201] for a starting approximation to the optical flow.
2. Solve the OFC equation to obtain y.
3. Solve (backward) the adjoint OFC equation to obtain p.
4. Update the right-hand sides of the elliptic system.
5. Apply a few cycles of multigrid to solve the control equations.
6. Go to 2 and repeat I_{loop} times.

Results presented in [55, 56] show that the optimal control approach allows accurate and robust determination of optical flow also in the limit case where only two image frames are given.

A known benchmark for verification of optical flow solvers is the "Hamburg taxi sequence"; see [24]. It consists of a sequence of frames of a taxi coming from the right in the main road and turning right into a side street in Hamburg (Germany). One photo of the sequence and the corresponding brightness pattern are depicted in Figure 5.5.

We consider a sequence of five photos of the moving taxi taken at regular intervals ($T = 4$). The space-time computational domain is a $128 \times 96 \times 128$ grid, where 128 time subdivisions are taken in the time direction. This grid can be obtained from a coarse $4 \times 3 \times 4$ mesh by halving the mesh size six times. Our algorithm is applied with $\alpha = 5.0$, $\beta = 0.25$, $\gamma = 0.25$, and $I_{loop} = 10$. In Figure 5.5 the optical flow computed with the optimal control approach at $t = 2$ is presented. Comparing with the solution obtained with the Horn and Schunck method [24], the optimal control approach provides a better reconstruction optical flow for the taxi sequence.

5.7 The Collective Smoothing Multigrid Approach

A collective smoothing multigrid (CSMG) approach means solving the optimality system for the state, the adjoint, and the control variables simultaneously in the multigrid process by using collective smoothers for the optimizations variables. The CSMG approach is in contrast to the sequential solving of the state, adjoint, and control equations. The last approach requires that the uncontrolled state equations be solvable, and thus it cannot be applied to singular optimal control problems [236], where the uncontrolled system may not

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

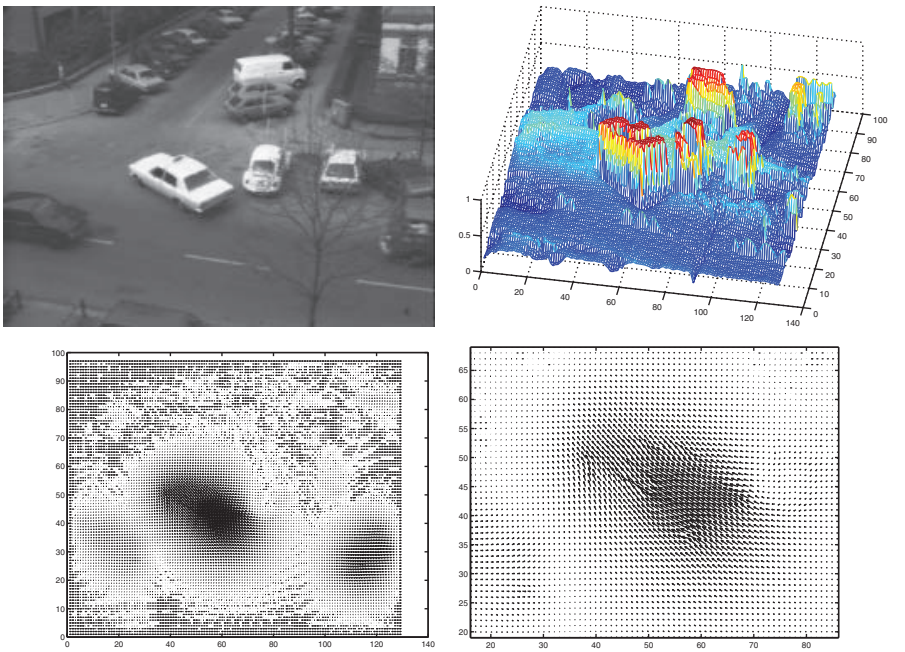

Figure 5.5. *First frame of the taxi sequence (top left); the corresponding brightness distribution (top right). Optical flow for the taxi sequence (bottom left). Close-ups of the solution containing the region of the taxi (bottom right). This figure first appeared in A. Borzì, K. Ito, and K. Kunisch, Optimal control formulation for determining optical flow, SIAM J. Sci. Comput.,* 24(3) (2002), 818–847.

have a solution or blowup in finite time. A CSMG-based scheme aims at realizing the tight coupling in /the optimality system along the hierarchy of grids. By employing collective smoothing, that is, by realizing the coupling in the optimality system at the smoothing step level, robustness and typical multigrid efficiency is achieved; see, e.g., [64].

Strategies of development of collective smoothers for optimality systems appear well established in an AMG context [51, 52] that also provide an example of application of the CSMG approach to convection-diffusion problems. For these problems, another recent contribution in a geometric multigrid context can be found in [203]. Further extensions of the CSMG strategy to problems with control or state constraints are given in [47, 48, 59]. The CSMG approach has also been successfully applied to parameter identification problems [12, 345]. In the control-constrained case, the CSMG approach allows one to construct robust multigrid schemes that apply also in the case $\nu = 0$, thus allowing the investigation of bang-bang control problems. In particular, using the multigrid scheme in [48, 59] it is possible to show the phenomenon of "chattering control" [35] for elliptic systems, which appears to be a less investigated problem. Further extension of the CSMG method can be found in [44], where this method is combined with sparse-grid techniques for solving elliptic optimal control problems with random coefficients.

The results above concern multigrid methods for linear and nonlinear elliptic optimality systems with linear and bilinear control mechanisms. Early works concerning multigrid solution of parabolic optimal control problems are [171, 172]. Within the CSMG frame-

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

work, recent contributions are given in [46, 49, 50, 53, 58, 67, 150]. The starting point for these recent developments is represented by space-time parabolic multigrid methods [170, 347] and also the approach presented in [169]. Based on the CSMG strategy it is also possible to solve bang-bang parabolic control problems [150]. In [6] the CSMG approach is extended to the solution of integral Fredholm optimal control problems.

In the following sections we discuss the development and convergence properties of CSMG schemes for control problems governed by PDEs.

5.7.1 CSMG Schemes for Elliptic Control Problems

We discuss in detail the design of a collective smoothing iteration for an elliptic optimal control problem with control constraints [48, 59]. This procedure appears to be robust with respect to changes of the value of the weight and, in particular, it allows the choice $\nu = 0$. This fact makes the CSMG scheme a useful tool to investigate bang-bang control phenomena [147].

Consider the following basic elliptic distributed optimal control problem

$$\begin{cases} \min J(y,u) & := & \frac{1}{2}\|y-z\|^2_{L^2(\Omega)} + \frac{\nu}{2}\|u\|^2_{L^2(\Omega)}, & \\ -\Delta y & = & u+g & \text{in } \Omega, \\ y & = & 0 & \text{on } \partial\Omega, \end{cases} \tag{5.71}$$

where we require that $u \in U_{ad}$ and the set of admissible controls is the closed convex subset of $L^2(\Omega)$ given by

$$U_{ad} = \{u \in L^2(\Omega) \,|\, \underline{u}(\mathbf{x}) \le u(\mathbf{x}) \le \overline{u}(\mathbf{x}) \text{ a.e. in } \Omega\}, \tag{5.72}$$

where $\underline{u}$ and $\overline{u}$ are elements of $L^\infty(\Omega)$.

As illustrated in Chapter 2, the solution to (5.71) is characterized by the following optimality system

$$\begin{array}{rcll} -\Delta y = & u+g & \text{in } \Omega, \\ y = & 0 & \text{on } \partial\Omega, \\ -\Delta p = & -(y-z) & \text{in } \Omega, \\ p = & 0 & \text{on } \partial\Omega, \\ (\nu u - p, v-u) \ge & 0 & \forall v \in U_{ad}. \end{array} \tag{5.73}$$

Notice that the last equation in (5.73) giving the optimality condition is equivalent to (see [235, 245])

$$u = \max\left\{\underline{u}, \min\left\{\overline{u}, \frac{1}{\nu}p(u)\right\}\right\} \text{ in } \Omega \quad \text{if } \nu > 0. \tag{5.74}$$

The unique solution u to (5.71)–(5.72) with $\nu = 0$ corresponds to the optimality condition given by [59]

$$p = \quad \min\{0, p+u-\underline{u}\} + \max\{0, p+u-\overline{u}\} \quad \text{in } \Omega. \tag{5.75}$$

We illustrate the construction of a smoothing iteration for the constrained-control problem with $\nu > 0$ in the framework of finite differences. Recall the discretization setting

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

introduced in Chapter 3, and consider the discrete optimality system at $\mathbf{x} \in \Omega_h$, where $\mathbf{x} = (ih, jh)$ and i, j index the grid points, e.g., lexicographically. We have

$$-(y_{i-1\,j} + y_{i+1\,j} + y_{i\,j-1} + y_{i\,j+1}) + 4y_{ij} - h^2 u_{ij} = h^2 g_{ij}, \tag{5.76}$$

$$-(p_{i-1\,j} + p_{i+1\,j} + p_{i\,j-1} + p_{i\,j+1}) + 4p_{ij} + h^2 y_{ij} = h^2 z_h, \tag{5.77}$$

$$(\nu u_{ij} - p_{ij})(v_{ij} - u_{ij}) \geq 0 \quad \forall v_h \in U_{adh}. \tag{5.78}$$

A collective smoothing step at $\mathbf{x}$ consists in updating the values y_{ij} and p_{ij} such that the resulting residuals of the two equations at that point are zero. The neighboring variables are considered constant during this process. Therefore, replacing these two constants in (5.76) and (5.77), we obtain y_{ij} and p_{ij} as functions of u_{ij} as follows

$$y_{ij} = (A_{ij} + h^2 u_{ij})/4 \tag{5.79}$$

and

$$p_{ij} = (4B_{ij} - h^2 A_{ij} - h^4 u_{ij})/16. \tag{5.80}$$

Now to obtain the u_{ij} update, replace the expression for p_{ij}, as a function of u_{ij}, in the inequality constraint and define the auxiliary variable

$$\tilde{u}_{ij} = \frac{1}{16\nu + h^4}(4B_{ij} - h^2 A_{ij}). \tag{5.81}$$

Here $\tilde{u}_{ij}$ is defined as the solution to the optimality condition equation without constraints, i.e., $\nabla \hat{J}(u) = \nu u - p(u) = 0$, and therefore (5.81) defines the u_{ij} update in the case of no constraints. In the presence of constraints, the new value for u_{ij} resulting from the smoothing step is given by projection of $\tilde{u}_{ij}$ as follows

$$u_{ij} = \begin{cases} \overline{u}_{ij} & \text{if } \tilde{u}_{ij} > \overline{u}_{ij}, \\ \tilde{u}_{ij} & \text{if } \underline{u}_{ij} \leq \tilde{u}_{ij} \leq \overline{u}_{ij}, \\ \underline{u}_{ij} & \text{if } \tilde{u}_{ij} < \underline{u}_{ij} \end{cases} \tag{5.82}$$

for all $\mathbf{x} = (ih, jh) \in \Omega_h$. With the new value of u_{ij} given, new values for y_{ij} and p_{ij} are obtained. This completes the description of the collective smoothing step. It satisfies the inequality constraint; see [59]. Further, in case $\nu = 0$ the smoothing iteration defined above satisfies (5.75). Because of (5.82) we can consider that the present iteration belongs to the class of projected iterative schemes [88].

We report results of experiments with the following objective function

$$z(x_1, x_2) = \sin(2\pi x_1)\sin(\pi x_2).$$

We choose the following constraints: $\underline{u} = -30$ and $\overline{u} = 30$.

We obtain that the constraints are active in large portions of the domain for the three choices of $\nu = \{10^{-4}, 10^{-6}, 10^{-8}\}$ considered here. For $\nu = 10^{-6}$ this can be seen in Figure 5.6. From the results of numerical experiments reported in Table 5.3 we observe that for $\nu = 10^{-4}$ the multigrid convergence behavior is similar to that observed in the unconstrained case. Reducing the value of ν results in steeper gradients of the adjoint and control variables, particularly close to the boundary where p and u are required to be zero. Furthermore, decreasing ν results in an increasingly more complex switching structure of

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

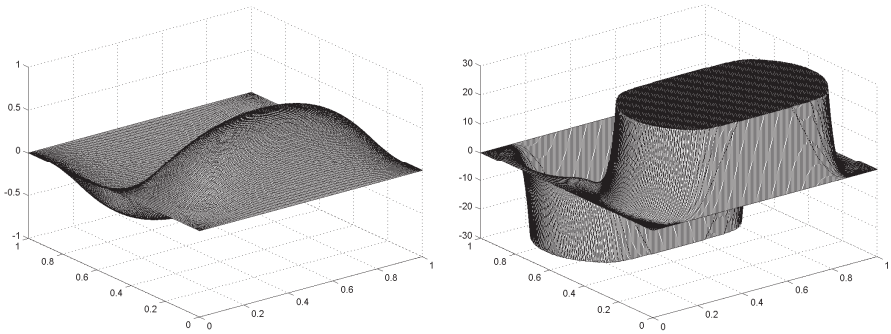

Figure 5.6. *Numerical solutions for the state (left) and the control (right);* $\nu = 10^{-6}$ *and* 513×513 *mesh. Reprinted with permission from A. Borzì and K. Kunisch, A multigrid scheme for elliptic constrained optimal control problems, Comput. Optim. Appl.,* 31(3) (2005), 309–333.

Table 5.3. *Results of experiments.*

$\nu = 10^{-4}$			
Mesh	$\rho(y)$, $\rho(p)$	$\|y-z\|_0$	$\|r(y)\|_0$, $\|r(p)\|_0$
129×129	0.04, 0.04	$1.11\,10^{-1}$	$3.1\,10^{-10}$, $1.2\,10^{-13}$
257×257	0.03, 0.04	$1.11\,10^{-1}$	$6.8\,10^{-10}$, $7.1\,10^{-14}$
513×513	0.03, 0.04	$1.11\,10^{-1}$	$4.9\,10^{-10}$, $1.5\,10^{-13}$
1025×1025	0.03, 0.03	$1.11\,10^{-1}$	$3.2\,10^{-10}$, $7.2\,10^{-13}$
$\nu = 10^{-6}$			
Mesh	$\rho(y)$, $\rho(p)$	$\|y-z\|_0$	$\|r(y)\|_0$, $\|r(p)\|_0$
129×129	0.56, 0.56	$5.30\,10^{-2}$	$1.3\,10^{-6}$, $2.2\,10^{-10}$
257×257	0.52, 0.51	$5.30\,10^{-2}$	$1.5\,10^{-7}$, $1.3\,10^{-11}$
513×513	0.03, 0.03	$5.30\,10^{-2}$	$3.5\,10^{-10}$, $5.3\,10^{-14}$
1025×1025	0.03, 0.03	$5.30\,10^{-2}$	$2.2\,10^{-10}$, $2.2\,10^{-13}$
$\nu = 10^{-8}$			
Mesh	$\rho(y)$, $\rho(p)$	$\|y-z\|_0$	$\|r(y)\|_0$, $\|r(p)\|_0$
129×129	0.63, 0.63	$5.28\,10^{-2}$	$1.6\,10^{-3}$, $8.3\,10^{-8}$
257×257	0.54, 0.54	$5.28\,10^{-2}$	$2.4\,10^{-6}$, $7.4\,10^{-11}$
513×513	0.64, 0.60	$5.28\,10^{-2}$	$2.5\,10^{-7}$, $3.7\,10^{-12}$
1025×1025	0.68, 0.66	$5.28\,10^{-2}$	$2.7\,10^{-7}$, $2.1\,10^{-12}$
2049×2049	0.74, 0.71	$5.28\,10^{-2}$	$7.8\,10^{-7}$, $3.5\,10^{-12}$
4097×4097	0.76, 0.70	$5.28\,10^{-2}$	$7.4\,10^{-8}$, $2.9\,10^{-12}$

the control between upper and lower bounds; see Figure 5.7. The results for $\nu = 10^{-6}$ in Table 5.3 suggest that once the mesh size is sufficiently fine to resolve completely the switching structure the typical multigrid convergence rate is obtained. They further indicate that the multigrid convergence factor depends only weakly on the mesh size provided the problem is sufficiently well resolved on the mesh.

The ability of the multigrid scheme in solving constrained control problems with very small value of ν allows us to investigate the occurrence of bang-bang control for the

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

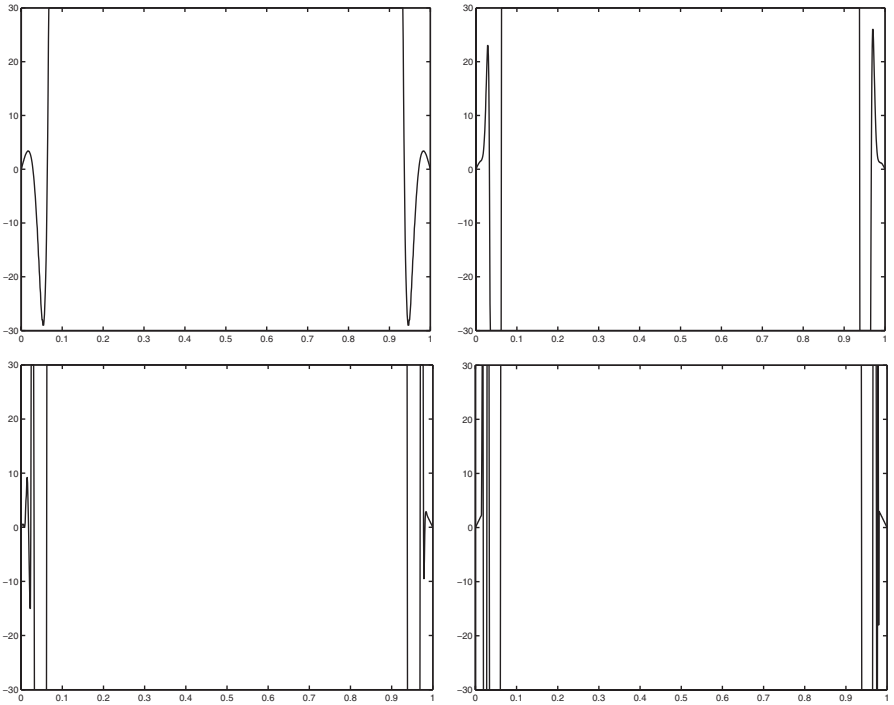

Figure 5.7. *The control function for* $x_1 = 3/4$ *and* $x_2 \in [0,1]$ *obtained with* $\nu = 10^{-8}$ *(top left),* $\nu = 10^{-10}$ *(top right),* $\nu = 10^{-12}$ *(bottom left), and* $\nu = 0$ *(bottom right);* 2049×2049 *mesh. Reprinted with permission from A. Borzì and K. Kunisch, A multigrid scheme for elliptic constrained optimal control problems, Comput. Optim. Appl.,* 31(3) (2005), 309–333.

present class of problems. In particular, with the choice of z given above we can observe fast switching of the control function in the x_2 direction as depicted in Figure 5.7. In this figure we give plots of the control function for $x_1 = 3/4$ and $x_2 \in [0,1]$ for the following choices $\nu \in \{10^{-8}, 10^{-10}, 10^{-12}, 0\}$. We can see that as the value of ν is reduced the number of switching points increases.

The solution obtained for $\nu = 0$ is interesting. In this case, by further refining the mesh size, additional switching points can be seen closer to the boundary while the existing switching points obtained at the previous coarser grids are retained.

We complete this discussion considering another desired state given by

$$z_1(x_1, x_2) = \sin(4\pi x_1)\sin(2\pi x_2).$$

The difference between this objective function and the previous one is that the gradient of z_1 is larger close to the boundary. For the choice $\nu = 0$ the constraints are everywhere active; i.e., the control is bang-bang. Moreover, no fast switching of the control occurs. In Figure 5.8 the optimal control and the corresponding state for $\nu = 0$ are depicted. The numerical results in Table 5.4 document the convergence factors.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

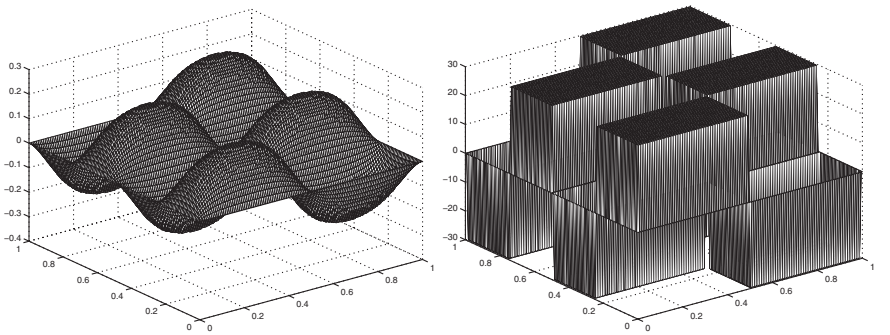

Figure 5.8. *Numerical solutions with z_1 and $\nu = 0$. The state (left) and the control (right);* 257×257 *mesh. Reprinted with permission from A. Borzì and K. Kunisch, A multigrid scheme for elliptic constrained optimal control problems, Comput. Optim. Appl.,* 31(3) (2005), 309–333.

Table 5.4. *Results of experiments with z_1 and $\nu = 0$.*

Mesh	$\rho(y)$, $\rho(p)$	$\|y-z\|_0$	$\|r(y)\|_0$, $\|r(p)\|_0$
513×513	0.12, 0.13	$3.70\,10^{-1}$	$2.9\,10^{-8}$, $1.3\,10^{-13}$
1025×1025	0.12, 0.13	$3.70\,10^{-1}$	$2.5\,10^{-8}$, $4.2\,10^{-13}$
2049×2049	0.12, 0.16	$3.70\,10^{-1}$	$1.9\,10^{-8}$, $1.6\,10^{-12}$

The approach described above can be extended to the case of boundary optimal control problems with constraints. Consider

$$\begin{cases} \min J(y,u) & := & \frac{1}{2}||y-z||^2_{L^2(\Omega)} + \frac{\nu}{2}||u||^2_{L^2(\partial\Omega)}, & \\ -\Delta y + y & = & g & \text{in } \Omega, \\ \frac{\partial y}{\partial n} & = & u & \text{on } \partial\Omega, \end{cases} \tag{5.83}$$

where Ω is an open bounded set of $\mathbb{R}^2$, $g \in L^2(\Omega)$, and $z \in L^2(\Omega)$ is the objective function. The set of admissible controls is given in this case by

$$U_{ad} = \{u \in L^2(\partial\Omega) \,|\, \underline{u}(\mathbf{x}) \le u(\mathbf{x}) \le \overline{u}(\mathbf{x}) \text{ a.e. in } \partial\Omega\}, \tag{5.84}$$

where $\underline{u}$ and $\overline{u}$ are functions of $L^\infty(\partial\Omega)$.

For the existence of a unique solution to (5.83)–(5.84) we refer the reader to [235, 339]. The solution is characterized by the following optimality system

$$\begin{array}{rcll} -\Delta y + y & = & g & \text{in } \Omega, \\ \frac{\partial y}{\partial n} & = & u & \text{on } \partial\Omega, \\ -\Delta p + p & = & -(y-z) & \text{in } \Omega, \\ \frac{\partial p}{\partial n} & = & 0 & \text{on } \partial\Omega, \\ (\nu u - p, v-u) & \ge & 0 & \forall v \in U_{ad}. \end{array} \tag{5.85}$$

To implement the control on the boundary, one can approximate $\frac{\partial}{\partial n}$ by centered difference quotient and combine the Neumann boundary conditions with the stencil of the

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Laplace operator considered at the boundary. We discuss this approach explicitly for one lateral boundary of $\Omega = (0,1)\times(0,1)$.

Let $\mathbf{x} = (ih, jh)$ be a boundary grid point on the side $x = 0$. Based on the finite differences framework of Chapter 3, we obtain the following discretized state and adjoint equations

$$\begin{aligned} -(y_{i-1\,j} + y_{i+1\,j} + y_{i\,j-1} + y_{i\,j+1}) + (4+h^2)y_{ij} &= h^2 g_{ij}, \\ y_{i-1\,j} - y_{i+1\,j} &= 2hu_{ij}, \\ -(p_{i-1\,j} + p_{i+1\,j} + p_{i\,j-1} + p_{i\,j+1}) + (4+h^2)p_{ij} + h^2 y_{ij} &= h^2 z_{ij}, \\ p_{i-1\,j} - p_{i+1\,j} &= 0. \end{aligned}$$

Summing up the minus Laplacian stencil with the normal derivative, the ghost variables outside of Ω are eliminated. On the boundary, we obtain

$$\begin{aligned} -(2y_{i+1\,j} + y_{i\,j-1} + y_{i\,j+1}) + (4+h^2)y_{ij} - 2hu_{ij} &= h^2 g_{ij}, \\ -(2p_{i+1\,j} + p_{i\,j-1} + p_{i\,j+1}) + (4+h^2)p_{ij} + h^2 y_{ij} &= h^2 z_{ij}, \\ (\nu u_{ij} - p_{ij})(v_{ij} - u_{ij}) \geq 0 \quad \forall v_h \in U_{adh}. \end{aligned}$$

On the corners we consider the minus Laplacian stencil with the normal derivatives in both directions. The equations obtained in this way have the same structure as (5.76) and (5.77), and therefore the development of the collective smoothing iteration proceeds along the same lines as described above; see [59].

An example of application of the CSMG for solving boundary optimal control problems is the following. Consider the desired target given by

$$z(x_1, x_2) = (x_1^2 - x_2^2)\sin(\pi x_1)\sin(\pi x_2),$$

and let $g = 0$. We choose constraints given by $\underline{u} = -1$ and $\overline{u} = 1$ which are active in part of the boundary for $\nu \leq 10^{-6}$. The multigrid setting is the same as in previous sections. Results for this case are reported in Table 5.5.

Table 5.5. *Results of experiments with a boundary control problem;* 1025×1025 *mesh.*

ν	$\rho(y)$, $\rho(p)$	$\lvert y - z\rvert_0$	$\lvert r(y)\rvert_0$, $\lvert r(p)\rvert_0$
10^{-6}	0.05, 0.05	$8.09\,10^{-2}$	$1.7\,10^{-10}$, $2.9\,10^{-13}$
10^{-8}	0.14, 0.12	$8.09\,10^{-2}$	$3.7\,10^{-8}$, $2.9\,10^{-13}$
10^{-10}	0.28, 0.28	$8.09\,10^{-2}$	$4.7\,10^{-5}$, $9.9\,10^{-11}$
0	0.25, 0.26	$8.09\,10^{-2}$	$3.5\,10^{-5}$, $4.8\,10^{-11}$

In the presence of nonlinearities in the state equation it is not possible to solve explicitly the optimality system at the grid point level. Instead, a collective local Newton–Gauss–Seidel iteration can be applied, resulting in a CSMG scheme that also in this case provides convergence factors that are almost independent of ν and of the mesh size [48, 47, 57].

Consider the following nonlinear optimal control problem

$$\left\{ \begin{array}{rcll} \min J(y,u) & := & \frac{1}{2}||y-z||^2_{L^2(\Omega)} + \frac{\nu}{2}||u||^2_{L^2(\Omega)}, & \\ -\Delta y + G(y) & = & u + f & \text{in } \Omega, \\ y & = & 0 & \text{on } \partial\Omega, \end{array} \right. \tag{5.86}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

where U_{ad} is as in (5.72). Existence of solutions to (5.86) can be established under suitable conditions for various forms of the nonlinearity; see, e.g., [57, 148, 236]. We take $G \in C^\infty$ monotonically increasing function with $G(0) = 0$, but the following procedure remains effective for other nonlinearities as well.

Optimal solutions are characterized by the following optimality system

$$\begin{aligned} -\Delta y + G(y) &= u + g && \text{in } \Omega, \\ y &= 0 && \text{on } \partial\Omega, \\ -\Delta p + G'(y)\, p &= -(y - z) && \text{in } \Omega, \\ p &= 0 && \text{on } \partial\Omega, \\ (\nu u - p, v - u) &\geq 0 && \forall v \in U_{ad}. \end{aligned} \tag{5.87}$$

Similar to the linear case, we consider the discrete version of this optimality system at the grid point i, j and thus obtain the following system for the three scalar variables y_{ij}, p_{ij}, and u_{ij}. We have

$$-A_{ij} + 4\, y_{ij} + h^2\, G(y_{ij}) - h^2\, u_{ij} = 0, \tag{5.88}$$

$$-B_{ij} + 4\, p_{ij} + h^2\, G'(y_{ij})\, p_{ij} + h^2\, y_{ij} = 0, \tag{5.89}$$

$$(\nu u_{ij} - p_{ij})(v_{ij} - u_{ij}) \geq 0 \quad \forall v_h \in U_{adh}. \tag{5.90}$$

The inverse of the Jacobian for (5.88)–(5.89) is given by

$$J_{ij}^{-1} = \frac{1}{\det J_{ij}} \begin{pmatrix} 4 + h^2 G'(y_{ij}) & 0 \\ -h^2(1 + G''(y_{ij})\, p_{ij}) & 4 + h^2 G'(y_{ij}) \end{pmatrix}, \tag{5.91}$$

where $\det J_{ij} = (4 + h^2 G'(y_{ij}))(4 + h^2 G'(y_{ij}))$. Notice that also in the case of nonmonotone nonlinearities such that $G' < 0$, we can choose h sufficiently small to guarantee that $\det J_{ij} \neq 0$. Also notice that second-order necessary conditions for a minimum require that $(1 + G''(y_{ij})\, p_{ij}) \geq 0$. Hence, given u_{ij}, a local Newton update for the state and the adjoint variables $\hat{y}_{ij}$ and $\hat{p}_{ij}$ at i, j can be performed, and it is given by

$$\begin{pmatrix} \hat{y}_{ij} \\ \hat{p}_{ij} \end{pmatrix} = \begin{pmatrix} y_{ij} \\ p_{ij} \end{pmatrix} + J_{ij}^{-1} \begin{pmatrix} r_{ij}^y \\ r_{ij}^p \end{pmatrix}, \tag{5.92}$$

where $r_{ij}^y = A_{ij} - 4\, y_{ij} - h^2\, G(y_{ij}) + h^2\, u_{ij}$ and $r_{ij}^p = B_{ij} - 4\, p_{ij} - h^2\, G'(y_{ij})\, p_{ij} - h^2\, y_{ij}$ denote the residual of (5.88) and (5.89), respectively. Notice that r_{ij}^y depends explicitly on u_{ij}. This fact allows us to write $\hat{p}_{ij}$ as a function of u_{ij} as follows

$$\begin{aligned} \hat{p}_{ij}(u_{ij}) = p_{ij} &+ \frac{1}{\det J_{ij}} \left(-h^2(1 + G''(y_{ij})\, p_{ij})(A_{ij} - 4\, y_{ij} - h^2\, G(y_{ij})) \right) \\ &+ \frac{1}{\det J_{ij}} \left((4 + h^2 G'(y_{ij}))(B_{ij} - 4\, p_{ij} - h^2\, G'(y_{ij})\, p_{ij} - h^2\, y_{ij}) \right) \\ &- \frac{1}{\det J_{ij}} (1 + G''(y_{ij})\, p_{ij})\, h^4\, u_{ij}. \end{aligned} \tag{5.93}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Now to obtain first the update for u_{ij}, replace $\hat{p}_{ij}$ in the inequality constraint. From $\nu\tilde{u} - \hat{p}(\tilde{u}) = 0$ at i, j, we obtain the auxiliary variable

$$\begin{aligned}\tilde{u}_{ij} = &\left(\nu + \frac{(1+G''(y_{ij})\,p_{ij})h^4}{\det J_{ij}}\right)^{-1}\\ &\times\Bigg[p_{ij} + \frac{1}{\det J_{ij}}\left(-h^2(1+G''(y_{ij})\,p_{ij})(A_{ij} - 4\,y_{ij} + h^2\,G(y_{ij}))\right)\\ &+ \frac{1}{\det J_{ij}}\left((4+h^2G'(y_{ij}))(B_{ij} - 4\,p_{ij} - h^2\,G'(y_{ij})\,p_{ij} - h^2\,y_{ij})\right)\Bigg].\end{aligned}$$

This $\tilde{u}$ provides the update for the control in the case of no constraints. In the presence of constraints we apply the projection given by (5.82).

It should be clear that the CSMG multigrid procedure just described can be easily extended to solve the optimality system (5.87) with the fourth-order discretization discussed in Section 3.2. See [48].

A CSMG Scheme for State-Constrained Elliptic Control Problems

In state-constrained optimal control problems, bounds are given to the admissible set of values of the state variable. Consider the following

$$\begin{cases} \min J(y,u) := \frac{1}{2}\|y-z\|^2_{L^2(\Omega)} + \frac{\nu}{2}\|u\|^2_{L^2(\Omega)}, & \\ \Delta y + G(y) = u+g & \text{in } \Omega,\\ y = 0 & \text{on } \partial\Omega,\\ y_L \le y \le y_H & \text{a.e. in } \Omega, \end{cases} \tag{5.94}$$

where y_L and y_H are continuous functions. Existence and uniqueness of solutions to state-constrained semilinear elliptic optimal control problems depend on the given constraints and on the nonlinearity. For a nonempty solution set, uniqueness can be proved for sufficiently regular G such that the state operator is monotone; see, e.g., [207, 255, 339].

The solution approach to state-constrained optimal control problems through Lagrange multipliers associated with the state constraints leads to difficulties [254]. In particular, the fact that the Lagrange multipliers associated with the state constraints are only regular Borel measures prevents us from using classical approximation techniques. The remedy is to introduce appropriate regularization; see [207, 254] and the references given therein.

In the following, we consider the Lavrentiev regularization approach because it elegantly accommodates our framework. The Lavrentiev-type regularization consists in approximating the pointwise state constraints $y_L(x) \le y(x) \le y_H(x)$ with the following

$$y_L(x) \le y(x) - \lambda u(x) \le y_H(x) \quad \text{a.e. in } \Omega,$$

where $\lambda > 0$ is a small parameter. As a result, the associated Lagrange multipliers can be assumed to be functions in $L^2(\Omega)$; see, e.g., [254]. The following regularized state-constrained optimal control problem results

$$\begin{cases} \min J(y,u) := \frac{1}{2}||y-z||^2_{L^2(\Omega)} + \frac{\nu}{2}||u||^2_{L^2(\Omega)}, & \\ -\Delta y + G(y) = u+g, & \\ y = 0, & \\ y_L \le y - \lambda u \le y_H. & \end{cases} \tag{5.95}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Now, introduce the auxiliary variable $v = y - \lambda u$ (use $v = y + \lambda u$ if you have the minus Laplacian) and express the control function u in terms of $v \in L^2(\Omega)$. The regularized state-constrained optimal control problem becomes

$$\left\{ \begin{array}{rcl} \min J(y,v) & := & \frac{1}{2}||y-z||^2_{L^2(\Omega)} + \frac{\nu}{2\lambda^2}||y-v||^2_{L^2(\Omega)}, \\ -\Delta y + G(y) - y/\lambda + v/\lambda & = & g, \\ y & = & 0, \\ y_L \quad \leq v & \leq & y_H. \end{array} \right. \tag{5.96}$$

Notice that after the transformation, an optimal control problem is obtained having a "control-constrained" structure. The solution to (5.96) is characterized by the following optimality system

$$\begin{array}{rcll} -\Delta y + G(y) - y/\lambda + v/\lambda & = & g & \text{in } \Omega, \\ y & = & 0 & \text{on } \partial\Omega, \\ -\Delta p + G'(y)p - p/\lambda + (y-z) + \gamma(y-v) & = & 0 & \text{in } \Omega, \\ p & = & 0 & \text{on } \partial\Omega, \\ (p/\lambda - \gamma(y-v), t-v) & \geq & 0, & \end{array} \tag{5.97}$$

where $\gamma = \nu/\lambda^2$ and the inequality must hold for all $t \in V_{ad}$, and V_{ad} is defined by

$$V_{ad} = \{v \in L^2(\Omega) \,|\, y_L(x) \leq v(x) \leq y_H(x) \text{ a.e. in } \Omega\}.$$

Next, the construction of the CSMG smoother for the regularized state-constrained optimality system given by (5.97) is discussed. We use the finite difference framework of Chapter 3 and define the two constants

$$A_{ij} = -(y_{i-1\,j} + y_{i+1\,j} + y_{i\,j-1} + y_{i\,j+1}) - h^2 g_{ij}$$

and

$$B_{ij} = -(p_{i-1\,j} + p_{i+1\,j} + p_{i\,j-1} + p_{i\,j+1}) - h^2 z_{ij}.$$

Consider the following finite difference discretized state and adjoint equations:

$$A_{ij} - \alpha\, y_{ij} + h^2 G(y_{ij}) + (h^2/\lambda)\, v_{ij} = 0, \tag{5.98}$$

$$B_{ij} - \alpha\, p_{ij} + h^2 G'(y_{ij})\, p_{ij} + (1+\gamma)h^2 y_{ij} - \gamma\, h^2 v_{ij} = 0, \tag{5.99}$$

where $\alpha = (4 + h^2/\lambda)$. In addition, we have the inequality

$$(p_{ij}/\lambda - \gamma\, y_{ij} + \gamma\, v_{ij}) \cdot (t_{ij} - v_{ij}) \geq 0$$

for all $t \in V_{adh} = \{v \in L^2_h(\Omega_h) \,|\, y_L \leq v \leq y_H \text{ a.e. in } \Omega_h\}$.

Now consider the Jacobian of the system (5.98)–(5.99) with respect to y_{ij}, p_{ij}, that is,

$$J_{ij} = \begin{pmatrix} -\alpha + h^2 G'(y_{ij}) & 0 \\ (1+\gamma)h^2 + h^2 G''(y_{ij})\, p_{ij} & -\alpha + h^2 G'(y_{ij}) \end{pmatrix}.$$

Hence, the following local Newton update for y_{ij} and p_{ij} at (i,j) results

$$\begin{pmatrix} \hat{y}_{ij} \\ \hat{p}_{ij} \end{pmatrix} = \begin{pmatrix} y_{ij} \\ p_{ij} \end{pmatrix} + J_{ij}^{-1} \begin{pmatrix} r_{ij}(v_{ij}) \\ s_{ij}(v_{ij}) \end{pmatrix}, \tag{5.100}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

where $r_{ij}(v_{ij})$ and $s_{ij}(v_{ij})$ denote the residuals of (5.98) and (5.99), respectively. Both residuals depend explicitly on the control variable v_{ij}. Therefore the update above defines the values $\hat{y}_{ij}(v_{ij})$ and $\hat{p}_{ij}(v_{ij})$ as functions of v_{ij}.

Similarly to the control-constrained case, we now denote with $\tilde{v}_{ij}$ the solution to the unconstrained optimality condition equation

$$p_{ij}(v_{ij})/\lambda - \gamma\, y_{ij}(v_{ij}) + \gamma\, v_{ij} = 0.$$

It is given by $\tilde{v}_{ij} = N_{ij}/D_{ij}$, where

$$\begin{aligned} N_{ij} = &-(\lambda(\alpha B_{ij} + (1+\gamma)A_{ij}h^2 - A_{ij}\alpha\gamma\lambda - h^2(B_{ij} - A_{ij}\gamma\lambda + ((1+\gamma)h^2 - \alpha\gamma\lambda)y_{ij})G'(y_{ij}) \\ &- h^4\gamma\lambda y_{ij}G'(y_{ij})^2 + h^2A_{ij}\,p_{ij}G''(y_{ij}) - \alpha h^2 p_{ij}y_{ij}G''(y_{ij}) + h^2G(y_{ij})(h^2(1+\gamma) \\ &- \alpha\gamma\lambda + h^2\gamma\lambda G'(y_{ij}) + h^2 p_{ij}G''(y_{ij})))) \end{aligned}$$

and

$$\begin{aligned} D_{ij} = &\, h^4(1+\gamma) - 2\alpha h^2\gamma\lambda + \alpha^2\gamma\lambda^2 + 2h^2\gamma\lambda(h^2 - \alpha\lambda)G'(y_{ij}) \\ &+ h^4\gamma\lambda^2 G'(y_{ij})^2 + h^4\, p_{ij}\, G''(y_{ij}). \end{aligned}$$

Now recall that the update to v_{ij} must satisfy the constraints $y_L(x_{ij}) \le v_{ij} \le y_H(x_{ij})$. Therefore a feasible update is given by

$$v_{ij} = \begin{cases} y_{Hij} & \text{if} \quad \tilde{v}_{ij} \ge y_{Hij}, \\ \tilde{v}_{ij} & \text{if} \quad y_{Lij} < \tilde{v}_{ij} < y_{Hij}, \\ y_{Lij} & \text{if} \quad \tilde{v}_{ij} \le y_{Lij}. \end{cases} \tag{5.101}$$

Updates for the adjoint and state variables are obtained using (5.100), which defines the mappings $p_{ij} = \hat{p}_{ij}(v_{ij})$ and $y_{ij} = \hat{y}_{ij}(v_{ij})$, respectively.

The effectiveness of the resulting CSMG scheme can be seen in the following computation of a state-constrained optimal control problem. Take $z(x_1, x_2) = \sin(2\pi x_1)\sin(\pi x_2)$ and $y_L(x) = -1/2$ and $y_H(x) = 1/2$. Results for the choice $\nu = 10^{-7}$ and $\lambda = 10^{-3}$ are reported in Figures 5.9 and 5.10. In Figure 5.9, the constrained state solution is depicted. Convergence history is reported in Figure 5.10. We notice an increase of the value of the sum of the L^2-norm of the residuals during the first few iterations of the smoothing scheme. This behavior results from an increase of the residual of the state equation, while the residuals of the adjoint and control equations decrease monotonically. On the other hand, we observe typical convergence behavior of the multigrid scheme based on the proposed smoother.

The results reported in Table 5.6 show typical multigrid convergence factors that are mesh independent. These values are obtained choosing $\nu \approx \lambda^2$. With ν held fixed and decreasing λ the resulting convergence factors worsen.

Local Fourier Analysis: Linear Elliptic Case

In this section, we discuss the extension of the TG local Fourier analysis [85, 361, 340] in the case of the optimality system (5.73) assuming no constraints on the control; see [48, 61].

Consider a sequence of (infinite) grids, $G_k = \{(j_1h_k, j_2h_k),\ \boldsymbol{j} = (j_1, j_2) \in \mathbb{Z}^2\}$, and on these grids define the Fourier components

$$\phi_k(\boldsymbol{\theta}, \boldsymbol{j}) = e^{ij_1\theta_1}\, e^{ij_2\theta_2}.$$

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

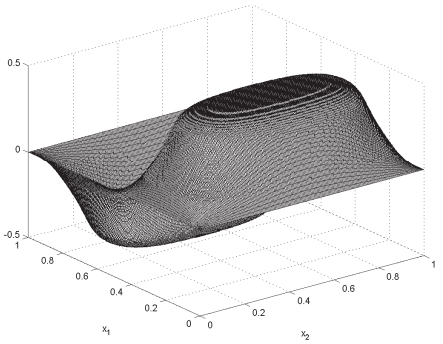

Figure 5.9. *State-constrained case. The optimal state for $\nu = 10^{-7}$ and $\lambda = 10^{-3}$. Reprinted with permission from A. Borzì, Smoothers for control- and state-constrained optimal control problems, Comput. Vis. Sci.,* 11(1) (2008), 59–66.

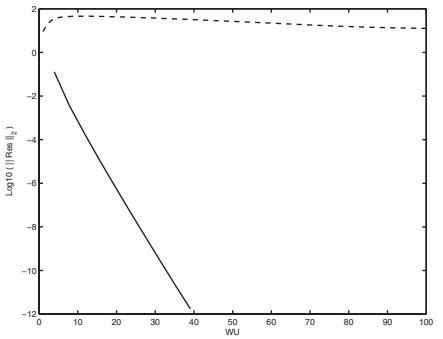

Figure 5.10. *State-constrained case. Convergence history for smoothing only (dashed line) and multigrid W(1,1)-cycle; $\nu = 10^{-7}$ and $\lambda = 10^{-3}$. Reprinted with permission from A. Borzì, Smoothers for control- and state-constrained optimal control problems, Comput. Vis. Sci.,* 11(1) (2008), 59–66.

Table 5.6. *Convergence factors choosing $\nu = \lambda^2$.*

Mesh	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$
257×257	0.06	0.06	0.07
513×513	0.07	0.07	0.08
1025×1025	0.07	0.07	0.07

For any LF $\boldsymbol{\theta} = (\theta_1, \theta_2) \in [-\pi/2, \pi/2)^2$, consider

$$\boldsymbol{\theta}^{(0,0)} := (\theta_1, \theta_2), \qquad \boldsymbol{\theta}^{(1,1)} := (\overline{\theta_1}, \overline{\theta_2}),$$
$$\boldsymbol{\theta}^{(1,0)} := (\overline{\theta_1}, \theta_2), \qquad \boldsymbol{\theta}^{(0,1)} := (\theta_1, \overline{\theta_2}),$$

where

$$\overline{\theta_j} = \begin{cases} \theta_j + \pi \text{ if } \theta_j < 0, \\ \theta_j - \pi \text{ if } \theta_j \geq 0. \end{cases}$$

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

We have $\phi(\boldsymbol{\theta}^{(0,0)},\cdot) = \phi(\boldsymbol{\theta}^{(1,1)},\cdot) = \phi(\boldsymbol{\theta}^{(1,0)},\cdot) = \phi(\boldsymbol{\theta}^{(0,1)},\cdot)$ for $\boldsymbol{\theta}^{(0,0)} \in [-\pi/2,\pi/2)^2$ and $(x_1,x_2) \in G_{k-1}$. That is, we have a quadruple of distinct Fourier components that coincide (aliases) on G_{k-1} with the LF component $\phi(\boldsymbol{\theta}^{(0,0)},\cdot)$.

Denote with $\boldsymbol{\alpha} = (\alpha_1,\alpha_2)$ and consider $\boldsymbol{\alpha} \in \{(0,0),(1,1),(1,0),(0,1)\}$; then on G_{k-1} we have $\phi_k(\boldsymbol{\theta}^{\boldsymbol{\alpha}},\mathbf{x}) = \phi_{k-1}(2\boldsymbol{\theta}^{(0,0)},\mathbf{x})$. The four components $\phi_k(\boldsymbol{\theta}^{\boldsymbol{\alpha}},\cdot)$ are called harmonics. Their span is denoted with

$$E_k^{\theta} = \text{span}[\phi_k(\boldsymbol{\theta}^{\boldsymbol{\alpha}},\cdot) : \boldsymbol{\alpha} \in \{(0,0),(1,1),(1,0),(0,1)\}].$$

The purpose of this analysis is to investigate the action of the smoothing and coarse-grid correction operators on couples (ϵ_y,ϵ_p) defined by

$$\epsilon_y(\boldsymbol{j}) = \sum_{\boldsymbol{\alpha},\boldsymbol{\theta}} Y_{\boldsymbol{\alpha},\boldsymbol{\theta}}\,\phi_k(\boldsymbol{\theta}^{\boldsymbol{\alpha}},\boldsymbol{j}) \quad \text{and} \quad \epsilon_p(\boldsymbol{j}) = \sum_{\boldsymbol{\alpha},\boldsymbol{\theta}} P_{\boldsymbol{\alpha},\boldsymbol{\theta}}\,\phi_k(\boldsymbol{\theta}^{\boldsymbol{\alpha}},\boldsymbol{j}).$$

Here (ϵ_y,ϵ_p) represent the error functions for y_h and p_h and $W_{\boldsymbol{\alpha},\boldsymbol{\theta}} = (Y_{\boldsymbol{\alpha},\boldsymbol{\theta}}, P_{\boldsymbol{\alpha},\boldsymbol{\theta}})$ denote the corresponding Fourier coefficients. With this decomposition of the error, the action of one smoothing step can be expressed as $W^{(1)}_{\boldsymbol{\alpha},\boldsymbol{\theta}} = \hat{S}(\boldsymbol{\theta})\,W^{(0)}_{\boldsymbol{\alpha},\boldsymbol{\theta}}$, where $\hat{S}(\boldsymbol{\alpha},\boldsymbol{\theta})$ is the Fourier symbol [340] of the smoothing operator. To determine $\hat{S}(\boldsymbol{\alpha},\boldsymbol{\theta})$, recall that the functions $\phi_k(\boldsymbol{\theta}^{\boldsymbol{\alpha}},\mathbf{x})$ are eigenfunctions of any discrete operator described by a difference stencil on the G_k grid. Therefore we have $S_k\phi_k(\boldsymbol{\theta}^{\boldsymbol{\alpha}},\boldsymbol{j}) = \hat{S}_k(\boldsymbol{\theta})\phi_k(\boldsymbol{\theta},\boldsymbol{j})$; that is, the symbol of S_k is its (formal) eigenvalue.

Now, consider the collective Gauss–Seidel (CGS) step applied to the following optimality system obtained with the replacement $u_h = p_h/\nu$, that is,

$$-\Delta_h y_h - p_h/\nu = g_h, \tag{5.102}$$

$$-\Delta_h p_h + y_h = z_h. \tag{5.103}$$

In this case, one smoothing step at $\mathbf{x}$ corresponds to an update which sets the residuals at $\mathbf{x}$ equal to zero. In terms of Fourier modes $\boldsymbol{\theta}$, $\hat{S}_k(\boldsymbol{\theta})$ is given by

$$\hat{S}_k(\boldsymbol{\theta}) = \begin{bmatrix} -(e^{-i\theta_1}+e^{-i\theta_2}-4) & -h_k^2/\nu \\ h_k^2 & -(e^{-i\theta_1}+e^{-i\theta_2}-4) \end{bmatrix}^{-1} \times \begin{bmatrix} (e^{i\theta_1}+e^{i\theta_2}) & 0 \\ 0 & (e^{i\theta_1}+e^{i\theta_2}) \end{bmatrix}, \tag{5.104}$$

which represents the Fourier symbol of the smoothing operator.

The smoothing factor of S_k measures the action of this iteration on the HF error components and can be defined as follows

$$\mu(S_k) = \sup\left\{|r(\hat{S}_k(\boldsymbol{\theta}))| : \boldsymbol{\theta} \in \{\boldsymbol{\theta}^{(1,1)},\boldsymbol{\theta}^{(1,0)},\boldsymbol{\theta}^{(0,1)}\}\right\}, \tag{5.105}$$

where r denotes the spectral radius. Notice that by local Fourier analysis the problem of computing the smoothing factor is reduced to that of determining the spectral radius of $\hat{S}(\boldsymbol{\theta})$, a 2×2 matrix. This task may be performed using any symbolic package. Notice that later, alternative equivalent formulation of the smoothing factor will be given.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

The next step is to construct the Fourier symbol of the TG coarse-grid correction operator given by

$$\widehat{CG}_k^{k-1}(\boldsymbol{\theta}) = [\hat{I}_k - \hat{I}_{k-1}^k(\boldsymbol{\theta})(\widehat{A}_{k-1}(2\boldsymbol{\theta}))^{-1}\hat{I}_k^{k-1}(\boldsymbol{\theta})\widehat{A}_k(\boldsymbol{\theta})].$$

The Fourier symbol of the coarse-grid operator $\widehat{A}_{k-1}(\boldsymbol{\theta})$ is

$$\begin{bmatrix} -\frac{2(\cos(2\theta_1)+\cos(2\theta_2))-4}{h_{k-1}^2} & -1/\nu \\ 1 & -\frac{2(\cos(2\theta_1)+\cos(2\theta_2))-4}{h_{k-1}^2} \end{bmatrix},$$

and similarly one constructs $\widehat{A}_k(\boldsymbol{\theta})$ corresponding to the four harmonics, that is,

$$\begin{bmatrix} l(\boldsymbol{\theta}^{(0,0)}) & 0 & 0 & 0 & -1/\nu & 0 & 0 & 0 \\ 0 & l(\boldsymbol{\theta}^{(1,1)}) & 0 & 0 & 0 & -1/\nu & 0 & 0 \\ 0 & 0 & l(\boldsymbol{\theta}^{(1,0)}) & 0 & 0 & 0 & -1/\nu & 0 \\ 0 & 0 & 0 & l(\boldsymbol{\theta}^{(0,1)}) & 0 & 0 & 0 & -1/\nu \\ 1 & 0 & 0 & 0 & l(\boldsymbol{\theta}^{(0,0)}) & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & l(\boldsymbol{\theta}^{(1,1)}) & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & l(\boldsymbol{\theta}^{(1,0)}) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & l(\boldsymbol{\theta}^{(0,1)}) \end{bmatrix},$$

where

$$l(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = -\frac{2(\cos(\theta_1^{\alpha_1}) + \cos(\theta_2^{\alpha_2})) - 4}{h_k^2}.$$

The Fourier symbol of restriction operator is

$$\hat{I}_k^{k-1}(\boldsymbol{\theta}) = \begin{bmatrix} I(\boldsymbol{\theta}^{(0,0)}) & I(\boldsymbol{\theta}^{(1,1)}) & I(\boldsymbol{\theta}^{(1,0)}) & I(\boldsymbol{\theta}^{(0,1)}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I(\boldsymbol{\theta}^{(0,0)}) & I(\boldsymbol{\theta}^{(1,1)}) & I(\boldsymbol{\theta}^{(1,0)}) & I(\boldsymbol{\theta}^{(0,1)}) \end{bmatrix},$$

where

$$I(\boldsymbol{\theta}^{\boldsymbol{\alpha}}) = \frac{1}{4}(1+\cos(\theta_1^{\alpha_1}))(1+\cos(\theta_2^{\alpha_2})).$$

For the prolongation operator we have $\hat{I}_{k-1}^k(\boldsymbol{\theta}) = \hat{I}_k^{k-1}(\boldsymbol{\theta})^\top$. Finally, the symbol of the TG method with ν_1 presmoothing steps and ν_2 postsmoothing steps is given by

$$\widehat{TG}_k^{k-1}(\boldsymbol{\theta}) = \hat{S}_k(\boldsymbol{\theta})^{\nu_2}\,\widehat{CG}_k^{k-1}(\boldsymbol{\theta})\,\hat{S}_k(\boldsymbol{\theta})^{\nu_1}.$$

This 8×8 matrix corresponds to the pairs $(Y_{\boldsymbol{\alpha},\boldsymbol{\theta}}, P_{\boldsymbol{\alpha},\boldsymbol{\theta}})$, $\boldsymbol{\alpha}\in\{(0,0),(1,1),(1,0),(0,1)\}$. In this framework the TG convergence factor is defined as follows

$$\rho(TG_k^{k-1}) = \sup\left\{r\left(\widehat{TG}_k^{k-1}(\boldsymbol{\theta})\right) : \boldsymbol{\theta}\in[-\pi/2,\pi/2)^2\right\}. \tag{5.106}$$

In Table 5.7, values of $\mu(S_k)$ and of $\rho(TG_k^{k-1})$ corresponding to the setting $h_k = 1/64$ and $\nu\in\{10^{-4},10^{-8}\}$ are reported. These results show robustness of the multigrid solver with respect to values of ν and suggest mesh independence and typical multigrid efficiency.

We report in Table 5.8 the values of $\rho(TG_k^{k-1})$ and that of $\mu(\mathcal{S}_k)^{m_1+m_2}$ obtained with the TG analysis with the forward Gauss–Seidel smoother. For comparison, the observed

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.7. *Convergence factors and smoothing factors obtained with local Fourier analysis; $h = 1/64$ and different ν.*

(m_1, m_2)	ν	μ	ρ
(1,1)	10^{-4}	0.50	0.20
(2,2)	10^{-4}	0.50	0.08
(1,1)	10^{-8}	0.55	0.27
(2,2)	10^{-8}	0.55	0.12

Table 5.8. *Estimated and observed convergence factors (averages).*

(m_1, m_2)	$\mu^{m_1+m_2}$	ρ	ρ_{exp}
(1,1)	0.25	0.25	0.30
(2,1)	0.125	0.12	0.12
(2,2)	0.06	0.08	0.08
(3,2)	0.03	0.06	0.06
(3,3)	0.01	0.05	0.05

values ρ_{exp} of convergence factor defined as the "asymptotic" value of the ratio between the discrete L^2-norms of residuals resulting from two successive multigrid cycles on the finest mesh are reported. Notice that the values reported in Table 5.8 are typical of the standard Poisson model problem. These values have been obtained considering the mesh size value h ranging in the interval $[0.01, 0.25]$ corresponding to the interval of mesh sizes used in the multigrid code. The value of the weight ν has been taken in the interval $[10^{-6}, 1]$.

CSMG Convergence Theory

TG local Fourier analysis of multigrid schemes applied to optimality systems provides sharp convergence estimates at the cost of simplifying assumptions. On the other hand, the multigrid theory provided in this section does not require special assumptions on the boundary; it applies to polygonal domains and guarantees convergence of the multigrid method to weak solutions of the optimality system.

We discuss convergence of the CSMG scheme in the framework of [61, 78, 82]. For a related theoretical framework see [308].

Consider the following optimality system

$$-\Delta_h y_h - p_h/\nu = g_h, \tag{5.107}$$

$$-\Delta_h p_h + y_h = z_h. \tag{5.108}$$

The starting point for this analysis is given in Section 5.2.4, where we have presented the theoretical framework of [82] for the scalar Poisson problem

$$\begin{aligned} -\Delta y &= f \text{ in } \Omega, \\ y &= 0 \text{ on } \partial\Omega. \end{aligned} \tag{5.109}$$

After discretization, this problem becomes

$$\hat{A}_k y_k = f_k. \tag{5.110}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Let $\hat{P}_{k-1} : V_k \to V_{k-1}$ (resp., $I_k^{k-1} : V_k \to V_{k-1}$) be the $\hat{A}_k$ (resp., L_k^2) projections defined by $(\hat{A}_{k-1}\hat{P}_{k-1}u, v)_{k-1} = (\hat{A}_k u, I_{k-1}^k v)_k$ (resp., $(I_k^{k-1}u, v)_{k-1} = (u, I_{k-1}^k v)_k$) for all $u \in V_k$ and $v \in V_{k-1}$. Let $\hat{R}_k : V_k \to V_k$ be an iteration operator. Then the V-cycle multigrid algorithm to solve (5.110) in recursive form is given by Algorithm 5.3. For this algorithm we proved the following convergence result (Theorem 5.7). (As in the scalar case, for the purpose of a simplified analysis, we take $\nu_1 = 1$ and $\nu_2 = 0$.)

Theorem 5.15. *Let $\hat{R}_k$ satisfy* (5.26) *and* (5.27) *for $k > 1$. Then there exists a positive constant $\delta < 1$ such that*

$$(\hat{A}_k \hat{M}_k u, u)_k \le \hat{\delta}\,(\hat{A}_k u, u)_k \quad \forall u \in V_k,$$

where $\hat{M}_k = I_k - \hat{B}_k \hat{A}_k$.

To investigate convergence of the multigrid scheme applied to the optimality system, we first consider the decoupled symmetric system as follows

$$\begin{array}{rcl} -\nu\Delta y &=& \nu g \text{ in } \Omega, \\ y &=& 0 \text{ on } \partial\Omega, \\ -\Delta p &=& z \text{ in } \Omega, \\ p &=& 0 \text{ on } \partial\Omega. \end{array} \tag{5.111}$$

This system is exactly two copies of the Poisson problem; hence the multigrid convergence theory for this system inherits the properties of the scalar case. In fact, if we define

$$\hat{\mathcal{A}}_k = \begin{pmatrix} \nu\hat{A}_k & 0 \\ 0 & \hat{A}_k \end{pmatrix}, \tag{5.112}$$

and analogously $\hat{\mathcal{B}}_k$, $\hat{\mathcal{A}}_k$, etc., as the system counterparts of $\hat{B}_k$, $\hat{A}_k$, etc., then the multigrid algorithm has exactly the same form as Algorithm 5.3 with $\hat{\mathcal{B}}_k$, $\hat{\mathcal{A}}_k$, etc., replacing $\hat{B}_k$, $\hat{A}_k$, etc. As a consequence we have the following theorem. Let $\mathbf{w}_k = (y_k, p_k) \in V_k \times V_k =: \mathcal{V}_k$.

Theorem 5.16. *Under the assumption of Theorem* 5.7, *there exists a positive constant $\delta < 1$ such that*

$$(\hat{\mathcal{A}}_k \hat{\mathcal{M}}_k \mathbf{w}, \mathbf{w})_k \le \delta\,(\hat{\mathcal{A}}_k \mathbf{w}, \mathbf{w})_k, \tag{5.113}$$

where δ has the same form as in Theorem 5.7.

To analyze the optimality system, we define

$$\mathcal{A}_k = \hat{\mathcal{A}}_k + d_k,$$

where

$$d_k = \begin{pmatrix} 0 & -I_k \\ I_k & 0 \end{pmatrix}.$$

We note that

$$|(d_k \mathbf{w}, \mathbf{w}')| \le C\,|\mathbf{w}|_0\,|\mathbf{w}'|_0 \tag{5.114}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

for some constant C. Now, the multigrid algorithm corresponding to this nonsymmetric problem has exactly the same recursive form as (5.25) with $\mathcal{B}_k$, $\mathcal{A}_k$, etc., replacing B_k, A_k, etc., and thus

$$\mathcal{M}_k = \mathcal{I}_k - \mathcal{B}_k\mathcal{A}_k = [\mathcal{I}_k - \mathcal{I}_{k-1}^k\mathcal{P}_{k-1} + \mathcal{I}_{k-1}^k(\mathcal{I}_{k-1} - \mathcal{B}_{k-1}\mathcal{A}_{k-1})\mathcal{P}_{k-1}]\mathcal{S}_k, \tag{5.115}$$

where $\mathcal{I}_k$ is the identity operator on $\mathcal{V}_k$.

Next, let $\mathcal{S}_k$ and $\hat{\mathcal{S}}_k$ represent the CGS smoothing and two copies of the scalar Gauss–Seidel smoothing, respectively. Based on a subspace decomposition of $\mathcal{V}_k = \sum_{i=1}^{\ell}\mathcal{V}_k^i$ one can prove the following two lemmas [61].

Lemma 5.17. *There exists some constant $C_{\mathcal{S}}$ independent of k such that*

$$|(\hat{\mathcal{A}}_k(\mathcal{S}_k - \hat{\mathcal{S}}_k)\mathbf{w},\mathbf{v})_k| \le C_{\mathcal{S}}\, h_k\, |\mathbf{w}|_1\, |\mathbf{v}|_1 \tag{5.116}$$

for all $\mathbf{w},\mathbf{v} \in \mathcal{V}_k$.

Lemma 5.18. *The following inequalities hold:*

$$|(\hat{\mathcal{A}}_{k-1}(\hat{\mathcal{P}}_{k-1} - \mathcal{P}_{k-1})\mathbf{w},\mathbf{v})_{k-1}| \le C_{\mathcal{P}}\, h_{k-1}\, |\mathbf{w}|_1|\mathbf{v}|_1 \text{ for } \mathbf{w} \in \mathcal{V}_k, \mathbf{v} \in \mathcal{V}_{k-1} \tag{5.117}$$

and

$$|(\hat{\mathcal{A}}_k(\mathcal{I}_k - \mathcal{I}_{k-1}^k\mathcal{P}_{k-1})\mathbf{w},\mathbf{v})_k| \le C_I\, h_k\, |\mathbf{w}|_1\, |\mathbf{v}|_1 \text{ for } \mathbf{w} \in \mathcal{V}_k, \mathbf{v} \in \mathcal{V}_k, \tag{5.118}$$

where $C_{\mathcal{P}}$ and C_I are some constants independent of k.

We assume that I_{k-1}^k and I_k^{k-1} represent the bilinear interpolation and full-weighting restriction operators. The prolongation operator satisfies the following conditions [82]:

$$(\hat{A}_k I_{k-1}^k u_{k-1}, I_{k-1}^k u_{k-1})_k \le (\hat{A}_{k-1}u_{k-1}, u_{k-1})_{k-1} \qquad \forall u_{k-1} \in V_{k-1}, \tag{5.119}$$

$$(I_{k-1}^k u_{k-1}, I_{k-1}^k u_{k-1})_k \le (u_{k-1}, u_{k-1})_{k-1} \qquad \forall u_{k-1} \in V_{k-1}. \tag{5.120}$$

As a consequence of this lemma we have the following

$$(\mathcal{A}_k\mathcal{I}_{k-1}^k\mathbf{w}_{k-1}, \mathcal{I}_{k-1}^k\mathbf{w}_{k-1})_k \le (\mathcal{A}_{k-1}\mathbf{w}_{k-1}, \mathbf{w}_{k-1})_{k-1} \tag{5.121}$$

for all $\mathbf{w}_{k-1} = (u_{k-1}, v_{k-1}) \in \mathcal{V}_{k-1}$. Now we prove the following theorem.

Theorem 5.19. *There exist positive constants h_0 and $\tilde{\delta} < 1$ such that for all $h_1 < h_0$ we have*

$$(\hat{\mathcal{A}}_k\mathcal{M}_k\mathbf{w},\mathbf{w})_k \le \tilde{\delta}\,(\hat{\mathcal{A}}_k\mathbf{w},\mathbf{w})_k \quad \forall \mathbf{w} \in \mathcal{V}_k,$$

where $\tilde{\delta} = \delta + Ch_1$ and δ is as in Theorem 5.16.

Proof. Denoting the operator norm $\|\cdot\|_{\hat{\mathcal{A}}_k}$ by $\|\cdot\|$, we show that $\|\mathcal{M}_k - \hat{\mathcal{M}}_k\| \le c_k h_1$, where c_k is uniformly bounded. The error operator $\mathcal{M}_k$ can be written as

$$\mathcal{M}_k = (\mathcal{I}_k - \mathcal{I}_{k-1}^k\mathcal{B}_{k-1}\mathcal{A}_{k-1}\mathcal{P}_{k-1})\mathcal{S}_k,$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

and $\hat{\mathcal{M}}_k$ has similar representation. We compare the error operators and write their difference as

$$\begin{aligned}\mathcal{M}_k - \hat{\mathcal{M}}_k = &(\mathcal{I}_k - \mathcal{I}_{k-1}^k \mathcal{B}_{k-1}\mathcal{A}_{k-1}\mathcal{P}_{k-1})(\mathcal{S}_k - \hat{\mathcal{S}}_k) \\ &- \mathcal{I}_{k-1}^k \mathcal{B}_{k-1}\mathcal{A}_{k-1}(\mathcal{P}_{k-1} - \hat{\mathcal{P}}_{k-1})\hat{\mathcal{S}}_k + \mathcal{I}_{k-1}^k(\mathcal{M}_{k-1} - \hat{\mathcal{M}}_{k-1})\hat{\mathcal{P}}_{k-1}\hat{\mathcal{S}}_k.\end{aligned}$$

Thus in terms of the operator norm, we have

$$\begin{aligned}\|\mathcal{M}_k - \hat{\mathcal{M}}_k\| \le &\|\mathcal{I}_k - \mathcal{I}_{k-1}^k \mathcal{B}_{k-1}\mathcal{A}_{k-1}\mathcal{P}_{k-1}\|\,\|\mathcal{S}_k - \hat{\mathcal{S}}_k\| \\ &+ \|\mathcal{B}_{k-1}\mathcal{A}_{k-1}\|\,\|\mathcal{P}_{k-1} - \hat{\mathcal{P}}_{k-1}\|\,\|\hat{\mathcal{S}}_k\| \\ &+ \|\mathcal{M}_{k-1} - \hat{\mathcal{M}}_{k-1}\|\,\|\hat{\mathcal{P}}_{k-1}\hat{\mathcal{S}}_k\|.\end{aligned} \tag{5.122}$$

Let us make the induction hypothesis: $\|\mathcal{M}_{k-1} - \hat{\mathcal{M}}_{k-1}\| \le c_{k-1}h_1$, where c_{k-1} is a constant to be defined below. By the triangle inequality and Theorem 5.16,

$$\|\mathcal{M}_{k-1}\| \le \delta + c_{k-1}h_1 \tag{5.123}$$

and

$$\|\mathcal{B}_{k-1}\mathcal{A}_{k-1}\| \le 1 + \delta + c_{k-1}h_1. \tag{5.124}$$

Using the induction hypothesis, (5.119), Lemma 5.17, and Lemma 5.18, we have

$$\begin{aligned}&\|\mathcal{I}_k - \mathcal{I}_{k-1}^k \mathcal{B}_{k-1}\mathcal{A}_{k-1}\mathcal{P}_{k-1}\| \\ &\quad\le \|\mathcal{I}_k - \mathcal{I}_{k-1}^k \mathcal{P}_{k-1}\| + \|\mathcal{I}_{k-1} - \mathcal{B}_{k-1}\mathcal{A}_{k-1}\|\,\|\mathcal{P}_{k-1}\| &(5.125)\\ &\quad\le C_I h_{k-1} + \|\mathcal{M}_{k-1}\|(1 + C_{\mathcal{P}}h_{k-1}) &(5.126)\\ &\quad\le C_I(h_{k-1} + \delta + c_{k-1}h_1), &(5.127)\end{aligned}$$

where we assumed C_I sufficiently large so that $1 + C_{\mathcal{P}}h_{k-1} \le C_I$. To prove the second inequality (5.126) we used the fact that $\|\hat{\mathcal{P}}_{k-1}\| \le 1$ and the chain of inequalities $\|\mathcal{P}_{k-1}\| \le \|\hat{\mathcal{P}}_{k-1}\| + \|\mathcal{P}_{k-1} - \hat{\mathcal{P}}_{k-1}\| \le 1 + C_{\mathcal{P}}h_{k-1}$.

Collecting (5.122)–(5.124), and using (5.119), Lemma 5.17, Lemma 5.18, and (5.125)–(5.127), we see that

$$\begin{aligned}\|\mathcal{M}_k - \hat{\mathcal{M}}_k\| &\le C_I C_{\mathcal{S}}(h_{k-1} + \delta + c_{k-1}h_1)h_k \\ &\quad + C_{\mathcal{P}}(1 + \delta + c_{k-1}h_1)h_{k-1} + c_{k-1}h_1 \\ &\le \left(\frac{C_I C_{\mathcal{S}}}{2} + C_{\mathcal{P}}\right) h_{k-1}(1 + \delta + c_{k-1}h_1) + c_{k-1}h_1\end{aligned}$$

for all k.

Now let $\hat{C} := \frac{C_I C_{\mathcal{S}}}{2} + C_{\mathcal{P}}$ and define

$$c_k := c_{k-1} + \hat{C}h_1^{-1}h_{k-1}(1 + \delta + c_{k-1}h_1). \tag{5.128}$$

To see that the sequence c_k is uniformly bounded in k, one notes that $c_j \le c_k$ for $j \le k$ and

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

hence

$$\begin{aligned} c_k &= c_{k-1} + \hat{C} h_1^{-1}(1+\delta+c_{k-1}h_1)h_{k-1} \\ &= c_1 + \hat{C} h_1^{-1} \sum_{j=2}^{k} (1+\delta+c_{j-1}h_1)h_{j-1} \\ &\le c_1 + \hat{C} h_1^{-1} \sum_{j=2}^{k} (1+\delta+c_k h_1)h_{j-1} \\ &\le c_1 + 2\hat{C}(1+\delta) + 2\hat{C}h_1 c_k. \end{aligned}$$

Now move the c_k term to the left to get

$$c_k \le (c_1 + 2\hat{C}(1+\delta))/(1-2\hat{C}h_1),$$

provided that h_1 is small enough. Therefore, if the coarsest grid is sufficiently fine, we have $\tilde{\delta} = \delta + Ch_1 < 1$. □

The constants in (5.116), (5.117), and (5.118) depend on the features of the optimality system such as, for example, nonsymmetry. They account for the induction hypothesis where the coarsest mesh size, h_1, enters the analysis and results in the estimate $\tilde{\delta} = \delta + C h_1$. The requirement for a sufficiently small h_1 has no correspondence to our numerical experience (using CGS). However, the estimate of Theorem 5.19 states that, for sufficiently small h_1, we have $\tilde{\delta} \approx \delta$; that is, the convergence factor of the multigrid method applied to the optimality system is close to the convergence factor of the multigrid scheme applied to the scalar Poisson problem. This fact agrees with our numerical experience.

5.7.2 Algebraic Multigrid Methods for Optimality Systems

In this section, we describe the extension of the AMG scheme discussed in Section 5.2.5 in order to solve optimality systems. In particular, we discuss the AMG solution of the following system of n_e linear elliptic equations

$$-\sum_{k=1}^{d} \frac{\partial}{\partial x_k}\left(d_k \frac{\partial u^{(l)}}{\partial x_k}\right) + \sum_{k=1}^{d} c_k \frac{\partial u^{(l)}}{\partial x_k} + \sum_{p=1}^{n_e} b^{(l,p)} u^{(p)} = f^{(l)}, \qquad l = 1,\ldots,n_e, \tag{5.129}$$

where the functions $d_k = d_k(\mathbf{x})$, $c_k = c_k(\mathbf{x})$, and $b^{(l,p)} = b^{(l,p)}(\mathbf{x})$ are in $L^\infty(\Omega)$ and represent the diffusion coefficients, the convection coefficients, and the reaction coefficients, respectively. The right-hand side is given by $f^{(l)} \in L_2(\Omega)$. We assume that on the boundary $\partial\Omega$, the solution is subject to general Robin boundary conditions.

Let us denote with $\hat{A}$ the differential part of (5.129) and denote with B the coupling operator. After discretization, the following system of algebraic equations is obtained

$$\hat{A}_k u_k^{(l)} + \sum_{p=1}^{n_e} B_k^{(l,p)} u_k^{(p)} = F_k^{(l)}, \qquad l = 1,\ldots,n_e, \tag{5.130}$$

where the boundary conditions enter the definition of $\hat{A}_k$ and of $F_k^{(l)}$. For $k = 1$, system (5.130) denotes the problem to be solved, which is also the "finest" algebraic problem in

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

the AMG solution process. In order to solve (5.130), the algebraic multigrid algorithm constructs a hierarchy of coarser problems denoted by (5.130) with $k = 2, \ldots, L$, where L is the index of the coarsest level (opposite to the geometrical multigrid case).

It is convenient to represent (5.130) in block form as follows

$$\hat{\mathcal{A}}_k \mathbf{u}_k + \mathcal{B}_k \mathbf{u}_k = \boldsymbol{\mathcal{F}}_k, \tag{5.131}$$

where $\hat{\mathcal{A}}_k$ and $\mathcal{B}_k$ have $\hat{A}_k$ and B_k as constitutive blocks. Here, we denote with $\mathbf{u}_k = (u_k^{(l)})_{l=1,n_e}$, with $\boldsymbol{\mathcal{F}}_k = (\mathcal{F}_k^{(l)})_{l=1,n_e}$, and with N_k the total number of (variables) points at level k.

As a smoother we use the CGS method based on a blockwise ordering of the unknowns and block splitting of $\mathcal{A}_k = \hat{\mathcal{A}}_k + \mathcal{B}_k$, that is, $\mathcal{A}_k = \mathcal{D}_k - \mathcal{L}_k - \mathcal{U}_k$, where $\mathcal{D}_k$ is the block-diagonal matrix, $\mathcal{L}_k$ is the lower block-triangular matrix, and $\mathcal{U}_k$ is the upper block-triangular matrix. Then, the CGS scheme can be written as

$$\mathbf{u}_{k,i}^{new} = \mathbf{u}_{k,i}^{old} + \mathcal{D}_{k,i}^{-1} \mathcal{R}_{k,i}, \quad i = 1, 2, \ldots, N_k, \tag{5.132}$$

where $\mathcal{R}_{k,i} = (\mathcal{R}_{k,i}^{(l)})_{l=1,n_e}$ denotes the (dynamic) residuals of the n_e equations at the point i immediately before the relaxation step and the block $\mathcal{D}_{k,i}$ in the case of $n_e = 2$ is given by

$$\mathcal{D}_i = \begin{pmatrix} a_{ii} + b_{ii}^{1,1} & b_{ii}^{1,2} \\ b_{ii}^{2,1} & a_{ii} + b_{ii}^{2,2} \end{pmatrix}. \tag{5.133}$$

For the coarse-grid correction a system of coarse algebraic problems is constructed at level $k+1$,

$$\hat{\mathcal{A}}_{k+1} \mathbf{e}_{k+1} + \mathcal{B}_{k+1} \mathbf{e}_{k+1} = \mathcal{I}_k^{k+1} \mathcal{R}_k, \tag{5.134}$$

where $\mathbf{e}_{k+1}$ aims to represent, on the coarse level, the error $\mathbf{e}_k$ on the next finer level. The operator $\mathcal{I}_k^{k+1}$ restricts the residual computed at level k to the level $k+1$. It represents the action of the AMG restriction operator I_k^{k+1} applied n_e times. The coarse matrix of coefficient $\hat{\mathcal{A}}_{k+1}$ and the mass matrix associated with any of the $\mathcal{B}_{k+1}^{(l,p)}$ terms are defined by the Galerkin formula

$$\hat{\mathcal{A}}_{k+1} = \mathcal{I}_k^{k+1} \hat{\mathcal{A}}_k \mathcal{I}_{k+1}^k \quad \text{and} \quad \mathcal{B}_{k+1} = \mathcal{I}_k^{k+1} \mathcal{B}_k \mathcal{I}_{k+1}^k.$$

Once the coarse grid problem is solved, a new approximation to the error at level k is obtained, $\mathcal{I}_{k+1}^k \mathbf{e}_{k+1}$, and the coarse-grid correction follows:

$$\mathbf{u}_k^{new} = \mathbf{u}_k + \mathcal{I}_{k+1}^k \mathbf{e}_{k+1}, \tag{5.135}$$

where $\mathcal{I}_{k+1}^k$ is an interpolation operator. It represents the action of I_{k+1}^k applied n_e times. Here $\mathbf{u}_k$ represents the current approximation at level k as it was obtained by the smoothing process and before coarsening.

Now, notice that the AMG scheme for differential systems illustrated above can be efficiently implemented in the case where the blocks $\hat{A}_k$ are all equal to a "master" block $\tilde{A}_k$ or some of them are the transpose of this block, $\hat{A}_{k'} = \tilde{A}_k^\top$. In particular, this configuration occurs in the case of optimality systems where the discretization of the state and adjoint equations may result in blocks that are the transpose of each other. It is clear that in these

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

cases an AMG scheme can be developed that requires one to store only the master block and the restriction operator I_k^{k+1}, constructed based only on $\hat{A}_k$. This is done in [51] for the case of elliptic control problems with jumping coefficients and in [52], where the following convection-diffusion optimality system is considered. We have

$$-\sum_{k=1}^{d} \frac{\partial}{\partial x_k}\left(d_k \frac{\partial y}{\partial x_k}\right) + \sum_{k=1}^{d} c_k \frac{\partial y}{\partial x_k} - \frac{1}{\nu} p = g \quad \text{in } \Omega, \tag{5.136}$$

$$\alpha \frac{\partial y}{\partial \nu} + \beta y = \gamma \quad \text{on } \partial\Omega, \tag{5.137}$$

$$-\sum_{k=1}^{d} \frac{\partial}{\partial x_k}\left(d_k \frac{\partial p}{\partial x_k}\right) - \sum_{k=1}^{d} c_k \frac{\partial p}{\partial x_k} + y = z \quad \text{in } \Omega, \tag{5.138}$$

$$\alpha \frac{\partial p}{\partial \nu} + \beta p = 0 \quad \text{on } \partial\Omega, \tag{5.139}$$

where the control equation $\nu u - p = 0$ has been used to eliminate the control function. It is assumed that $\sum_{k=1}^{d} \partial c_k / \partial x_k = 0$. The differential operator in (5.138) is the adjoint of the differential operator in (5.136): notice the change of sign of the convection term.

We discuss results of experiments where the desired target is given by

$$z(x, y, z) = \sin(3\pi x)\cos(3\pi y)\sin(\pi z).$$

The boundary conditions result from the following choice of the boundary parameters

$$\begin{array}{lllll} \text{planes } x = 0, x = 1: & \alpha = 1, & \beta = 0, & \gamma = 0; \\ \text{planes } y = 0, y = 1: & \alpha = 1, & \beta = 0, & \gamma = 0; \\ \text{planes } z = 0, z = 1: & \alpha = 0, & \beta = 1, & \gamma = 0. \end{array} \tag{5.140}$$

We solve the optimal control problem (5.136)–(5.140), with $d_k(\mathbf{x}) = 1$, and the convecting recirculating flow given by

$$\begin{array}{lcl} c_1 & = & -\sin\pi x \cos\pi y, \\ c_2 & = & \sin\pi y \cos\pi x, \\ c_3 & = & 0. \end{array} \tag{5.141}$$

Two techniques are considered: standard coarsening and aggressive coarsening. The number of coarse points obtained for the present test case is reported in Table 5.9. Observe that the reduction factor of the number of points from one level to the next coarser one is almost independent of the number of initial points. Aggressive coarsening is used only to pass from the finest to the next coarser level. In the first coarsening step, standard coarsening is only approximately halving the number of variables while aggressive coarsening reduces this number by a factor of approximately eight.

In Table 5.10 we report the convergence behavior of AMG solving the optimal control problem. We notice a weak dependency of the convergence factor ρ on the size of the problem. This behavior is observed when solving the optimal control problem with recirculating convection which appears to be the "worst case" for the AMG solver.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.9. *Number of variables at various levels of the coarsening process.*

	Standard coarsening		Aggressive coarsening	
k	N_i	N_i	N_i	N_i
1	120000	499200	120000	499200
2	60000	249600	14198	58067
3	10000	41608	6259	25454
4	1255	5225	991	3985
5	272	1210	161	639
6	41	193	31	98
7	-	34	-	18

Table 5.10. *Convergence properties depending on N_i ($\nu = 10^{-6}$).*

	Standard coarsening				Aggressive coarsening			
N_i	ρ	No. iter.	c_i/c_r	$\|\|u-z\|\|$	ρ	No. iter.	c_i/c_r	$\|\|u-z\|\|$
120000	0.14	15	2.65/2.68	$2.79\,10^{-2}$	0.40	33	1.70/1.69	$2.79\,10^{-2}$
499200	0.19	21	2.69/2.73	$2.61\,10^{-2}$	0.51	42	1.72/1.72	$2.60\,10^{-2}$
712800	0.23	25	2.70/2.74	$2.59\,10^{-2}$	0.60	47	1.72/1.72	$2.60\,10^{-2}$

Table 5.11. *Tracking properties depending on ν; $N_i = 120000$, standard coarsening.*

ν	$\|\|u-z\|\|$	ρ	No. iter.
10^{-4}	$2.61\,10^{-1}$	0.08	12
10^{-6}	$2.79\,10^{-2}$	0.14	15
10^{-8}	$3.89\,10^{-3}$	0.03	8

Also in Table 5.10 we compare the performance of AMG when using standard coarsening and aggressive coarsening. The clear advantage of aggressive coarsening is smaller values of the complexity factors compared with those obtained by standard coarsening. However, as may be expected [325], the convergence factor obtained using aggressive coarsening is larger than that obtained by standard coarsening.

In Table 5.11 tracking errors and convergence factors depending on ν are reported. In all experiments we observed AMG convergence that does not deteriorate as ν tends to be small. Notice that the computational performance of gradient methods, applied to optimal control problems belonging to the class considered here, worsen as ν tends to zero; see the discussion in [36].

A CSMG Scheme for a Shape Optimization Problem

We now describe a CSMG multigrid scheme for the shape optimization problem (2.9)–(2.10), that is, a simplified shape optimization problem formulated as an elliptic boundary control problem. Here the shape should be optimized so that a flow along the boundary approaches a certain pressure distribution.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Let us assume that $\Omega = (0,1)^2$ and Γ_1 consists of the boundary points with $x_2 = 0$, the bottom boundary. The optimality system is given by

$$\begin{array}{rcll} -\Delta y & = & 0 & \text{in } \Omega, \\ \frac{\partial y}{\partial \vec{n}} & = & \frac{\partial u}{\partial x} & \text{on } \Gamma_1, \\ y & = & 0 & \text{on } \partial\Omega\backslash\Gamma_1, \\ -\Delta p & = & 0 & \text{in } \Omega, \\ \frac{\partial p}{\partial \vec{n}} & = & -\frac{\partial}{\partial x}(\frac{\partial y}{\partial x} - P(x)) & \text{on } \Gamma_1, \\ p & = & 0 & \text{on } \partial\Omega\backslash\Gamma_1. \end{array} \tag{5.142}$$

As shown in [331], an appropriate choice for determining the control is to set $u = \frac{\partial p}{\partial x}$.

Denote first-order backward and forward partial derivatives of v_h in the x_i direction by ∂_i^- and ∂_i^+, respectively. They are given by

$$\partial_i^- v_h(\mathbf{x}) = \frac{v_h(\mathbf{x}) - v_h(\mathbf{x} - \hat{\imath}\, h)}{h} \qquad \text{and} \qquad \partial_i^+ v_h(\mathbf{x}) = \frac{v_h(\mathbf{x} + \hat{\imath}\, h) - v_h(\mathbf{x})}{h},$$

where $\hat{\imath}$ denotes the i coordinate direction vector and v_h is extended by 0 on grid points outside of Ω; see [174]. We have the five-point Laplacian $\Delta_h = \partial_1^+\partial_1^- + \partial_2^+\partial_2^-$.

In this framework, the discretization of (5.142) gives

$$\begin{array}{rcll} -\Delta_h y & = & 0 & \text{in } \Omega, \\ \partial_h^n y & = & \partial_1^+\partial_1^- p_h & \text{on } \Gamma_1, \\ y & = & 0 & \text{on } \partial\Omega\backslash\Gamma_1, \\ -\Delta_h p & = & 0 & \text{in } \Omega, \\ \partial_h^n p & = & -\partial_1^+\partial_1^- y_h + P_x & \text{on } \Gamma_1, \\ p & = & 0 & \text{on } \partial\Omega\backslash\Gamma_1, \end{array} \tag{5.143}$$

where $P_x = \frac{\partial P}{\partial x}$, and $\partial_h^n v_h = -(v_{i\,j+1} - v_{i\,j-1})/2h$ at Γ_1.

Notice that to solve (5.143) we need to realize the coupling on Γ_1 while in the interior of the domain we have two Laplace equations. For this purpose we combine the Neumann boundary conditions with the stencil of the discrete elliptic operators considered at the boundary. For $\mathbf{x} = (ih,\, jh)$ being a boundary grid point on the side $x_2 = 0$ we have

$$-(2y_{i+1\,j} + y_{i\,j-1} + y_{i\,j+1} - 4y_{i\,j}) - \frac{2}{h}(p_{i+1\,j} - 2p_{i\,j} + p_{i-1\,j}) = 0,$$

$$-(2p_{i+1\,j} + p_{i\,j-1} + p_{i\,j+1} - 4p_{i\,j}) + \frac{2}{h}(y_{i+1\,j} - 2y_{i\,j} + y_{i-1\,j}) = 2hP_x.$$

The equations obtained in this way have the same structure as (5.76) and (5.77) (without constraints) and the application of the CGS iteration follows along the same lines as above. Hence we can apply the CSMG multigrid method previously described. Notice that bilinear prolongation and full-weighting restriction have to be used to guarantee the right scaling for the coarse-grid problem formulation at the boundary; see the discussion in [340]. Clearly, on the boundary these transfer operators are mirrored.

To numerically validate the FAS multigrid algorithm for solving (5.143), consider the desired profile given by $P_x = 1$. Results for this case are reported in Table 5.12 for a V-cycle with two pre- and postsmoothing steps. These results show typical multigrid efficiency.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.12. *Results of experiments for a shape optimization problem.*

Mesh	129×129	257×257	517×517
ρ	0.08	0.08	0.08

CSMG Schemes for Bilinear Optimization Problems

An important class of optimization problems is represented by bilinear optimal control problems and inverse problems. Bilinear optimal control problems with PDEs is a less investigated subject representing a nonlinear control strategy with the aim of obtaining better system response than possible with linear control. Bilinear structures typically arise in quantum control problems and in parameter identification problems as we discuss in the applications chapter. A multigrid scheme for solving parameter identification problems of the form $\nabla \cdot (e^u \nabla y) = q$ is presented in [12]. Here we illustrate a CSMG scheme and a MGOPT scheme discussed in [345] to solve the following bilinear optimal control problem

$$\begin{cases} \min \quad J(y,u) := \frac{1}{2}\|y-z\|^2_{L^2(\Omega)} + \frac{\nu}{2}\|u\|^2_{L^2(\Omega)}, \\ -\Delta y - u\, y \;\; = g \quad \text{in} \quad \Omega, \\ \qquad\quad y \;\; = 0 \quad \text{on} \quad \partial\Omega, \end{cases} \tag{5.144}$$

where $z \in L^2(\Omega)$ is the target function. Similar to the linear case, we choose $f \in L^2(\Omega)$ and $U = L^2(\Omega)$. The solution to problem (5.144) is characterized by the following optimality system

$$\begin{aligned} -\Delta y - uy &= g && \text{in } \Omega, \\ y &= 0 && \text{on } \partial\Omega, \\ -\Delta p + y - up &= z && \text{in } \Omega, \\ p &= 0 && \text{on } \partial\Omega, \\ \nu u - yp &= 0 && \text{in } \Omega. \end{aligned} \tag{5.145}$$

For a given u, $y(u)$ and $p(u)$ are the solutions of the state and adjoint equations with homogeneous Dirichlet boundary conditions. Their existence requires that the operator $(\Delta + u)$ be invertible. Now notice that u cannot be constant on Ω since it satisfies $u = yp/\nu$ and it inherits the homogeneous boundary conditions as the state and the adjoint variables. Then we can use Lemma 3.2 from [158] based on results given in [139].

We can now derive the reduced gradient and the reduced Hessian as follows

$$\nabla \hat{J}(u) = \nu u - y\, p \tag{5.146}$$

and

$$\nabla^2 \hat{J}(u) = \nu \mathrm{I} + y(\Delta + u)^{-2} y + p(\Delta + u)^{-1} y + y(\Delta + u)^{-1} p. \tag{5.147}$$

In this case, we cannot state positivity of the reduced Hessian, and it is difficult to find the ellipticity and Lipschitz constants, unlike in the linear case. However, since $p = (\Delta + u)^{-1}(y - z)$ we can expect that for sufficiently accurate tracking, i.e., small $\|y - z\|$, and moderate values of ν the reduced Hessian is a positive definite operator. This situation may take place whenever z is (almost) attainable. That is, there exists a u such that $y(u) \approx z$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

Next, we define the collective smoothing iteration for the finite difference discretization of the bilinear elliptic optimal control problem given above. We have

$$\begin{aligned} -\Delta_k y_k - u_k y_k &= g_k, \\ -\Delta_k p_k + y_k - u_k p_k &= z_k, \\ \nu u_k - y_k p_k &= 0. \end{aligned} \tag{5.148}$$

In this case, we have

$$\begin{aligned} -(y_{i-1,j} + y_{i+1,j} + y_{i,j-1} + y_{i,j+1}) + 4y_{ij} - h^2 u_{ij}\, y_{ij} &= h^2 g_{ij}, \\ -(p_{i-1,j} + p_{i+1,j} + p_{i,j-1} + p_{i,j+1}) + 4p_{ij} + h^2 y_{ij} - h^2 u_{ij}\, p_{ij} &= h^2 z_{ij}, \\ \nu u_{ij} - y_{ij}\, p_{ij} &= 0. \end{aligned}$$

We first set

$$\begin{aligned} A_{ij} &= -(y_{i-1,j} + y_{i+1,j} + y_{i,j-1} + y_{i,j+1}) - h^2 g_{ij}, \\ B_{ij} &= -(p_{i-1,j} + p_{i+1,j} + p_{i,j-1} + p_{i,j+1}) - h^2 z_{ij}. \end{aligned} \tag{5.149}$$

Now, the values A_{ij} and B_{ij} are considered constant during the update of the variables at ij. Hence, we have the following system of equations of three variables y_{ij}, u_{ij}, and p_{ij}

$$\begin{aligned} A_{ij} + 4y_{ij} - h^2 u_{ij}\, y_{ij} &= 0, \\ B_{ij} + 4p_{ij} + h^2 y_{ij} - h^2 u_{ij}\, p_{ij} &= 0, \\ \nu u_{ij} - y_{ij}\, p_{ij} &= 0. \end{aligned}$$

We see that the resulting system of equations is nonlinear, and thus computing the updates for the variables u_{ij}, y_{ij}, and p_{ij} requires us to apply a local Newton step. This approach results in a nonrobust smoothing iteration apparently because we have multiple solutions for u_{ij} that are close. In fact, as we show below, the condition $\nu u_{ij} - y_{ij}(u_{ij})\, p_{ij}(u_{ij}) = 0$ results in a quartic polynomial equation for u_{ij}, and therefore four roots are possible. To determine these solutions we construct the quartic polynomial and solve it exactly by using the Cardano–Tartaglia formula. In this way we can explore among the possible solutions of the optimization step. To construct the quartic polynomial, we can define $y_{ij} = y_{ij}(u_{ij})$ and $p_{ij} = p_{ij}(u_{ij})$ as functions of u_{ij},

$$\begin{aligned} y_{ij}(u_{ij}) &= \tfrac{-1}{4-h^2 u_{ij}} A_{ij}, \\ p_{ij}(u_{ij}) &= \tfrac{1}{(4-h^2 u_{ij})^2}(h^2 A_{ij} + h^2 B_{ij} u_{ij} - 4B_{ij}), \end{aligned}$$

and equate the reduced gradient to zero, i.e., $\nabla \hat{J}(u) = \nu u - y(u)p(u) = 0$. Hence we have a quartic polynomial equation in u_{ij} given by

$$\nu h^6 u_{ij}^4 - 12\nu h^4 u_{ij}^3 + 48\nu h^2 u_{ij}^2 - (64\nu + h^2 A_{ij} B_{ij}) u_{ij} - (h^2 A_{ij}^2 - 4A_{ij} B_{ij}) = 0.$$

The solutions of the quartic polynomial are either four real or two real and two complex. The two complex conjugate solutions can be disregarded. In order to find the minimizer, we choose the minimum real solution of the quartic polynomial which minimizes

$$\hat{J}_{ij}(u) = \frac{1}{2}(y_{ij}(u) - z_{ij})^2 + \frac{\nu}{2} u_{ij}^2.$$

With this condition, we get a robust and efficient CSMG smoothing iteration.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

We see that in the bilinear case the construction of an appropriate CSMG scheme is more involved. It is therefore interesting to consider the solution of the bilinear control problem using the MGOPT method. To apply this method, we choose a classical optimization scheme as single-grid solver and embed this scheme in the MGOPT algorithm as illustrated in Section 5.4. We consider two gradient-type schemes: the steepest descent method and the NCG scheme discussed in Chapter 4. To evaluate the gradient (5.146) at u, we solve the state and the adjoint equations very accurately. The line search in the coarse-grid correction step uses the Armijo rule [265].

In the following, we report results of numerical experiments obtained with the CSMG scheme and with the MGOPT scheme applied to the minimization of the reduced cost functional.

Results of numerical experiments are obtained with the following setting. Let $\Omega = (0,1) \times (0,1)$, and let $f, z \in L^2(\Omega)$ be given by

$$g(x,y) = 1,$$
$$z(x,y) = \begin{cases} 2 & \text{on } (0.25, 0.75) \times (0.25, 0.75), \\ 1 & \text{otherwise.} \end{cases}$$

Notice that the target z is not attainable by any control because of its nonzero value at the boundary of the domain.

Numerical results are shown in Tables 5.13 and 5.14 and in Figure 5.11. We can see from Table 5.13 that the CSMG method exhibits almost independence of the number of iterations on ν on the size of the mesh where the problem is being solved. On the other hand the computational effort required by the MGOPT scheme increases as $1/\nu$. In Table 5.13, we also report on a separate column the CPU time for the computation of the roots of the quartic polynomial, as this takes about 70% of the whole computational time. We show in Table 5.14 the computational time for the one-grid optimization scheme using steepest descent and NCG schemes. For solution processes exceeding 20,000 seconds we stopped the calculation. This table shows that MGOPT with NCG is faster than MGOPT with steepest descent.

Table 5.13. *Results of bilinear elliptic optimal control problem using the CSMG method (* time for computing the roots).*

ν	Mesh	Iter.	$\hat{J}$	$\|\nabla \hat{J}\|_{L_2}$	ρ	Time (sec)	Time (sec)*
	65×65	8	0.863	2.06e-11	0.081	47.6	33.2
1e-2	129×129	9	0.843	1.85e-12	0.083	221.9	155.8
	257×257	9	0.834	1.90e-12	0.084	912.8	641.3
	65×65	10	0.158	1.55e-08	0.287	59.7	41.5
1e-4	129×129	10	0.151	1.46e-08	0.366	243.1	170.1
	257×257	10	0.148	1.45e-08	0.434	997.5	696.7

The presence of constraints on the optimization function u can be easily implemented in the CSMG smoothing by projection of the u-update obtained from the solution of the quartic polynomial equation.

The MGOPT solution of control-constrained bilinear control problems is discussed in [344], where a projected gradient optimization scheme is considered.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.14. *Results of CPU time (seconds) of bilinear elliptic optimal control problem using the steepest descent (SD) method, MGOPT with SD (MGOPT[1]), NCG, and MGOPT with NCG (MGOPT[2]) (– longer than* 20,000 *seconds).*

ν	Mesh	SD	MGOPT[1]	NCG	MGOPT[2]
	65×65	1.2	1.0	0.8	0.8
1e-2	129×129	5.2	4.9	4.0	3.7
	257×257	25.1	19.1	19.0	18.8
	65×65	–	–	17079.0	160.7
1e-4	129×129	–	–	–	805.7
	257×257	–	–	–	4112.4

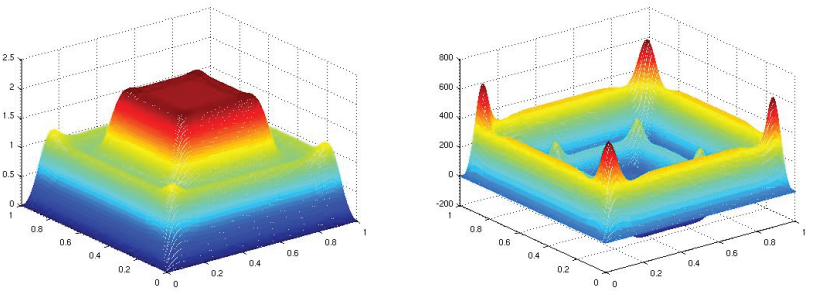

Figure 5.11. *Numerical solutions for the state (left) and control (right) variables of the bilinear elliptic optimal control problem with* $\nu = 10^{-6}$.

For an application, consider a unit square domain $\Omega = (0,1) \times (0,1)$, with $g, z \in L^2(\Omega)$, given by

$$g(x_1,x_2) = \sin(2\pi x_1)\sin(2\pi x_2),$$
$$z(x_1,x_2) = 1 + \sin(2\pi x_1)\sin(2\pi x_2).$$

Notice that the target function z is not attainable by any control due to its boundary values. We have box constraints.

The numerical results for computational performance are shown in Table 5.15. In Figure 5.12, the optimal solution for $\nu = 10^{-4}$ is depicted. See [344] for more details.

5.7.3 A CSMG Scheme with FEM Discretization

In this section, we discuss the realization of the CSMG scheme in the case of FEM discretization. Consider the optimality system

$$\begin{array}{rcll} -\nu\Delta y - p &=& \nu f & \text{in } \Omega, \\ -\Delta p + y &=& z & \text{in } \Omega, \end{array} \tag{5.150}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.15. *Results of control-constrained bilinear elliptic optimal control problem using MGOPT with gradient projection method.*

ν	Mesh	$\|r_y\|_{L^2}$	$\|r_p\|_{L^2}$	$\|u^l - u^l(1)\|_{L^2}$	Time (sec)
10^{-2}	129×129	2.086e-14	1.017e-13	4.581e-06	2.7
	257×257	6.798e-14	3.710e-13	6.044e-06	11.8
	513×513	6.290e-13	3.580e-12	6.277e-06	49.3
10^{-4}	129×129	1.567e-14	9.977e-14	6.057e-06	3.6
	257×257	4.219e-14	2.912e-13	1.841e-06	16.2
	513×513	1.631e-13	9.025e-13	1.364e-06	59.4

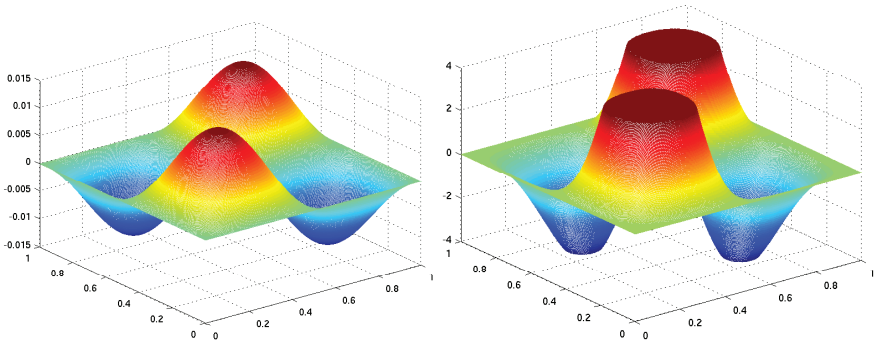

Figure 5.12. *Numerical solutions for the state (left) and control (right) variables of the control-constrained bilinear elliptic optimal control problem using* $\nu = 10^{-4}$.

together with the boundary conditions

$$y = 0 \quad \text{and} \quad p = 0 \quad \text{on} \quad \partial\Omega. \tag{5.151}$$

We convert this problem into its variational formulation as follows: find $(y, p) \in V \times V$ such that

$$\begin{aligned} \int_\Omega (\nu \nabla y \cdot \nabla v_1 - p v_1 - \nu f v_1)\, dx = 0 \quad \forall v_1 \in V, \\ \int_\Omega (\nabla p \cdot \nabla v_2 + y v_2 - z v_2)\, dx = 0 \quad \forall v_2 \in V, \end{aligned} \tag{5.152}$$

where $V \times V$ is the solution space. Next we replace $V \times V$ in (5.152) by a finite-dimensional subspace $V_k \times V_k \subset V \times V$, and the dimension of V_k is n_k. We assume that linearly independent basis functions ϕ_i, $i = 1, 2, \ldots, n_k$, span V_k. Then for $i = 1, 2, \ldots, n_k$, we have

$$\begin{aligned} \int_{\Omega_k} (\nu \nabla_k y_k \cdot \nabla_k \phi_i - p_k \phi_i - \nu f_k \phi_i)\, dx &= 0, \\ \int_{\Omega_k} (\nabla_k p_k \cdot \nabla_k \phi_i + y_k \phi_i - z_k \phi_i)\, dx &= 0. \end{aligned} \tag{5.153}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

We can write the approximate solutions y_k and p_k in terms of basis functions ϕ_i of V_k as follows

$$y_k(x) = \sum_{j=1}^{n_k} Y_j \phi_j(x) \quad \text{and} \quad p_k(x) = \sum_{j=1}^{n_k} P_j \phi_j(x), \tag{5.154}$$

where Y_j and P_j, $j = 1, \ldots, n_k$, are the unknowns. Thus (5.153) can be rewritten as

$$\begin{pmatrix} \nu Q_k & -M_k \\ M_k & Q_k \end{pmatrix} \begin{pmatrix} Y_k \\ P_k \end{pmatrix} = \begin{pmatrix} \nu G_k^1 \\ G_k^2 \end{pmatrix}, \tag{5.155}$$

where

$$\begin{aligned} Q_{i,j} &= \int_\Omega \nabla\phi_j \cdot \nabla\phi_i \, dx, & M_{i,j} &= \int_\Omega \phi_j \phi_i \, dx, \\ G_i^1 &= \int_\Omega f_h \phi_i \, dx, & G_i^2 &= \int_\Omega z_h \phi_i \, dx, \end{aligned} \tag{5.156}$$

where for simplicity we have omitted the index k. Here Q_k and M_k are called the stiffness and mass matrices, respectively. Let

$$\tilde{A} = \begin{pmatrix} \nu Q & -M \\ M & Q \end{pmatrix}, \quad \tilde{U} = \begin{pmatrix} Y \\ P \end{pmatrix}, \quad \text{and } \tilde{F} = \begin{pmatrix} \nu G^1 \\ G^2 \end{pmatrix};$$

then we have

$$\tilde{A}\tilde{U} = \tilde{F}, \tag{5.157}$$

where $\tilde{A}$ is a $2n_k \times 2n_k$ matrix and vectors $\tilde{U}$ and $\tilde{F}$ are of length $2n_k$. In order to introduce a collective smoothing scheme, we define the following inner product

$$(\tilde{A}_{i+d,\cdot}, \tilde{U})_d = \sum_{\substack{j=1 \\ j \neq \{i, i+n_k\}}}^{2n_k} \tilde{a}_{i+d,j} \tilde{u}_j \, . \tag{5.158}$$

By choosing the ith and $(i + n_k)$th rows of matrix $\tilde{A}$, we get

$$\begin{pmatrix} \tilde{A}_{i,i} & \tilde{A}_{i,i+n_k} \\ \tilde{A}_{i+n_k,i} & \tilde{A}_{i+n_k,i+n_k} \end{pmatrix} \begin{pmatrix} \tilde{U}_i \\ \tilde{U}_{i+n_k} \end{pmatrix} = \begin{pmatrix} \tilde{F}_i - (\tilde{A}_{i,\cdot}, \tilde{U})_0 \\ \tilde{F}_{i+n_k} - (\tilde{A}_{i+n_k,\cdot}, \tilde{U})_{n_k} \end{pmatrix}.$$

Let

$$\tilde{D} = \det \begin{pmatrix} \tilde{A}_{i,i} & \tilde{A}_{i,i+n_k} \\ \tilde{A}_{i+n_k,i} & \tilde{A}_{i+n_k,i+n_k} \end{pmatrix} = \tilde{A}_{i,i}\tilde{A}_{i+n_k,i+n_k} - \tilde{A}_{i,i+n_k}\tilde{A}_{i+n_k,i}.$$

Thus we obtain a collective update given by

$$\begin{pmatrix} \tilde{U}_i \\ \tilde{U}_{i+n_k} \end{pmatrix} = \frac{1}{\tilde{D}} \begin{pmatrix} \tilde{A}_{i+n_k,i+n_k} & -\tilde{A}_{i,i+n_k} \\ -\tilde{A}_{i+n_k,i} & \tilde{A}_{i,i} \end{pmatrix} \begin{pmatrix} \tilde{F}_i - (\tilde{A}_{i,\cdot}, \tilde{U})_0 \\ \tilde{F}_{i+n_k} - (\tilde{A}_{i+n_k,\cdot}, \tilde{U})_{n_k} \end{pmatrix}. \tag{5.159}$$

This collective smoothing solves the system componentwise by treating both variables collectively.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Next, we consider the elliptic optimal control problem with constraints on the control. Recall the state and the adjoint equations in the optimality system,

$$\begin{aligned} -\Delta y - u &= f, \\ -\Delta p + y &= z, \end{aligned} \tag{5.160}$$

together with the homogeneous Dirichlet boundary conditions.

In this case, we have two equations, a variational inequality, and three unknowns. We will use the components of the system matrix $\tilde{A}$ in (5.157) and change the vector $\tilde{U}$ as needed for each equation. Thus, we have

$$\begin{aligned} (Q \quad -M)\begin{pmatrix} Y \\ U \end{pmatrix} &= G^1, \\ (M \qquad Q)\begin{pmatrix} Y \\ P \end{pmatrix} &= G^2. \end{aligned} \tag{5.161}$$

Then the following control-constrained optimality system results

$$\begin{aligned} QY - MU &= G^1, \\ QP + MY &= G^2, \\ (\nu U - P, \Phi - U) &\geq 0 \qquad \forall \Phi \in \mathcal{U}_k. \end{aligned} \tag{5.162}$$

A collective smoothing step updates the values Y_i, U_i, and P_i such that the resulting residuals of the state and adjoint equations at that point are zero. We first set

$$C_1 = \sum_{\substack{j=1 \\ j\neq i}}^{n_k} q_{i,j} Y_j, \quad C_2 = \sum_{\substack{j=1 \\ j\neq i}}^{n_k} q_{i,j} P_j, \quad C_3 = \sum_{\substack{j=1 \\ j\neq i}}^{n_k} m_{i,j} Y_j, \text{ and } \quad C_4 = \sum_{\substack{j=1 \\ j\neq i}}^{n_k} m_{i,j} U_j. \tag{5.163}$$

The values C_1, C_2, C_3, and C_4 are considered constant during the updates for the variables at i. Hence, we have the following system of equations of three variables Y_i, U_i, and P_i

$$\begin{aligned} C_1 + q_{i,i} Y_i - C_4 - m_{i,i} U_i - G_i^1 &= 0, \\ C_2 + q_{i,i} P_i + C_3 + m_{i,i} Y_i - G_i^2 &= 0. \end{aligned}$$

Since this is a linear system, we can compute the updates for the variables Y_i and P_i as functions of U_i in the following way:

$$\begin{aligned} Y_i(U_i) &= \frac{1}{q_{i,i}} \left(G_i^1 - C_1 + C_4 + m_{i,i} U_i \right), \\ P_i(U_i) &= \frac{1}{q_{i,i}^2} \left[q_{i,i} \left(G_i^2 - C_2 - C_3 \right) - m_{i,i} \left(G_i^1 - C_1 + C_4 + m_{i,i} U_i \right) \right]. \end{aligned} \tag{5.164}$$

To obtain an update U_i, replace the expression for P_i in the inequality constraint and define the auxiliary variable as

$$\widetilde{U}_i = \frac{1}{\nu q_{i,i}^2 + m_{i,i}^2} \left[q_{i,i} \left(G_i^2 - C_2 - C_3 \right) - m_{i,i} \left(G_i^1 - C_1 + C_4 \right) \right]. \tag{5.165}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Then the new value for u_i resulting from the smoothing step is given by

$$U_i = \begin{cases} \overline{u}_i & \text{if} \quad \widetilde{U}_i \geq \overline{u}_i, \\ \widetilde{U}_i & \text{if} \quad \underline{u}_i < \widetilde{U}_i < \overline{u}_i, \\ \underline{u}_i & \text{if} \quad \widetilde{U}_i \leq \underline{u}_i. \end{cases} \tag{5.166}$$

With this new value of U_i, new values for Y_i and P_i are obtained. This completes the description of the collective smoothing step for the finite element control constrained case.

The collective smoothing step defined by (5.164)–(5.166) satisfies the inequality constraint in the optimality system. Consider any grid point wherein $\widetilde{U} \leq \underline{u}$; then, from (5.166), $U = \underline{u}$. Thus, $(\Phi - U) \geq 0$ for any $\Phi \in \mathcal{U}$. On the other hand, we have

$$\begin{aligned} \nu U - P &= \nu U - \frac{1}{q_{i,i}^2}\left[q_{i,i}\left(G_i^2 - C_2 - C_3\right) - m_{i,i}\left(G_i^1 - C_1 + C_4 + m_{i,i}U_i\right)\right] \\ &= \frac{1}{q_{i,i}^2}\left[\left(q_{i,i}^2\nu + m_{i,i}^2\right)U_i - q_{i,i}\left(G_i^2 - C_2 - C_3\right) - m_{i,i}\left(G_i^1 - C_1 + C_4\right)\right] \\ &\geq \frac{1}{q_{i,i}^2}\left[\left(q_{i,i}^2\nu + m_{i,i}^2\right)\widetilde{U} - q_{i,i}\left(G_i^2 - C_2 - C_3\right) - m_{i,i}\left(G_i^1 - C_1 + C_4\right)\right] \\ &= 0. \end{aligned}$$

Therefore, $(\nu U - P, \Phi - U) \geq 0$ for all $\Phi \in \mathcal{U}$. Similarly, we can prove that if $\widetilde{U} \geq \overline{u}$, then the choice $U = \overline{u}$ satisfies the inequality constraint. The case $\underline{u} < \widetilde{U} < \overline{u}$ is obvious.

A typical multigrid method uses a sequence of l nested triangulations of Ω of increasing fineness $\mathcal{T}_1 \subset \mathcal{T}_2 \subset \cdots \subset \mathcal{T}_l$, where $\mathcal{T}_l$ denotes the finest grid. Let $\mathcal{T}_1$ be given and $\mathcal{T}_k, k \geq 2$, be obtained from $\mathcal{T}_{k-1}$ via a regular subdivision; i.e., edge midpoints of $\mathcal{T}_{k-1}$ are connected by new edges to form $\mathcal{T}_k$. Furthermore, h_k denotes the mesh size of $\mathcal{T}_k$, i.e., $h_k := \max_{T \in \mathcal{T}_k} \operatorname{diam} T$ and n_k denotes the number of nodes in $\mathcal{T}_k$, where $n_{k-1} < n_k$. To each triangulation $\mathcal{T}_k, k = 1, \ldots, l$, we define the associated sequence of finite element spaces $V_1 \subset V_2 \subset \cdots \subset V_l$. The mesh hierarchy induces linear systems $A_k u_k = f_k, k = 1, 2, \ldots, l$, on each grid level k. Note that $\mathcal{T}_{k-1} \subset \mathcal{T}_k \Rightarrow V_{k-1} \subset V_k$ and $h_k = \frac{1}{2}h_{k-1}$ since each triangle $T \in \mathcal{T}_{k-1}$ is subdivided into four similar triangles in $\mathcal{T}_k$. In addition to the grid levels, we need transfer operators between coarser and finer grids. Since we have assumed that the finite element discretization has piecewise linear basis functions, we can now define an interpolation operator from a coarse triangulation $\mathcal{T}_{k-1}$ to a fine triangulation $\mathcal{T}_k$ as follows

$$\left(I_{k-1}^k\right)_{ij} = \begin{cases} 1 & \text{for } i = j, \text{ where the nodes } P_i \text{ are both in the triangulations} \\ & \mathcal{T}_k \text{ and } \mathcal{T}_{k-1}\ \forall i, j = 1, 2, \ldots, n_{k-1}, \\ \frac{1}{2} & \text{for } j = i_1 \text{ and } j = i_2, n_{k-1} < i \leq n_k, \\ & \text{where the nodes } P_{i1} \text{ and } P_{i2} \text{ are the boundary nodes of every triangle} \\ & T \in \mathcal{T}_{k-1}, \text{ of where the nodes } P_i \text{ are defined,} \\ 0 & \text{otherwise.} \end{cases}$$

Here, we assume that the nodes are numbered hierarchically: first the nodes of triangulation $\mathcal{T}_1$, then the newly added nodes of $\mathcal{T}_2$, and so on. The restriction operator is defined as the transpose of the interpolation operator as follows

$$I_k^{k-1} = \left(I_{k-1}^k\right)^\top. \tag{5.167}$$

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Notice that after the restriction of the residual in the CSMG algorithm, all components which belong to Dirichlet boundary nodes must be set to zero since we know that the residual on the boundary is zero.

Next, we present the results of the numerical experiments using the CSMG scheme with finite elements on different domains. For all computations, we use $\gamma_1 = \gamma_2 = 2$ pre- and postsmoothing steps. We report the CPU time until the norm of the residual, $\|r\|_{L^2}$, satisfies a stopping tolerance of $tol = 10^{-7}$. We also report the observed convergence factor.

First, we consider the following elliptic optimal control problem

$$\begin{aligned} \min J(y,u) &:= \tfrac{1}{2}\|y-z\|^2_{L^2} + \tfrac{\nu}{2}\|u\|^2_{L^2}, \\ -\Delta y &= f+u \quad \text{in} \quad \Omega, \\ y &= 0 \quad \text{on} \quad \partial\Omega. \end{aligned} \tag{5.168}$$

Let $\Omega = (0,1)\times(0,1)$, and let $f,\ z \in L^2(\Omega)$ be given by

$$\begin{aligned} f &= 0, \\ z &= \sin(2\pi x_1)\sin(\pi x_2). \end{aligned}$$

The target function z is shown in Figure 5.13 together with the chosen domain $\Omega_1 = \Omega$, which is a square with a circle hole with radius $r = 0.25$ and center at $(x_1, x_2) = (0.5, 0.5)$.

Results of numerical experiments for this case are reported in Table 5.16. We can see that choosing different values for the parameter ν, the multigrid method converges within three iterations. That is, the algorithm exhibits an independence of the number of iterations on the parameter ν and also on the size of the mesh. From the CPU times we find that the complexity of the problem is $O(n\log(n))$, where n denotes the number of nodes. These results are in agreement with the estimate given by Theorem 5.19 in the previous section.

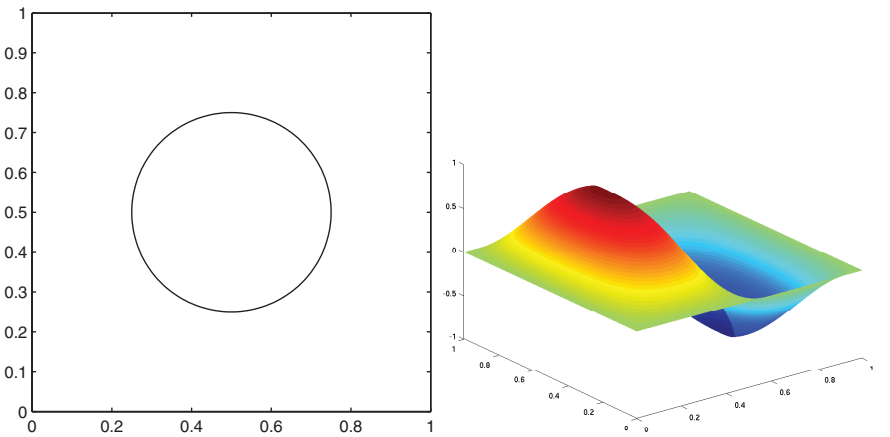

Figure 5.13. *Domain Ω_1: Square on $(0,1)\times(0,1)$ minus a circle with radius $r = 0.25$ and center at $(0.5,0.5)$ (left) and the target function z (right). Reprinted with permission from O. Lass, M. Vallejos, A. Borzì, and C.C. Douglas, Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing,* 84(1-2) (2009), 27–48.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.16. *Results of the elliptic optimal control problem with unconstrained control on the domain Ω_1 using the CSMG method with finite elements.*

ν	n	Iter.	ρ_y	ρ_p	$\|r_y\|_{L^2}$	$\|r_p\|_{L^2}$	Time (s)
10^{-2}	24960	3	0.081	0.053	3.15e-08	6.81e-07	0.44
	99072	3	0.085	0.058	1.77e-08	4.34e-07	1.93
	394752	3	0.089	0.062	9.73e-09	2.62e-07	8.23
10^{-4}	24960	3	0.039	0.037	3.25e-09	2.84e-07	0.44
	99072	3	0.042	0.039	1.99e-09	1.61e-07	1.99
	394752	3	0.046	0.041	1.19e-09	9.01e-08	8.26
10^{-6}	24960	3	0.043	0.067	3.40e-10	4.22e-07	0.43
	99072	3	0.045	0.069	1.89e-10	2.29e-07	1.94
	394752	3	0.048	0.068	1.04e-10	1.19e-07	8.48

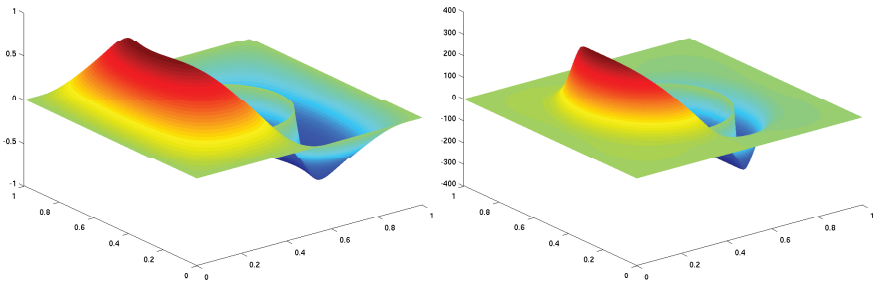

Figure 5.14. *Numerical solutions for the state (left) and control (right) variables of the elliptic optimal control with unconstrained control problem using $\nu = 10^{-6}$. Reprinted with permission from O. Lass, M. Vallejos, A. Borzì, and C.C. Douglas, Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing,* 84(1-2) (2009), 27–48.

The numerical optimal solutions y and u for $\nu = 10^{-6}$ are shown in Figure 5.14.

Next, we discuss the elliptic optimal control problem with constrained control given as follows

$$\begin{aligned} \min J(y,u) &:= \tfrac{1}{2}\|y-z\|^2_{L^2} + \tfrac{\nu}{2}\|u\|^2_{L^2}, \\ -\Delta y - u &= f \quad \text{in} \quad \Omega, \\ y &= 0 \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{5.169}$$

where f, $z \in L^2(\Omega)$ are the same as in the unconstrained control case. Here, $u \in \mathcal{U} \subset L^2(\Omega)$ since

$$\mathcal{U} = \left\{ u \in L^2(\Omega) | -30 \le u(x) \le 30 \text{ a.e. in } \Omega \right\}. \tag{5.170}$$

First, we consider this problem on the same domain as depicted in Figure 5.13. Results of numerical experiments for this case are given in Table 5.17. We also consider the domain Ω_2 shown in Figure 5.15 defined as $\frac{3}{4}$ of a circle with radius $r = 1$ and center at $(0.5, 0.5)$. For this latter case we report results of numerical experiments in Table 5.18.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.17. *Results of the elliptic optimal control problem with constrained control on the domain Ω_1 using the CSMG method with finite elements.*

ν	n	Iter.	ρ_y	ρ_p	$\|r_y\|_{L^2}$	$\|r_p\|_{L^2}$	Time (s)
10^{-4}	24960	5	0.070	0.075	3.23e-07	3.12e-09	0.55
	99072	5	0.074	0.079	2.03e-07	1.81e-09	2.61
	394752	5	0.077	0.081	1.21e-07	1.01e-09	12.10
10^{-6}	24960	10	0.393	0.391	6.86e-07	3.97e-09	1.09
	99072	9	0.392	0.391	8.97e-07	5.13e-09	4.77
	394752	9	0.391	0.391	4.50e-07	2.55e-09	20.80

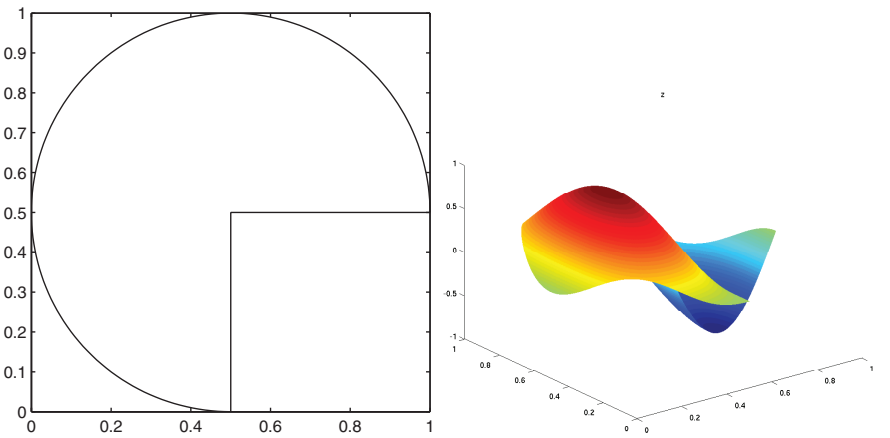

Figure 5.15. *Domain Ω_2: $\frac{3}{4}$ of a circle with radius $r = 1$ and center at $(0.5, 0.5)$ (left) and the target function z (right). Reprinted with permission from O. Lass, M. Vallejos, A. Borzì, and C.C. Douglas, Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing,* 84(1-2) (2009), 27–48.

Table 5.18. *Results of the elliptic optimal control problem with constrained control on the domain Ω_2 using the CSMG method with finite elements.*

ν	n	Iter.	ρ_y	ρ_p	$\|r_y\|_{L^2}$	$\|r_p\|_{L^2}$	Time (s)
10^{-4}	24865	5	0.083	0.080	3.50e-07	3.03e-09	0.6
	98881	5	0.097	0.084	2.54e-07	1.65e-09	2.7
	394369	5	0.112	0.086	2.24e-07	8.99e-10	11.7
10^{-6}	25361	17	0.762	0.540	9.73e-07	1.67e-09	1.9
	100897	16	0.236	0.364	3.00e-07	1.75e-09	8.6
	402497	15	0.401	0.411	6.17e-07	2.42e-09	35.4

In the constrained control case, we see that by choosing different values for the parameter ν, the multigrid method exhibits a convergence behavior which is almost independent of the mesh size. However, larger values of the convergence factor are obtained by

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

decreasing the value of the optimization weight. This is reasonable since the presence of box constraints may result in steep gradients of the control function. The numerical solutions y and u on Ω_1 and choosing $\nu = 10^{-6}$ are shown in Figure 5.16. The optimal solutions for two different choices of $\nu \in \{10^{-6}, 10^{-4}\}$ computed on Ω_2 are shown in Figure 5.17.

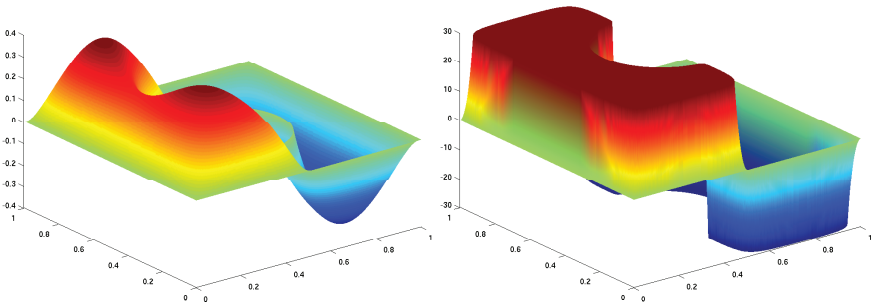

Figure 5.16. *Numerical solutions for the state (left) and control (right) variables of the elliptic optimal control problem with constrained control on the domain Ω_1 using $\nu = 10^{-6}$. Reprinted with permission from O. Lass, M. Vallejos, A. Borzì, and C.C. Douglas, Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing,* 84(1-2) (2009), 27–48.

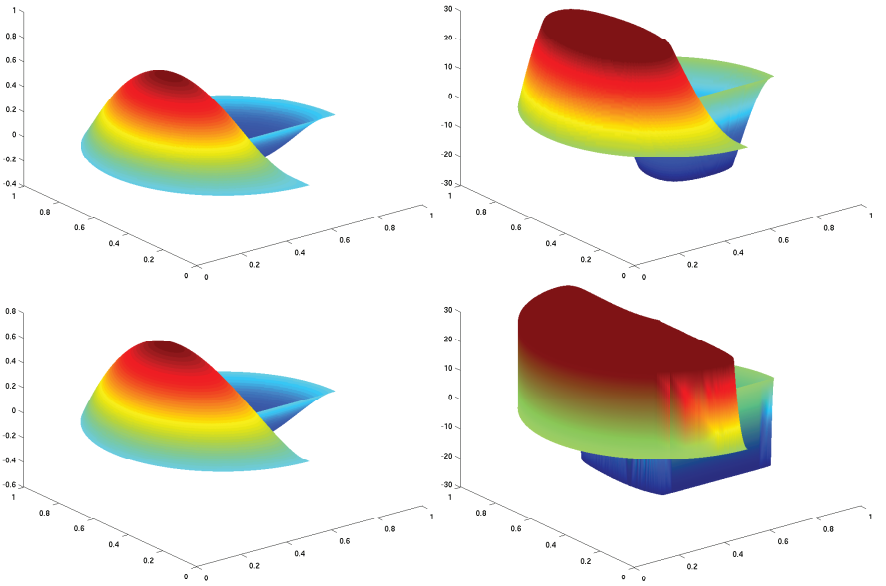

Figure 5.17. *Numerical solutions y (top left) and u (top right) on the domain Ω_2 and parameter $\nu = 10^{-4}$. Numerical solutions y (bottom left) and u (bottom right) on the domain Ω_2 and parameter $\nu = 10^{-6}$. Reprinted with permission from O. Lass, M. Vallejos, A. Borzì, and C.C. Douglas, Implementation and analysis of multigrid schemes with finite elements for elliptic optimal control problems, Computing,* 84(1-2) (2009), 27–48.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

We can see steeper gradients of the control function as we reduce the value of the weight. We remark that quite similar results are obtained for nonconvex domains and in the case of domains with reentrant corners.

5.7.4 CSMG Schemes for Parabolic Control Problems

In this section, we start our discussion on multigrid schemes for parabolic optimal control problems. We describe space-time CSMG multigrid schemes that have been presented in, e.g., [46, 49, 50, 53, 58, 150]. These schemes are based on smoothing methods that provide a robust implementation of the time coupling in the optimality system.

Control of parabolic systems has many applications in biology [61, 180], chemistry [62, 229], and physiology [3, 272]. To solve the related optimal control problems, the solution of the corresponding optimality systems which are governed by reaction-diffusion equations with opposite time orientation is considered. Of particular importance in applications and for benchmark purposes are singular optimal control problems [236]. Especially in these cases, the coupling between state variables and controls must be realized in a robust way in order to guarantee convergence of the algorithms. Robust coupling is also necessary for designing algorithms that permit efficient solution to optimization problems with a computational performance that is independent of the value of the optimization parameters. The results presented in this chapter and in the references mentioned above show that the space-time multigrid methods considered here do meet these requirements. These objectives are achieved by using appropriate smoothing techniques and by solving optimality systems for distributed control or boundary control in one shot in the whole space-time cylinder.

The CSMG approach represents an extension of space-time multigrid schemes for parabolic problems [170, 202, 347] to the case of reaction-diffusion systems with opposite time orientation. For this particular structure, two different smoothing schemes in combination with semicoarsening in space are considered. Other coarsening strategies are possible; see [57, 58]. In the case of tracking along trajectories a pointwise relaxation is presented which can be successfully applied [57, 49, 50] to solve singular parabolic optimal control problems. In the case of terminal observation, block relaxation is the most robust choice. Block smoothing is also advantageous in the case of reaction-diffusion problems with very small diffusion as it occurs in the chemical turbulence modeling and physiology; see [61, 62, 46] and the references given therein.

A disadvantage of any space-time approach is the requirement of storing the dependent variables for all time steps. This is certainly a limitation that arises when open-loop optimal control problems on a large time interval are considered. However, in the limiting case of very long time intervals, this difficulty is overcomed by combining space-time multigrid schemes with receding-horizon techniques [4, 208].

Depending on the application, reaction-diffusion processes can be controlled through source terms or through boundary terms. In the case of distributed control through source terms, the following optimal control problem is formulated

$$\left\{\begin{array}{rcll} \min J(y,u), & & & \\ -\partial_t y + G(y) + \sigma\,\Delta y &=& f+u & \text{in } Q, \\ y &=& y_0 & \text{in } \Omega\times\{t=0\}, \\ y &=& g & \text{on } \Sigma, \end{array}\right. \tag{5.171}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

where $Q = \Omega \times (0,T)$ and $\Sigma = \partial\Omega \times (0,T)$. To be specific, we take $y_0 \in H_0^1(\Omega)$, $g \in C(\Sigma)$, and $f, u \in L^2(Q)$. In (5.171), the nonlinear term $G(y)$ models the reaction kinetics for the state y and u represents the control function which for the moment we assume to be unconstrained. Here $\sigma > 0$ is the diffusion coefficient.

Alternatively, we consider the following optimal Neumann boundary control problem

$$\begin{cases} \min J(y,u), & & \\ -\partial_t y + G(y) + \sigma\,\Delta y &= f & \text{in } Q, \\ y &= y_0 & \text{in } \Omega \times \{t=0\}, \\ -\frac{\partial y}{\partial n} &= u & \text{on } \Sigma, \end{cases} \tag{5.172}$$

where $u \in L^2(\Sigma)$.

Control may be required to track a desired trajectory given by $y_d(\mathbf{x},t) \in L^2(Q)$ or to reach a desired terminal state $y_T(\mathbf{x}) \in L^2(\Omega)$. For this purpose, the following cost functional is considered

$$J(y,u) = \frac{\alpha}{2}||y - y_d||^2_{L^2(Q)} + \frac{\beta}{2}||y(\cdot,T) - y_T||^2_{L^2(\Omega)} + \frac{\nu}{2}||u||^2_{L^2(X)}, \tag{5.173}$$

with $X = Q$ or $X = \Sigma$ corresponding to distributed control or boundary control, respectively. Here, $\nu > 0$ is the weight of the cost of the control and $\alpha \geq 0$, $\beta \geq 0$, $\alpha + \beta > 0$, are optimization parameters. The case $\alpha = 1$, $\beta = 0$ corresponds to tracking without terminal observation, while the case $\alpha = 0$ and $\beta = 1$ means reaching a desired final target configuration no matter which trajectory is followed.

Existence of solutions to the optimal control problems (5.171) and (5.172) can be established under suitable conditions for various forms of the nonlinearity; see, e.g., [49, 148, 207, 216, 236, 262, 339].

The solution to (5.171) is characterized by the following first-order optimality system

$$\begin{aligned} -\partial_t y + G(y) + \sigma\,\Delta y &= f + u && \text{in } Q, \\ \partial_t p + G'(y)p + \sigma\,\Delta p + \alpha(y - y_d) &= 0 && \text{in } Q, \\ \nu u - p &= 0 && \text{in } Q, \\ y = g,\ p &= 0 && \text{on } \Sigma. \end{aligned} \tag{5.174}$$

In the case of boundary control (5.172), the optimal solution satisfies the following

$$\begin{aligned} -\partial_t y + G(y) + \sigma\,\Delta y &= f && \text{in } Q, \\ \partial_t p + G'(y)p + \sigma\,\Delta p + \alpha(y - y_d) &= 0 && \text{in } Q, \\ \nu u - p &= 0 && \text{on } \Sigma, \\ -\frac{\partial y}{\partial n} = u,\ -\frac{\partial p}{\partial n} &= 0 && \text{on } \Sigma. \end{aligned} \tag{5.175}$$

In both cases we have the initial condition $y(\mathbf{x},0) = y_0(\mathbf{x})$ for the state variable (evolving forward in time). The terminal condition for the adjoint variable (evolving backward in time) is given by

$$p(\mathbf{x},T) = \beta(y(\mathbf{x},T) - y_T(\mathbf{x})). \tag{5.176}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Notice that the optimality systems above are unusual in scientific computing because of the parabolic equations with opposite time orientation and the terminal condition.

We assume sufficient regularity of the data, y_d, y_T, such that these functions are properly approximated by their values at grid points. We use the finite difference framework of Chapter 3. We obtain the following

$$\begin{aligned} -\partial_t^+ y_h^m + G(y_h^m) + \sigma \Delta_h y_h^m &= u_h^m, \\ \partial_t^- p_h^m + G'(y_h^m) p_h^m + \sigma \Delta_h p_h^m + \alpha(y_h^m - y_{dh}^m) &= 0, \\ \nu u_h^m - p_h^m &= 0. \end{aligned} \tag{5.177}$$

The discretization of (5.175) is similar, requiring, in addition, specification of the approximation of the Neumann boundary conditions. This is done by considering the optimality system on the boundary and discretizing the boundary derivative using second-order centered finite differences to eliminate the (ghost) variables outside of the domain.

We recall that under suitable conditions the following estimates are obtained

$$||y_h - y||_0 \le c\,h^2, \quad ||p_h - p||_0 \le c\,h^2, \quad \text{and } ||u_h - u||_0 \le c\,h^2,$$

assuming there exist positive constants $c_1 \le c_2$ such that $c_1 h^2 \le \delta t \le c_2 h^2$.

Two CSMG Smoothing Strategies: Linear Case

In this section, we discuss a CSMG scheme to solve reaction-diffusion optimal control problems approximated by finite differences and backward Euler schemes. The space-time CSMG schemes discussed in this section are formulated on the entire space-time cylinder where the parabolic optimality systems are defined. We consider L grid levels indexed by $k = 1, \ldots, L$, where $k = L$ refers to the finest grid. The mesh of level k is denoted by $Q_k = Q_{h_k, \delta t_k}$, where $h_k = h_1/2^{k-1}$ and $\delta t_k = \delta t$. This choice corresponds to semicoarsening in space. This choice appears to be the most appropriate for our purpose. However, other choices are possible [49], such as standard coarsening, which results in being robust in the control of transient, i.e., short, time intervals.

In correspondence of semicoarsening in space, no interpolation and restriction in time are needed. However, other choices of intergrid transfer strategies are possible; see [202]. In our implementation, we choose I_k^{k-1} to be the full-weighted restriction operator [340] in space with no averaging in the time direction. The mirrored version of this operator applies also to the boundary points. The prolongation I_{k-1}^k is defined by bilinear interpolation in space.

Now, we discuss the design of two robust collective smoothing schemes for solving (5.174). For simplicity of illustration, we first consider the linear case with $G = 0$ and eliminate the control variable by means of the optimality condition $\nu u_h^m - p_h^m = 0$. We have

$$\begin{aligned} -\left[1 + 4\sigma\gamma\right] y_{i\,j\,m} + \sigma\gamma \left[y_{i+1\,j\,m} + y_{i-1\,j\,m} + y_{i\,j+1\,m} + y_{i\,j-1\,m}\right] + y_{i\,j\,m-1} \\ - \frac{\delta t}{\nu}\, p_{i\,j\,m} = 0, \quad 2 \le m \le N_t + 1, \end{aligned} \tag{5.178}$$

$$\begin{aligned} -\left[1 + 4\sigma\gamma\right] p_{i\,j\,m} + \sigma\gamma \left[p_{i+1\,j\,m} + p_{i-1\,j\,m} + p_{i\,j+1\,m} + p_{i\,j-1\,m}\right] + p_{i\,j\,m+1} \\ + \delta t\, \alpha\, (y_{i\,j\,m} - y_{d\,i\,j\,m}) = 0, \quad 1 \le m \le N_t. \end{aligned} \tag{5.179}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Let us define a collective iteration step which is applied at any space-time grid point to update $w_{ijm} = (y_{ijm}, p_{ijm})$. For this purpose consider (5.178) and (5.179) for the two variables y_{ijm} and p_{ijm} at the grid point ijm. We can refer to the left-hand sides of (5.178) and (5.179) as the negatives of the residuals $r_y(w_{ijm})$ and $r_p(w_{ijm})$, respectively. A step of a collective smoothing iteration at this point consists of a local update given by

$$\begin{pmatrix} y \\ p \end{pmatrix}^{(1)}_{ijm} = \begin{pmatrix} y \\ p \end{pmatrix}^{(0)}_{ijm} + \begin{bmatrix} -(1+4\sigma\gamma) & -\delta t/\nu \\ \delta t\,\alpha & -(1+4\sigma\gamma)+\delta t \end{bmatrix}^{(0)\,-1}_{ijm} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ijm}, \tag{5.180}$$

where r_y and r_p denote the residuals at ijm prior to the update. While a sweep of this smoothing iteration can be performed in any ordering of i, j, the problem of how to proceed along time direction arises.

To solve this problem the first vector component of (5.180) marching in the forward direction is used to update the state variable and the adjoint variable p is being updated using the second component of (5.180) marching backward in time. In this way a robust iteration is obtained given by the following algorithm; see, e.g., [57, 58, 49].

ALGORITHM 5.10. Time-splitted CGS iteration: linear case (TS-CGS).

1. Set the starting approximation.

2. For $m = 2, \ldots, N_t$ do

3. For ij in, e.g., lexicographic order do

$$y^{(1)}_{ijm} = y^{(0)}_{ijm} + \frac{[-(1+4\sigma\gamma)]\,r_y(w) + \frac{\delta t}{\nu} r_p(w)}{[-(1+4\sigma\gamma)]^2 + \frac{\delta t^2}{\nu}\alpha}\Big|^{(0)}_{ijm},$$

$$p^{(1)}_{ij\,N_t-m+2} = p^{(0)}_{ij\,N_t-m+2} + \frac{[-(1+4\sigma\gamma)]\,r_p(w) - \delta t\,\alpha\, r_y(w)}{[-(1+4\sigma\gamma)]^2 + \frac{\delta t^2}{\nu}\alpha}\Big|^{(0)}_{ij\,N_t-m+2};$$

4. end.

As in the elliptic case, the TS-CGS scheme applies with few modifications to the case of boundary control [46]. Results of local Fourier analysis, discussed below, show that the TS-CGS scheme has good smoothing properties, independently of the value of ν.

In the regime of small σ (or γ), however, the TS-CGS iteration cannot provide robust smoothing because of lack of strong coupling in the space directions. To overcome this problem, block relaxation of the variables that are strongly connected should be performed. For small σ (or γ) this means solving for the pairs of state and adjoint variables along the time direction for each space coordinate.

To describe this procedure, consider the discrete optimality system (5.178)–(5.179) at any i, j and for all time steps. Thus for each spatial grid point i, j a block-tridiagonal system is obtained, where each block is a 2×2 matrix corresponding to the pair (y, p).

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

This block-tridiagonal system has the following form

$$M = \begin{bmatrix} A_2 & C_2 & & & & \\ B_3 & A_3 & C_3 & & & \\ & B_4 & A_4 & C_4 & & \\ & & & & C_{N_t} & \\ & & & B_{N_t+1} & A_{N_t+1} & \end{bmatrix}. \tag{5.181}$$

Centered at t_m, the entries B_m, A_m, and C_m refer to the variables (y, p) at t_{m-1}, t_m, and t_{m+1}, respectively. The block A_m, $m = 2, \ldots, N_t$, is given by

$$A_m = \begin{bmatrix} -(1+4\sigma\gamma) & -\frac{\delta t}{\nu} \\ \delta t\,\alpha & -(1+4\sigma\gamma) \end{bmatrix}, \tag{5.182}$$

where all functions within the brackets [ ] are evaluated at t_m. Correspondingly, the B_m and C_m blocks are given by

$$B_m = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad C_m = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.183}$$

Clearly, for each time step, the variables neighboring the point ij are taken as constant and contribute to the right-hand side of the system.

It remains to discuss the block A_{N_t+1} for $\beta \neq 0$. At $t_m = T$, we have the terminal condition (5.176), which we rewrite as

$$\beta\,(y_h^m - y_{Th}^m) - p_h^m = 0, \quad m = N_t + 1.$$

Thus, the block A_{N_t+1} is given by

$$A_{N_t+1} = \begin{bmatrix} -(1+4\sigma\gamma) & -\frac{\delta t}{\nu} \\ \beta & -1 \end{bmatrix}. \tag{5.184}$$

For each i, j we have to solve a tridiagonal system $Mw = r$, where $w = (y_h^2, p_h^2, \ldots, y_h^{N_t+1}, p_h^{N_t+1})$ and $r = (r_y(w^2), r_p(w^2), \ldots, r_y(w^{N_t+1}), r_p(w^{N_t+1}))$. In particular we have $r_p(w^{N_t+1}) = p_h^{N_t+1} - \beta\,(y_h^{N_t+1} - y_{Th}^{N_t+1})$. Block-tridiagonal systems can be solved efficiently with $\mathcal{O}(N_t)$ effort. A block-tridiagonal solver is given in [61]. Summarizing, our collective t-line relaxation is given by the following algorithm [61, 62].

ALGORITHM 5.11. Time-line CGS iteration (TL-CGS).

1. Set the starting approximation.
2. For ij in, e.g., lexicographic order do

$$\begin{pmatrix} y \\ p \end{pmatrix}_{ij}^{(1)} = \begin{pmatrix} y \\ p \end{pmatrix}_{ij}^{(0)} + M^{-1} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ij};$$

3. end.

Also in this case r_y and r_p denote the residuals at i, j and for all m prior to the update. Since the solution in time is exact, no time splitting is required.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Two CSMG Smoothing Strategies: Nonlinear Case

In this section, we consider the nonlinear optimality system (5.177) in expanded form. Also in this case, we assume no constraints on the control, and therefore we can eliminate the control variable by means of the optimality condition $\nu u_h^m - p_h^m = 0$. We obtain

$$-\big[1+4\sigma\gamma\big]y_{ijm}+\sigma\gamma\big[y_{i+1jm}+y_{i-1jm}+y_{ij+1m}+y_{ij-1m}\big]+y_{ijm-1} \\ +\delta t\, G_\delta(y_{ijm})-\frac{\delta t}{\nu}\,p_{ijm}=\delta t\, f_{ijm},\quad 2\le m\le N_t+1, \tag{5.185}$$

$$-\big[1+4\sigma\gamma\big]p_{ijm}+\sigma\gamma\big[p_{i+1jm}+p_{i-1jm}+p_{ij+1m}+p_{ij-1m}\big]+p_{ijm+1} \\ +\delta t\, G'_\delta(y_{ijm})\,p_{ijm}+\delta t\,\alpha\,(y_{ijm}-y_{dijm})=0,\quad 1\le m\le N_t. \tag{5.186}$$

In case of terminal observation, at $t_m = T$ we have (5.176) in place of (5.186).

First, we define a pointwise smoothing scheme. Consider a collective Gauss–Seidel (–Newton) step which is applied at each space-time grid point to update $w_{ijm} = (y_{ijm}, p_{ijm})$. For this purpose consider (5.185) and (5.186) for the two variables y_{ijm} and p_{ijm} at the grid point ijm. We can refer to the left-hand sides of (5.185) and (5.186) as the negatives of the residuals $r_y(w_{ijm})$ and $r_p(w_{ijm})$, respectively. A step of a collective smoothing iteration at this point consists of a local (Newton) update given by

$$\begin{pmatrix} y \\ p \end{pmatrix}^{(1)}_{ijm} = \begin{pmatrix} y \\ p \end{pmatrix}^{(0)}_{ijm} + \begin{bmatrix} -(1+4\sigma\gamma)+\delta t\, G'_\delta & -\delta t/\nu \\ \delta t(\alpha+G''_\delta\, p) & -(1+4\sigma\gamma)+\delta t\, G'_\delta \end{bmatrix}^{(0)\,-1}_{ijm} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ijm}, \tag{5.187}$$

where r_y and r_p denote the residuals at ijm prior to the update. We need to take into account the opposite time orientation of the state equation and of the adjoint equation. For this purpose, to update the state variable we use the first vector component of (5.187) marching in the forward direction and the adjoint variable p is being updated using the second component of (5.187) marching backward in time. In this way a robust iteration is obtained given by the following nonlinear version of the TS-CGS scheme.

ALGORITHM 5.12. Time-splitted CGS iteration: nonlinear case (TS-CGS).

1. Set the starting approximation.
2. For $m = 2,\dots,N_t$ do
3. For ij in, e.g., lexicographic order do

$$y^{(1)}_{ijm} = y^{(0)}_{ijm} + \frac{[-(1+4\sigma\gamma)+\delta t G'_\delta]r_y(w)+\frac{\delta t}{\nu}r_p(w)}{[-(1+4\sigma\gamma)+\delta t G'_\delta]^2+\frac{\delta t^2}{\nu}(\alpha+G''_\delta p)}\Big|^{(0)}_{ijm},$$

$$p^{(1)}_{ijN_t-m+2} = p^{(0)}_{ijN_t-m+2} + \frac{[-(1+4\sigma\gamma)+\delta t G'_\delta]r_p(w)-\delta t(\alpha+G''_\delta p)r_y(w)}{[-(1+4\sigma\gamma)+\delta t G'_\delta]^2+\frac{\delta t^2}{\nu}(\alpha+G''_\delta p)}\Big|^{(0)}_{ijN_t-m+2};$$

4. end.

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

As for the linear case, in the regime of small σ (or γ), the TS-CGS iteration cannot provide robust smoothing because the coupling in the space direction becomes weak. To overcome this problem, we develop a time-line block-Newton relaxation of the state and adjoint variables.

To describe this block Gauss–Seidel–Newton procedure, consider the discrete optimality system (5.177) at any i, j and for all time steps. For simplicity, we use the optimality condition to eliminate the control variable. Thus for each spatial grid point i, j a block-tridiagonal system is obtained, where each block is a 2×2 matrix corresponding to the pair (y, p). This block-tridiagonal system has the following form

$$M = \begin{bmatrix} A_2 & C_2 & & & & \\ B_3 & A_3 & C_3 & & & \\ & B_4 & A_4 & C_4 & & \\ & & & & C_{N_t} & \\ & & & B_{N_t+1} & A_{N_t+1} & \end{bmatrix}. \tag{5.188}$$

Centered at t_m, the entries B_m, A_m, and C_m refer to the variables (y, p) at t_{m-1}, t_m, and t_{m+1}, respectively. The block A_m, $m = 2, \ldots, N_t$, is given by

$$A_m = \begin{bmatrix} -(1+4\sigma\gamma)+\delta t G'_\delta & -\frac{\delta t}{\nu} \\ \delta t(\alpha + G''_\delta p) & -(1+4\sigma\gamma)+\delta t G'_\delta \end{bmatrix}, \tag{5.189}$$

where all functions within the brackets [ ] are evaluated at t_m. Correspondingly, the B_m and C_m blocks are given by

$$B_m = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad C_m = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.190}$$

Clearly, for each time step, the variables neighboring the point ij are taken as constant and contribute to the right-hand side of the system.

It remains to discuss the block A_{N_t+1} for $\beta \neq 0$. At $t_m = T$, we have the terminal condition (5.176), which we rewrite as

$$\beta(y_h^m - y_{Th}^m) - p_h^m = 0, \quad m = N_t + 1.$$

Thus, the block A_{N_t+1} is given by

$$A_{N_t+1} = \begin{bmatrix} -(1+4\sigma\gamma)+\delta t G' & -\frac{\delta t}{\nu} \\ \beta & -1 \end{bmatrix}. \tag{5.191}$$

For each i, j we have to solve a tridiagonal system $Mw = r$, where $w = (y_h^2, p_h^2, \ldots, y_h^{N_t+1}, p_h^{N_t+1})$ and $r = (r_y(w^2), r_p(w^2), \ldots, r_y(w^{N_t+1}), r_p(w^{N_t+1}))$. In particular we have $r_p(w^{N_t+1}) = p_h^{N_t+1} - \beta(y_h^{N_t+1} - y_{Th}^{N_t+1})$. Block-tridiagonal systems can be solved efficiently with $\mathcal{O}(N_t)$ effort. A block-tridiagonal solver is given in [61]. Clearly, the nonlinear version of the TL-CGS relaxation is formally identical to that of the linear case.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

TG Fourier Analysis: Linear Parabolic Case

In this section, we illustrate the TG local Fourier analysis of the space-time CSMG scheme with TS-CGS and TL-CGS smoothing applied to a linear parabolic optimal control problem. For the analysis that follows, we consider a linear case with linear reaction $G_\delta(y) = \delta\, y$ and distributed control with the replacement $u = p/\nu$. We denote the parabolic optimality system with $A\,w = f$, where $w = (y, p)$, and assume an infinite grid and one space dimension. On the fine grid, consider the Fourier components $\phi(\boldsymbol{j},\boldsymbol{\theta}) = e^{i\,\boldsymbol{j}\cdot\boldsymbol{\theta}}$, where i is the imaginary unit, $\boldsymbol{j} = (j_x, j_t) \in \boldsymbol{Z} \times \boldsymbol{Z}$, $\boldsymbol{\theta} = (\theta_x, \theta_t) \in [-\pi, \pi)^2$, and $\boldsymbol{j}\cdot\boldsymbol{\theta} = j_x\theta_x + j_t\theta_t$.

In a semicoarsening setting, the frequency domain is spanned by the following two sets of frequencies:

$$\boldsymbol{\theta}^{(0,0)} := (\theta_x, \theta_t) \quad \text{and} \quad \boldsymbol{\theta}^{(1,0)} := (\overline{\theta}_x, \theta_t),$$

where $(\theta_x, \theta_t) \in ([-\pi/2, \pi/2) \times [-\pi, \pi))$ and $\overline{\theta}_x = \theta_x - sign(\theta_x)\pi$. The components $\phi(\cdot, \boldsymbol{\theta}^{\boldsymbol{\alpha}})$ are called harmonics. The first harmonics $\phi(\cdot, \boldsymbol{\theta}^{(0,0)})$ represents LF components in space. The second harmonics $\phi(\cdot, \boldsymbol{\theta}^{(1,0)})$ contains the HF components in space direction. Both have all frequency components in time direction. Using semicoarsening, we have that $\phi(\boldsymbol{j}, \boldsymbol{\theta}^{(0,0)}) = \phi(\boldsymbol{j}, \boldsymbol{\theta}^{(1,0)})$ on the coarse grid.

The action of the multigrid scheme is to reduce the HF error components by applying the smoothing operator S_k and to reduce the LF error components by coarse-grid correction given by

$$CG_k^{k-1} = [I_k - I_{k-1}^k\,(A_{k-1})^{-1}\, I_k^{k-1}\, A_k].$$

Denote $E_k^\theta = \text{span}[\phi_k(\cdot, \boldsymbol{\theta}^{\boldsymbol{\alpha}}) : \boldsymbol{\alpha} \in \{(0,0),(1,0)\}]$. Under the assumption that all multigrid components are linear and that $(A_{k-1})^{-1}$ exists, we have a representation of the TG operator $\boldsymbol{TG}_k^{k-1}$ on the space $E_k^\theta \times E_k^\theta$ by a 4×4 matrix given by

$$\widehat{TG}_k^{k-1}(\boldsymbol{\theta}) = \hat{S}_k(\boldsymbol{\theta})^{\nu_2}\, \widehat{CG}_k^{k-1}(\boldsymbol{\theta})\, \hat{S}_k(\boldsymbol{\theta})^{\nu_1},$$

where the hat denotes the Fourier symbol [340] of the given operator.

To determine the explicit form of the operator symbols given above, consider the action of the operators on the couple $(\tilde{y}, \tilde{p}) \in E_k^\theta \times E_k^\theta$ for a given $(\boldsymbol{j})$, where

$$\tilde{y} = \sum_{\boldsymbol{\theta}} \sum_{p=0,1} Y_{\boldsymbol{\theta}}^{(p,0)}\, \phi_k(\boldsymbol{j}, \boldsymbol{\theta}^{(p,0)}) \quad \text{and} \quad \tilde{p} = \sum_{\boldsymbol{\theta}} \sum_{p=0,1} P_{\boldsymbol{\theta}}^{(p,0)}\, \phi_k(\boldsymbol{j}, \boldsymbol{\theta}^{(p,0)}), \tag{5.192}$$

where $\sum_{\boldsymbol{\theta}}$ denotes formal summation in $\boldsymbol{\theta} = (\theta_x, \theta_t) \in ([-\pi/2, \pi/2) \times [-\pi, \pi))$ and $\tilde{W}_{\boldsymbol{\theta}}^{\boldsymbol{\alpha}} = (\tilde{Y}_{\boldsymbol{\theta}}^{\boldsymbol{\alpha}}, \tilde{P}_{\boldsymbol{\theta}}^{\boldsymbol{\alpha}})$ are the corresponding Fourier coefficients.

In the Fourier space the action of one smoothing step can be expressed by $\tilde{W}_{\boldsymbol{\theta}}^{(new)} = \hat{S}(\boldsymbol{\theta})\, \tilde{W}_{\boldsymbol{\theta}}^{(old)}$, where $\hat{S}(\boldsymbol{\theta})$ is the Fourier symbol [340] of the smoothing iteration. This operator applies to the two equations (5.185)–(5.186) acting on LF and HF components. It has the following form

$$\hat{S}_k(\boldsymbol{\theta}) = \text{diag}\{\hat{s}(\boldsymbol{\theta}^{(0,0)}), \hat{s}(\boldsymbol{\theta}^{(1,0)})\},$$

where $\hat{s}(\boldsymbol{\theta})$ is the 2×2 Fourier symbol of the smoothing scheme for a generic $\boldsymbol{\theta}$.

A way to characterize the smoothing property of the operator S_k is to assume an ideal coarse-grid correction which annihilates the LF error components and leaves the HF error

components unchanged. That is, one defines the projection operator $\widehat{Q}_k^{k-1}$ on $E_k^\theta \times E_k^\theta$ by

$$\widehat{Q}_k^{k-1}(\boldsymbol{\theta}) = \begin{bmatrix} Q_k^{k-1}(\boldsymbol{\theta}) & 0 \\ 0 & Q_k^{k-1}(\boldsymbol{\theta}) \end{bmatrix}, \quad \text{where } Q_k^{k-1}(\boldsymbol{\theta}) = \begin{cases} \text{diag}\{0,0\} & \text{if } \boldsymbol{\theta} = \boldsymbol{\theta}^{(0,0)}, \\ \text{diag}\{1,1\} & \text{if } \boldsymbol{\theta} = \boldsymbol{\theta}^{(1,0)}. \end{cases}$$

In this framework the smoothing property of S_k is defined as follows

$$\mu = \max\{r(\widehat{Q}_k^{k-1}(\boldsymbol{\theta})\,\hat{S}_k(\boldsymbol{\theta})) : \boldsymbol{\theta} \in ([-\pi/2, \pi/2) \times [-\pi, \pi))\}, \tag{5.193}$$

where r is the spectral radius. Compare with (5.105) and with (5.9).

Now consider applying the TS-CGS step. We obtain

$$\begin{pmatrix} -(1+2\sigma\gamma-\delta t\delta)+\sigma\gamma e^{-i\theta_x} & -\frac{\delta t}{\nu} \\ \alpha\delta t & -(1+2\sigma\gamma-\delta t\delta)+\sigma\gamma e^{-i\theta_x} \end{pmatrix} \begin{pmatrix} \tilde{Y}_{\boldsymbol{\theta}}^{(new)} \\ \tilde{P}_{\boldsymbol{\theta}}^{(new)} \end{pmatrix}$$
$$= \begin{pmatrix} -(e^{-i\theta_t}+\sigma\gamma e^{i\theta_x}) & 0 \\ 0 & -(e^{i\theta_t}+\sigma\gamma e^{i\theta_x}) \end{pmatrix} \begin{pmatrix} \tilde{Y}_{\boldsymbol{\theta}}^{(old)} \\ \tilde{P}_{\boldsymbol{\theta}}^{(old)} \end{pmatrix}.$$

Hence

$$\hat{s}(\boldsymbol{\theta}) = \begin{pmatrix} -(1+2\sigma\gamma-\delta t\delta)+\sigma\gamma e^{-i\theta_x} & -\frac{\delta t}{\nu} \\ \alpha\delta t & -(1+2\sigma\gamma-\delta t\delta)+\sigma\gamma e^{-i\theta_x} \end{pmatrix}^{-1} \tag{5.194}$$
$$\times \begin{pmatrix} -(e^{-i\theta_t}+\sigma\gamma e^{i\theta_x}) & 0 \\ 0 & -(e^{i\theta_t}+\sigma\gamma e^{i\theta_x}) \end{pmatrix}.$$

Next consider the case of TL-CGS relaxation. The Fourier symbol of the smoothing operator is given by the following 2×2 matrix

$$\hat{s}(\boldsymbol{\theta}) = -(A + B\,e^{-i\theta_t} + C\,e^{i\theta_t} + \tilde{I}\,e^{-i\theta_x})^{-1}(\tilde{I}\,e^{i\theta_x}),$$

where

$$A = \begin{bmatrix} -(1+2\sigma\gamma-\delta t\delta) & -\frac{\delta t}{\nu} \\ \alpha\delta t & -(1+2\sigma\gamma-\delta t\delta) \end{bmatrix}, B_m = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \text{ and } C_m = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

and $\tilde{I} = \sigma\gamma\, I$, I being the 2×2 identity matrix.

To investigate the action of the coarse-grid correction, we consider a full-weighting restriction operator whose symbol is given by

$$\hat{I}_k^{k-1}(\boldsymbol{\theta}) = \frac{1}{2} \begin{bmatrix} (1+\cos(\theta_x)) & 0 & (1-\cos(\theta_x)) & 0 \\ 0 & (1+\cos(\theta_x)) & 0 & (1-\cos(\theta_x)) \end{bmatrix}. \tag{5.195}$$

For the linear prolongation operator we have $\hat{I}_{k-1}^k(\boldsymbol{\theta}) = \hat{I}_k^{k-1}(\boldsymbol{\theta})^T$. The symbol of the fine grid operator is

$$\widehat{A}_k(\boldsymbol{\theta}) = \begin{bmatrix} a_y(\boldsymbol{\theta}^{(0,0)}) & -\delta t/\nu & 0 & 0 \\ \alpha\delta t & a_p(\boldsymbol{\theta}^{(0,0)}) & 0 & 0 \\ 0 & 0 & a_y(\boldsymbol{\theta}^{(1,0)}) & -\delta t/\nu \\ 0 & 0 & \alpha\delta t & a_p(\boldsymbol{\theta}^{(1,0)}) \end{bmatrix}, \tag{5.196}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

where

$$a_y(\boldsymbol{\theta}) = 2\sigma\gamma\cos(\theta_x) - e^{-i\theta_t} - 2\sigma\gamma + \delta t\delta - 1 \text{ and } a_p(\boldsymbol{\theta}) = 2\sigma\gamma\cos(\theta_x) - e^{i\theta_t} - 2\sigma\gamma + \delta t\delta - 1.$$

The symbol of the coarse-grid operator follows

$$\widehat{A}_{k-1}(\boldsymbol{\theta}) = \begin{bmatrix} \sigma\gamma\cos(2\theta_x)/2 - e^{-i\theta_t} - \sigma\gamma/2 + \delta t\delta - 1 & -\delta t/\nu \\ \alpha\delta t & \sigma\gamma\cos(2\theta_x)/2 - e^{i\theta_t} - \sigma\gamma/2 + \delta t\delta - 1 \end{bmatrix}.$$

Notice that on the coarser grid δt remains unchanged while $\gamma \to \gamma/4$ by coarsening.

Based on the representation on TG_k^{k-1} by a 4×4 matrix $\widehat{TG}_k^{k-1}(\boldsymbol{\theta})$ we can calculate the TG convergence factor given by

$$\rho(TG_k^{k-1}) = \sup\left\{r\left(\widehat{TG}_k^{k-1}(\boldsymbol{\theta})\right) : \boldsymbol{\theta} \in ([-\pi/2,\pi/2)\times[-\pi,\pi))\right\},$$

which requires determination of the spectral radius of a 4×4 matrix. It results that μ and ρ are almost independent of the value of the weight ν and of the mesh parameter γ for a large range of choices of values of these parameters. For $\sigma = 0$ no spatial coupling is present and the TL-CGS scheme becomes an exact solver; i.e., $\mu = 0$ results. On the other hand, for moderate values of ν and corresponding to smaller values of σ the convergence factor of the TS-CGS multigrid scheme worsens. In Table 5.19, local Fourier analysis quantitative estimates of the convergence factor of TL-CGS and TS-CGS multigrid schemes are given. We see that both schemes result in convergence factors that are typical of multigrid schemes for Poisson problems.

Table 5.19. *The convergence factor ρ for TL-CGS and TS-CGS multigrid schemes ($\nu_1 = \nu_2 = 1$); $\delta t = 1/64$, $\delta = 0$, $\sigma = 1$, $\alpha = 1$.*

	TL-CGS			TS-CGS		
$\gamma \setminus \nu$	10^{-8}	10^{-6}	10^{-4}	10^{-8}	10^{-6}	10^{-4}
16	0.01	0.12	0.12	0.01	0.12	0.14
32	0.04	0.13	0.13	0.04	0.13	0.14
48	0.08	0.13	0.13	0.08	0.13	0.13
64	0.11	0.13	0.13	0.11	0.13	0.13

The CSMG Scheme for Parabolic Control-Constrained Problems and Higher-Order Time Discretization

Parabolic control problems result in very large-sized algebraic systems so that techniques are needed in order to reduce the computational time required to determine the optimal control function. A possible strategy is to construct reduced models via, e.g., proper orthogonal decomposition [227] or consider adaptivity [190, 253]. In this section, we consider the idea of reducing the size of the algebraic problems by considering uniform high-order discretization of the optimality system characterizing the optimal solution. In fact, the use of higher-order schemes allows us to attain the required accuracy with much coarser meshes, thus reducing considerably the size of the algebraic problems to be solved while keeping the computational cost at a minimum.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

In addition to higher-order time discretization, in the following we discuss the presence of constraints on the control and present generalization of the TS-CGS and TL-CGS smoothing to accommodate these constraints.

We consider the parabolic distributed control problem (5.171) and we require that $u \in U_{ad}$, where $U_{ad} \subset L^2(Q)$ represents the following closed and convex set of admissible controls

$$U_{ad} := \{u \in L^2(Q) : \underline{u}(\mathbf{x},t) \le u(\mathbf{x},t) \le \overline{u}(\mathbf{x},t) \text{ a.e. in } Q\}, \tag{5.197}$$

where $\underline{u}$ and $\overline{u}$ are elements of $L^\infty(Q)$. We take $g = 0$ and assume an initial condition $y_0 \in H_0^1(\Omega)$.

Recall the following first-order optimality system

$$\begin{aligned} -\partial_t y + G(y) + \sigma \Delta y &= f + u \quad \text{in } Q, \\ \partial_t p + G'(y)p + \sigma \Delta p + \alpha(y - y_d) &= 0 \quad \text{in } Q, \\ y = 0,\ p &= 0 \quad \text{on } \Sigma, \\ (\nu u - p, v - u) &\ge 0 \quad \forall v \in U_{ad}, \end{aligned} \tag{5.198}$$

with initial condition $y(x,0) = y_0(x)$ for the state variable (evolving forward in time). The terminal condition for the adjoint variable (evolving backward in time) is given by

$$p(x,T) = \beta(y(x,T) - y_T(x)). \tag{5.199}$$

The optimality system corresponding to $\nu = 0$ is given by (5.198) with the (inequality) optimality condition replaced with

$$p = \min\{0, p + u - \underline{u}\} + \max\{0, p + u - \overline{u}\} \qquad \text{in } Q.$$

Following [59, 147] and considering the linear case $G = 0$, one can prove that the solution to (5.198) with $\nu = 0$ exists and is unique.

Now, we consider the second-order backward differentiation (BDF2) formula together with the Crank–Nicolson (CN) method in order to obtain a second-order time discretization scheme; see Chapter 3.

We have the following discrete optimality system

$$\begin{aligned} -\partial_{BD}^{+} y_h^m + G(y_h^m) + \sigma \Delta_h y_h^m &= f_h^m + u_h^m, \\ \partial_{BD}^{-} p_h^m + G'(y_h^m)p_h^m + \sigma \Delta_h p_h^m + \alpha(y_h^m - y_{dh}^m) &= 0, \\ (\nu u_h^m - p_h^m, v_h^m - u_h^m) &\ge 0 \quad \forall v \in U_{ad}^h, \end{aligned} \tag{5.200}$$

where we assume sufficient regularity of the data, y_d, y_T, and f, such that these functions are properly approximated by their values at grid points.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

In order to develop the CSMG smoothing scheme, let us write the discretized optimality system (5.200), in expanded form, for a space-time grid point $(i\,j\,m)$. We have

$$\begin{aligned} -\left(\tfrac{3}{2}+4\sigma\gamma\right) y_{ijm} + \sigma\gamma\left[y_{i+1jm}+y_{i-1jm}+y_{ij+1m}+y_{ij-1m}\right] + 2y_{ijm-1} \\ -\tfrac{1}{2}y_{ijm-2} + \delta t\, G(y_{ijm}) - \delta t\, u_{ijm} = \delta t\, f_{ijm},\ \ 3 \le m \le N_t+1, \end{aligned} \tag{5.201a}$$

$$\begin{aligned} -\left(\tfrac{3}{2}+4\sigma\gamma\right) p_{ijm} + \sigma\gamma\left[p_{i+1jm}+p_{i-1jm}+p_{ij+1m}+p_{ij-1m}\right] + 2p_{ijm+1} \\ -\tfrac{1}{2}p_{ijm+2} + \delta t\, G'(y_{ijm})p_{ijm} + \alpha\delta t(y_{ijm}-y_{dijm}) = 0,\ \ 1 \le m \le N_t-1, \end{aligned} \tag{5.201b}$$

$$\nu(u_{ijm}-p_{ijm})\cdot(v_{ijm}-u_{ijm}) \ge 0 \quad \forall v_h \in U^h_{ad}. \tag{5.201c}$$

In the case of terminal observation, at $t_m = T$ $(m = N_t+1)$, we have (5.199) in place of (5.201b).

Further, since we use the CN method to calculate the required first steps to initialize the BDF2 method, we also write the expanded form of (3.26) and (3.27) to analyze the cases $m = 2$, corresponding to the instant δt, and $m = N_t$, corresponding to the instant $T - \delta t$. Therefore, for the case $m = 2$ we approximate the state variable with the following CN discretization

$$\begin{aligned} -(1+2\sigma\gamma)\, y_{ijm} + \tfrac{\sigma\gamma}{2}\left[y_{i+1jm}+y_{i-1jm}+y_{ij+1m}+y_{ij-1m}\right] \\ +(1-2\sigma\gamma)\, y_{ijm-1} + \tfrac{\delta t}{2}G(y_{ijm}) - \tfrac{\delta t}{2}u_{ijm} \\ +\tfrac{\sigma\gamma}{2}\left[y_{i+1jm-1}+y_{i-1jm-1}+y_{ij+1m-1}+y_{ij-1m-1}\right] \\ +\tfrac{\delta t}{2}G(y_{ijm-1}) - \tfrac{\delta t}{2}u_{ijm-1} = \tfrac{\delta t}{2}(f_{ijm-1}+f_{ijm}),\ \ m=2, \end{aligned} \tag{5.202}$$

and we approximate the corresponding adjoint variable p_{ij2} with (5.201b). Moreover, for the case $m = N_t$ we approximate the corresponding state variable y_{ijN_t} with (5.201a), and we calculate the adjoint variable with the following CN discretization

$$\begin{aligned} -(1+2\sigma\gamma)\, p_{ijm} + \tfrac{\sigma\gamma}{2}\left[p_{i+1jm}+p_{i-1jm}+p_{ij+1m}+p_{ij-1m}\right] \\ +(1-2\sigma\gamma)\, p_{ijm+1} + \tfrac{\delta t}{2}G'(y_{ijm})p_{ijm} + \tfrac{\alpha\delta t}{2}(y_{ijm}-y_{dijm}) \\ +\tfrac{\sigma\gamma}{2}\left[p_{i+1jm+1}+p_{i-1jm+1}+p_{ij+1m+1}+p_{ij-1m+1}\right] \\ +\tfrac{\delta t}{2}G'(y_{ijm+1})p_{ijm+1} + \tfrac{\alpha\delta t}{2}(y_{ijm+1}-y_{dijm+1}) = 0,\ m = N_t. \end{aligned} \tag{5.203}$$

In this case, the approximation of the control is obtained equally as in (5.201c).

Now, we can proceed with the construction of a pointwise smoothing scheme following the same lines of development of the TS-CGS scheme in case of backward Euler discretization. Consider the optimality system (5.201) at the space-time grid points $(i\,j\,m)$ for $m = 3,\ldots,N_t-1$. A similar discussion follows for $m = 2$ and $m = N_t$, where the BDF2 and the CN discretization both appear in the optimality system. We start by defining the following quantities

$$\begin{aligned} a &= \left(\tfrac{3}{2}+4\sigma\gamma\right), \\ S_{ijm} &= \sigma\gamma\left[y_{i+1\,jm}+y_{i-1\,jm}+y_{i\,j+1\,m}+y_{i\,j-1\,m}\right] + 2y_{ijm-1} - \tfrac{1}{2}y_{ijm-2} - \delta t\, f_{i\,j\,m}, \\ R_{ijm} &= \sigma\gamma\left[p_{i+1\,jm}+p_{i-1\,jm}+p_{i\,j+1\,m}+p_{i\,j-1\,m}\right] + 2p_{ijm+1} - \tfrac{1}{2}p_{ijm+2} - \delta t\,\alpha\, y_{di\,j\,m}. \end{aligned}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

With this notation, the optimality system (5.201) at (i, j, m) can be written as follows

$$-a\, y_{ijm} + S_{ijm} + \delta t G(y_{ijm}) - \delta t u_{ijm} = 0, \tag{5.204a}$$

$$-a\, p_{ijm} + R_{ijm} + \delta t G'(y_{ijm}) p_{ijm} + \alpha \delta t\, y_{ijm} = 0, \tag{5.204b}$$

$$(\nu u_{ijm} - p_{ijm})(v_{ijm} - u_{ijm}) \geq 0 \quad \forall v_h \in U_{ad,h}. \tag{5.204c}$$

This is a nonlinear problem that includes an inequality constraint. To solve this problem, we generalize the nonlinear version of the TS-CGS scheme discussed in the case of backward discretization.

Consider (5.204a) and (5.204b). The Jacobian of these two equations is given by

$$J_{ijm} := \begin{pmatrix} -a + \delta t G' & 0 \\ \delta t(\alpha + G'' p_{ijm}) & -a + \delta t G' \end{pmatrix}$$

and its inverse is

$$J_{ijm}^{-1} = \frac{1}{(-a + \delta t G')^2} \begin{pmatrix} -a + \delta t G' & 0 \\ -\delta t(\alpha + G'' p_{ijm}) & -a + \delta t G' \end{pmatrix}. \tag{5.205}$$

Notice that, in the case of nonmonotone nonlinearity (e.g., singular control problems), we should choose δt sufficiently small to guarantee that $(-a + \delta t G')^2 \neq 0$.

Now, for a given u_{ijm}, a classical local Newton update for the auxiliary state and adjoint variables $\widehat{y}_{ijm}$ and $\widehat{p}_{ijm}$ is given by

$$\begin{pmatrix} \widehat{y} \\ \widehat{p} \end{pmatrix}_{ijm} = \begin{pmatrix} y \\ p \end{pmatrix}_{ijm} + J_{ijm}^{-1} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ijm}, \tag{5.206}$$

where $(r_y)_{ijm}$ and $(r_p)_{ijm}$ denote the residuals of (5.204a) and (5.204b), respectively. In the case of the BDF2 discretization, these residuals are given by

$$\begin{aligned} (r_y)_{ijm} &= a\, y_{ijm} - S_{ijm} - \delta t G(y_{ijm}) + \delta t u_{ijm} && \text{for } m = 3, \ldots, N_t + 1, \\ (r_p)_{ijm} &= a\, p_{ijm} - R_{ijm} - \delta t G'(y_{ijm}) p_{ijm} - \alpha \delta t\, y_{ijm} && \text{for } m = 1, \ldots, N_t - 1. \end{aligned}$$

Since $(r_y)_{ijm}$ depends explicitly on u_{ijm}, we can write $\widehat{p}_{ijm}$ as a function of u_{ijm} as follows

$$\begin{aligned} \widehat{p}_{ijm}(u_{ijm}) = p_{ijm} &+ \frac{-\delta t(\alpha + G'' p_{ijm})[a\, y_{ijm} - S_{ijm} - \delta t G(y_{ijm}]}{(-a + \delta t G')^2} \\ &+ \frac{a\, p_{ijm} - R_{ijm} - \delta t G'(y_{ijm}) p_{ijm} - \alpha \delta t\, y_{ijm}}{(-a + \delta t G')} \\ &- \frac{\delta t^2 (\alpha + G'' p) u_{ijm}}{(-a + \delta t G')^2}. \end{aligned} \tag{5.207}$$

We use $\widehat{p}_{ijm}$ in order to obtain the update for the control variable u_{ijm}. Let us recall that the gradient of the objective functional is given as $\nabla \hat{J}(u) = \nu u - p$. Therefore, from

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

$\nu\widetilde{u}_{ijm} - \widehat{p}_{ijm} = 0$, we obtain the auxiliary variable

$$\widetilde{u}_{ijm} = \left(\nu + \frac{\delta t^2(\alpha + G'' p)u_{ijm}}{(-a + \delta t G')^2}\right)^{-1} \times \left[p_{ijm} + \frac{-\delta t(\alpha + G'' p_{ijm})[a\, y_{ijm} - S_{ijm} - \delta t G(y_{ijm})]}{(-a + \delta t G')^2} + \frac{a\, p_{ijm} - R_{ijm} - \delta t G'(y_{ijm}) p_{ijm} - \alpha \delta t\, y_{ijm}}{(-a + \delta t G')}\right]. \tag{5.208}$$

Then, the new value for u_{ijm} resulting from the smoothing step is obtained by projection as follows

$$u_{ijm} = \begin{cases} \overline{u}_{ijm} & \text{if} \quad \tilde{u}_{ijm} > \overline{u}_{ijm}, \\ \widetilde{u}_{ijm} & \text{if} \quad \underline{u}_{ijm} \le \widetilde{u}_{ijm} \le \overline{u}_{ijm}, \\ \underline{u}_{ijm} & \text{if} \quad \tilde{u}_{ijm} < \underline{u}_{ijm}. \end{cases} \tag{5.209}$$

With u_{ijm} given, we can use (5.206) to obtain new values for y_{ijm} and p_{ijm}.

An iteration step of this smoothing scheme can be performed in any ordering for the spatial variables, and in order to take into account the opposite time orientation of the state and the adjoint equations we update the state variable y using the first vector component of (5.206) marching in the forward direction, and the adjoint variable p is being updated using the second component of (5.206) marching backward in time. In this way a robust iteration is obtained. Let us recall that the calculation of the initialization steps y_{ij2} and p_{ijN_t} are carried out in the same way, but using the combination of the CN scheme with the BDF2 method, as described above.

ALGORITHM 5.13. Projected time-splitted CGS iteration (P-TS-CGS).

1. Set the starting approximation: calculate y_{ij2}, p_{ij2}, u_{ij2}, y_{ijN_t}, p_{ijN_t}, and u_{ijN_t}.
2. For ij in, e.g., lexicographic order do
3. For $m = 3, \ldots, N_t - 1$: compute $(r_y)_{ijm}$, $\tilde{u}_{ijm}$ and projection u_{ijm}. Then, the state update is given by
$$y^{(1)}_{ijm} = y^{(0)}_{ijm} + \frac{(r_y)_{ijm}}{(-a + \delta t G')}.$$
4. For $m' = N_t - 1, \ldots, 3$ (backward): compute $(r_y)_{ijm'}$, $(r_p)_{ijm'}$, $\tilde{u}_{ijm'}$ and projection $u_{ijm'}$. Then, the adjoint update is given by
$$p^{(1)}_{ijm'} = p^{(0)}_{ijm'} + \frac{(-a + \delta t G')(r_p)_{ijm'} - \delta t(\alpha + G'' p)(r_y)_{ijm'}}{(-a + \delta t G')^2}.$$
5. end.

In the control-unconstrained case, the iteration above applies without projection and it becomes equivalent to the time-splitted CGS iteration.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

In order to construct a robust block time-line smoothing scheme, we proceed in a way similar to that followed for the pointwise approach in order to obtain an approximation for the controls u_{ijm}. Thus, we consider the residual equations $(r_y)_{ijm} = 0$ and $(r_p)_{ijm} = 0$ for all $m = 1, \ldots, N_t + 1$ at any fixed (i, j). The solution of this problem provides the mapping $u_{ijm} \to y_{ijm}$ and $u_{ijm} \to p_{ijm}$, and by requiring us to satisfy the (unconstrained) optimality condition, we obtain $\tilde{u}_{ijm}$ followed by projection, thus obtaining a new approximation for the control and consequently the updates for the state and adjoint variables.

Let us recall that the residuals $(r_y)_{ijm}$ are defined as follows

$$
\begin{aligned}
(r_y)_{ijm} &= y_{ijm} - \psi_{ijm} \quad \text{for } m = 1, \\
(r_y)_{ijm} &= (1 + 2\sigma\gamma)\, y_{ijm} - \tfrac{\sigma\gamma}{2}\left[y_{i+1jm} + y_{i-1jm} + y_{ij+1m} + y_{ij-1m}\right] - (1 - 2\sigma\gamma)\, y_{ijm-1} \\
&\quad - \tfrac{\delta t}{2} G(y_{ijm}) + \tfrac{\delta t}{2} u_{ijm} - \tfrac{\sigma\gamma}{2}\left[y_{i+1jm-1} + y_{i-1jm-1} + y_{ij+1m-1} + y_{ij-1m-1}\right] \\
&\quad - \tfrac{\delta t}{2} G(y_{ijm-1}) + \tfrac{\delta t}{2} u_{ijm-1} + \tfrac{\delta t}{2}(f_{ijm-1} + f_{ijm}) \quad \text{for } m = 2, \\
(r_y)_{ijm} &= \left(\tfrac{3}{2} + 4\sigma\gamma\right) y_{ijm} - \sigma\gamma\left[y_{i+1jm} + y_{i-1jm} + y_{ij+1m} + y_{ij-1m}\right] - 2y_{ijm-1} \\
&\quad + \tfrac{1}{2} y_{ijm-2} - \delta t\, G(y_{ijm}) + \delta t\, u_{ijm} + \delta t\, f_{ijm} \quad \text{for } 3 \le m \le N_t + 1.
\end{aligned}
$$

Note that the definition of $(r_y)_{ij1}$ is introduced considering a function ψ_h^m associated with the initial condition y_0. The introduction of this residual helps us to completely describe the system of equations in the time interval $[0, T]$, making it possible to obtain a desirable structure for the matrices involved in the solution of the equation.

Furthermore, the residuals $(r_p)_{ijm}$ are given by

$$
\begin{aligned}
(r_p)_{ijm} &= \left(\tfrac{3}{2} + 4\sigma\gamma\right) p_{ijm} - \sigma\gamma\left[p_{i+1jm} + p_{i-1jm} + p_{ij+1m} + p_{ij-1m}\right] - 2p_{ijm+1} \\
&\quad + \tfrac{1}{2} p_{ijm+2} - \delta t\, G'(y_{ijm}) p_{ijm} - \alpha\delta t(y_{ijm} - y_{dijm}) = 0 \quad \text{for } 1 \le m \le N_t + 1, \\
(r_p)_{ijm} &= (1 + 2\sigma\gamma)\, p_{ijm} - \tfrac{\sigma\gamma}{2}\left[p_{i+1jm} + p_{i-1jm} + p_{ij+1m} + p_{ij-1m}\right] \\
&\quad - (1 - 2\sigma\gamma)\, p_{ijm+1} - \tfrac{\delta t}{2} G'(y_{ijm}) p_{ijm} - \tfrac{\alpha\delta t}{2}(y_{ijm} - y_{dijm}) \\
&\quad - \tfrac{\sigma\gamma}{2}\left[p_{i+1jm+1} + p_{i-1jm+1} + p_{ij+1m+1} + p_{ij-1m+1}\right] \\
&\quad - \tfrac{\delta t}{2} G'(y_{ijm+1}) p_{ijm+1} - \tfrac{\alpha\delta t}{2}(y_{ijm+1} - y_{dijm+1}) = 0 \quad \text{for } m = N_t, \\
(r_p)_{ijm} &= \beta(y_{ijm} - y_{Tijm}) - p_{ijm} \quad \text{for } m = N_t + 1.
\end{aligned}
$$

Note that $(r_p)_{ijN_t+1}$ corresponds to the terminal condition (5.199).

To describe the block Gauss–Seidel procedure, consider the residual equations at any fixed i, j and for all time steps. For each spatial grid point i, j, the pair of state and adjoint equations corresponding to (y, p) at a given m corresponds to five 2×2 blocks for the pairs $(y, p)_{m-2}$, $(y, p)_{m-1}$, $(y, p)_m$, $(y, p)_{m+1}$, $(y, p)_{m+2}$. Considering all time steps, we obtain

a block-pentadiagonal system. This system has the following form

$$M = \begin{bmatrix} A_1 & D_1 & E_1 & & & & & \\ C_2 & A_2 & D_2 & E_2 & & & & \\ B_3 & C_3 & A_3 & D_3 & E_3 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & B_{N_t-1} & C_{N_t-1} & A_{N_t-1} & D_{N_t-1} & E_{N_t-1} \\ & & & & B_{N_t} & C_{N_t} & A_{N_t} & D_{N_t} \\ & & & & & B_{N_t+1} & C_{N_t+1} & A_{N_t+1} \end{bmatrix}. \tag{5.210}$$

Centered at t_m, the entries B_m C_m, A_m, D_m, and E_m refer to the variables (y, p) at t_{m-2}, t_{m-1}, t_m, t_{m+1}, and t_{m+2}, respectively. For $m = 3, \ldots, N_t - 1$, the block A_m is given by

$$A_m = \begin{bmatrix} -(\frac{3}{2} + 4\sigma\gamma) + \delta t G' & -\chi_{ijm}\frac{\delta t}{\nu} \\ \delta t(\alpha + G'' p) & -(\frac{3}{2} + 4\sigma\gamma) + \delta t G' \end{bmatrix}, \tag{5.211}$$

where all functions within the brackets [ ] are evaluated at t_m, and the indicator function χ_{ijm} is defined, for $m = 1, \ldots, N_t + 1$, by

$$\chi_{ijm} := \begin{cases} 1 & \text{if } \underline{u}_{ijm} \le \tilde{u}_{ijm} \le \overline{u}_{ijm}, \\ 0 & \text{otherwise,} \end{cases} \tag{5.212}$$

where $\tilde{u}_{ijm}$ is given by (5.208). Thanks to the introduction of this indicator term, we guarantee a correct updating of the state and adjoint variables, mainly in the grid points, where $\underline{u}_{ijm} \le \tilde{u}_{ijm} \le \overline{u}_{ijm}$.

The B_m, C_m, D_m, and E_m blocks are given by

$$B_m = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix}, C_m = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}, D_m = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}, \text{ and } E_m = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}. \tag{5.213}$$

Clearly, for each time step, the variables neighboring the point ij are taken as constant and contribute to the right-hand side of the system.

Next, according to the proposed approach, the blocks A_2, C_2, as well as A_{N_t} and D_{N_t}, are defined in a different way because a CN method is used at $t_2 = \delta t$ and at $t_{N_t} = T - \delta t$. We then obtain that

$$A_2 = \begin{bmatrix} -(1 + 2\sigma\gamma) + \frac{\delta t}{2} G' & -\chi_{ij2}\frac{\delta t}{2\nu} \\ \delta t(\alpha + G'' p) & -(\frac{3}{2} + 4\sigma\gamma) + \delta t G' \end{bmatrix},$$
$$C_2 = \begin{bmatrix} (1 - 2\sigma\gamma) + \frac{\delta t}{2} G' & -\chi_{ij1}\frac{\delta t}{2\nu} \\ 0 & 0 \end{bmatrix}, \tag{5.214}$$

where the functions within the brackets [ ] in A_2 are evaluated at $t_2 = \delta t$, while the functions within the brackets in C_2 are evaluated at $t_1 = 0$. The matrices D_2 and E_2 are defined in the same way as in (5.213).

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Further, we have that

$$A_{N_t} = \begin{bmatrix} -(\frac{3}{2}+4\sigma\gamma)+\delta t G' & -\chi_{ijN_t}\frac{\delta t}{\nu} \\ \frac{\delta t}{2}(\alpha+G''p) & -(1+2\sigma\gamma)+\frac{\delta t}{2}G' \end{bmatrix},$$
$$D_{N_t} = \begin{bmatrix} 0 & 0 \\ \frac{\delta t}{2}(\alpha+G''p) & (1-2\sigma\gamma)+\frac{\delta t}{2}G' \end{bmatrix}, \tag{5.215}$$

where the functions within the brackets [ ] in A_{N_t} are evaluated at $t_{N_t} = T - \delta t$, while the functions within the brackets in D_{N_t} are evaluated at $t_{N_t+1} = T$. Matrices B_{N_t} and C_{N_t} are also defined in the same way as in (5.213).

It remains to discuss the blocks A_1 and A_{N_t+1} for $\beta \neq 0$. As stated before, at $t_m = t_0$, we impose the following condition

$$y_{ijm} - \psi_{ijm} = 0, \quad m = 1, \tag{5.216}$$

where ψ_h^m is a given function associated with the initial condition. Thus, the block A_1 is given by

$$A_1 = \begin{bmatrix} 1 & 0 \\ \delta t(\alpha+G''p) & -(\frac{3}{2}+4\sigma\gamma)+\delta t G' \end{bmatrix}. \tag{5.217}$$

The other blocks D_1 and E_1 are defined equally as in (5.213). Clearly, thanks to the introduction of this superfluous condition (5.216), we obtain the structure of a pentadiagonal block matrix for system matrix M.

Further, at $t_m = T$, we have the terminal condition (5.199), which we rewrite as

$$\beta(y_{ijm} - y_{T,ijm}) - p_{ijm} = 0, \quad m = N_t + 1.$$

Thus, the block A_{N_t+1} is given by

$$A_{N_t+1} = \begin{bmatrix} -(1+4\sigma\gamma)+\delta t G' & -\chi_{ijN_t+1}\frac{\delta t}{\nu} \\ \beta & -1 \end{bmatrix}, \tag{5.218}$$

and the blocks B_{N_t+1} and E_{N_t+1} are as in (5.213).

Thus, for each i, j we have to solve a pentadiagonal system $Mw = r$, where $w = (y_h^2, p_h^2, \ldots, y_h^{N_t+1}, p_h^{N_t+1})$ and $r = (r_y(w^2), r_p(w^2), \ldots, r_y(w^{N_t+1}), r_p(w^{N_t+1}))$. Summarizing, our collective t-line relaxation is given by the following algorithm.

ALGORITHM 5.14. Projected time-line CGS iteration (P-TL-CGS).

1. Set the starting approximation.
2. For ij in, e.g., lexicographic order do: calculate u_{ij} by using (5.208) and (5.209), and construct $(r_y)_{ij}$ and $(r_p)_{ij}$. Then
$$\begin{pmatrix} y \\ p \end{pmatrix}_{ij}^{(1)} = \begin{pmatrix} y \\ p \end{pmatrix}_{ij}^{(0)} + M^{-1} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ij};$$
3. end.

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Also in this case we consider that the residuals r_y and r_p are constructed at i, j and for all m prior to the update. Since the solution in time is exact, no time splitting is required.

In the case of the control-unconstrained problem, the algorithm described above does not need the projection step to calculate the control. We use the optimality condition $\nu u_h^m - p_h^m = 0$ in order to eliminate the control in the construction of $(r_y)_{ijm}$, and, consequently, the indicator term χ_{ijm} is no longer needed in the sense that we can set $\chi_{ijm} = 1$ for all grid points i, j, m and the TL-CGS smoother results.

In Algorithm 5.14, the problem of how to solve the block-pentadiagonal system $Mw = r$ arises. Efficient solvers for these kinds of systems can be constructed generalizing the Thomas algorithm, which is a well-known method to solve block-tridiagonal systems (see [353]). For this purpose, in [150] a solver is presented that solves pentadiagonal systems of order N_t with $\mathcal{O}(4(N_t + 1))$ effort; see also [25].

In the control-unconstrained case, results in [150] show that the CSMG scheme with TS-CGS and with TL-CGS smoothing, and BDF2 discretization, provides space-time second-order accurate solutions with optimal computational complexity. In the following, we report results obtained in [150] for a control-constrained optimal control problem. The multigrid algorithm with the P-TS-CGS smoothing scheme is tested with an exact solution, which is constructed as follows. We choose $\underline{u}(\mathbf{x}, t) = -1/2$ and $\overline{u}(\mathbf{x}, t) = 1/2$ and the following

$$\begin{aligned} y(x_1, x_2, t) &= (1-t)\sin(\pi x_1)\sin(\pi x_2), \\ p(x_1, x_2, t) &= \nu(1-t)\sin(2\pi x_1)\sin(2\pi x_2), \\ u(x_1, x_2, t) &= \max\{-0.5, \min\{0.5, p(x_1, x_2, t)/\nu\}\}. \end{aligned} \tag{5.219}$$

The corresponding data is given by

$$\begin{aligned} f &= -u - \partial_t y + \Delta y, \\ y_d &= y + \partial_t p + \Delta p. \end{aligned} \tag{5.220}$$

Results of experiments are reported in Table 5.20. In this case, second-order accuracy is no longer achieved due to a loss of regularity of the control function when the constraints become active. In this situation, more advanced techniques [193, 289] are required to recover high-order accuracy.

Table 5.20. *Accuracy results for a constrained-control problem:* $\sigma = 1$, $\alpha = 1$, *and* $\beta = 0$.

ν	$N_x \times N_y \times N_t$	γ	$\|y - y_h\|$	$\|p - p_h\|$	$\|u - u_h\|$
10^{-3}	$64 \times 64 \times 64$	64	$1.88 \cdot 10^{-4}$	$4.49 \cdot 10^{-6}$	$3.56 \cdot 10^{-3}$
	$128 \times 128 \times 128$	128	$5.07 \cdot 10^{-5}$	$1.10 \cdot 10^{-6}$	$8.74 \cdot 10^{-4}$
	$256 \times 256 \times 256$	256	$1.32 \cdot 10^{-5}$	$2.68 \cdot 10^{-7}$	$2.13 \cdot 10^{-4}$
10^{-7}	$64 \times 64 \times 64$	64	$3.31 \cdot 10^{-5}$	$1.81 \cdot 10^{-7}$	$2.52 \cdot 10^{-2}$
	$128 \times 128 \times 128$	128	$7.60 \cdot 10^{-6}$	$3.03 \cdot 10^{-8}$	$8.67 \cdot 10^{-3}$
	$256 \times 256 \times 256$	256	$1.85 \cdot 10^{-6}$	$5.93 \cdot 10^{-9}$	$2.94 \cdot 10^{-3}$

In Table 5.21, we report results concerning the computational performance of the CSMG approach with constrained-control problems. We obtain convergence rates that show robustness and typical multigrid efficiency that improves on finer meshes. This is due to the fact that on fine meshes the active sets are better resolved. On the other hand,

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

Table 5.21. *Numerical results for the constrained tracking problem with TS-CGS and TL-CGS smoothing schemes. Parameters:* $\sigma = 1$, $\alpha = 1$, *and* $\beta = 0$. *Initial condition for state equation:* $y_0 = y(\mathbf{x}, 0)$.

	ν	γ	ρ	$\|r_y\|$	$\|r_p\|$	$\|y_h - R_h\, y_d\|$
TS-CGS	10^{-2}	32	0.034	$1.32 \cdot 10^{-9}$	$6.56 \cdot 10^{-11}$	$2.26 \cdot 10^{-1}$
		64	0.030	$6.32 \cdot 10^{-9}$	$3.41 \cdot 10^{-10}$	$2.29 \cdot 10^{-1}$
		128	0.029	$6.17 \cdot 10^{-8}$	$4.09 \cdot 10^{-9}$	$2.30 \cdot 10^{-1}$
	10^{-4}	32	0.081	$3.67 \cdot 10^{-9}$	$5.86 \cdot 10^{-12}$	$2.30 \cdot 10^{-3}$
		64	0.029	$4.43 \cdot 10^{-10}$	$5.03 \cdot 10^{-12}$	$2.29 \cdot 10^{-3}$
		128	0.028	$2.87 \cdot 10^{-9}$	$6.21 \cdot 10^{-11}$	$2.30 \cdot 10^{-3}$
	10^{-6}	32	0.548	$7.15 \cdot 10^{-6}$	$6.27 \cdot 10^{-9}$	$1.75 \cdot 10^{-4}$
		64	0.293	$1.95 \cdot 10^{-8}$	$1.94 \cdot 10^{-11}$	$4.35 \cdot 10^{-5}$
		128	0.091	$1.76 \cdot 10^{-9}$	$4.99 \cdot 10^{-13}$	$2.52 \cdot 10^{-5}$
TL-CGS	10^{-2}	32	0.034	$1.34 \cdot 10^{-9}$	$6.57 \cdot 10^{-11}$	$2.26 \cdot 10^{-1}$
		64	0.030	$6.30 \cdot 10^{-9}$	$3.41 \cdot 10^{-10}$	$2.29 \cdot 10^{-1}$
		128	0.029	$6.17 \cdot 10^{-8}$	$4.09 \cdot 10^{-9}$	$2.30 \cdot 10^{-1}$
	10^{-4}	32	0.082	$3.67 \cdot 10^{-9}$	$5.66 \cdot 10^{-12}$	$2.29 \cdot 10^{-3}$
		64	0.029	$6.82 \cdot 10^{-10}$	$5.05 \cdot 10^{-12}$	$2.29 \cdot 10^{-3}$
		128	0.028	$3.42 \cdot 10^{-9}$	$6.21 \cdot 10^{-11}$	$2.30 \cdot 10^{-3}$
	10^{-6}	32	0.503	$4.18 \cdot 10^{-3}$	$2.20 \cdot 10^{-5}$	$1.30 \cdot 10^{-4}$
		64	0.294	$1.56 \cdot 10^{-8}$	$1.32 \cdot 10^{-11}$	$3.86 \cdot 10^{-5}$
		128	0.149	$2.18 \cdot 10^{-9}$	$1.74 \cdot 10^{-12}$	$2.48 \cdot 10^{-5}$

smaller values of ν result in larger active sets and steeper gradient of the control function arise, making the problem more difficult to solve and thus explaining the worsening of the convergence factor.

In Figure 5.18, the state y and the control u for two instants are depicted. We observe that for small time periods, the control is active, but as the time period increases, the control becomes nonactive, since $p \to 0$ as $t \to T = 1$.

We complete this section reporting results of experiments for the limit case of $\nu = 0$. This discussion is possible due to the robustness of the CSMG approach where the smoothing scheme remains well defined choosing a zero weight of the control. This appears to be a unique feature of this solution strategy. In particular, with this scheme it is possible to investigate bang-bang control problems that arise, e.g., taking $\nu = 0$ and nonattainable target functions. This is a less investigated subject due to the difficulty of computing bang-bang solutions; see [48, 59, 147, 370].

We take $f = 0$ and $y_0(x) = 0$, $\underline{u}(\mathbf{x}, t) = -1$ and $\overline{u}(\mathbf{x}, t) = 1$, and consider the following target trajectory

$$y_d(x_1, x_2, t) = \sin(2\pi t)\, \sin(3\pi x_1) \sin(3\pi x_2).$$

With this target function and $\nu = 0$ we obtain a control which is everywhere active; that is, we have bang-bang control. In Figure 5.19, the optimal control and the corresponding state for $\nu = 0$ are depicted for two different instants of time.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

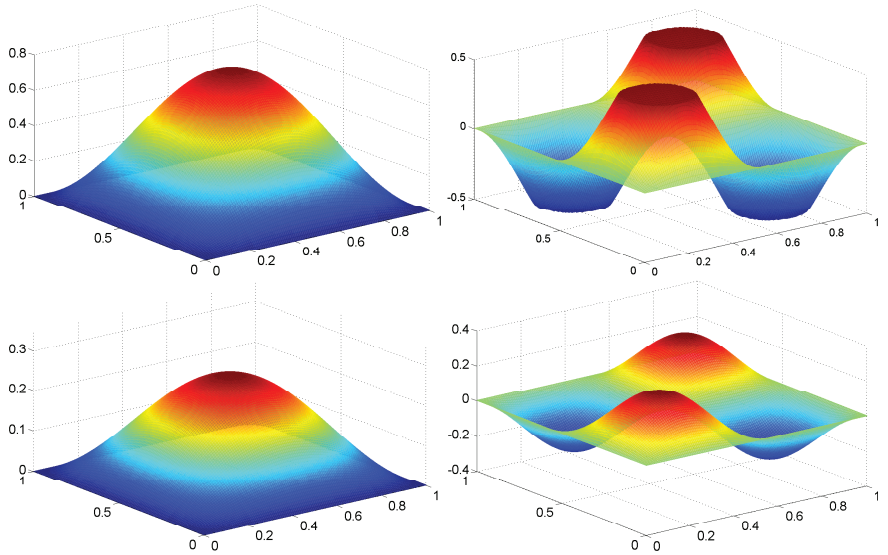

Figure 5.18. *Control constrained tracking problem: state y (left column) and u (right column) for $t = T/4 = 0.25$ and $t = 3T/4 = 0.75$. Parameters: $\alpha = 1$, $\beta = 0$, $\nu = 10^{-7}$, $\gamma = 64$, and $\sigma = 1$. Reprinted with permission from S. Gonzalez Andrade and A. Borzì, Multigrid second-order accurate solution of parabolic control-constrained problems, Computational Optimization and Applications, to appear.*

Local Fourier Analysis for BDF2 Discretization

We can use the local Fourier analysis framework to analyze the TS-CGS smoothing scheme applied to (5.201a)–(5.201b) resulting from the BDF2 discretization and without constraints on the control. To analyze the update procedure at (i, m), we consider that y_{im-1} and y_{im-2} as well as p_{im+1} and p_{im+2} have been updated in the previous iteration step and consequently they have the superindex (1). We obtain that

$$\begin{pmatrix} -(\frac{3}{2}+2\sigma\gamma)+\sigma\gamma e^{-i\theta_x} & -\frac{\delta t}{\nu} \\ \alpha\delta t & -(\frac{3}{2}+2\sigma\gamma)+\sigma\gamma e^{-i\theta_x} \end{pmatrix} \begin{pmatrix} \tilde{Y}^{(1)}_{\boldsymbol{\theta}} \\ \tilde{P}^{(1)}_{\boldsymbol{\theta}} \end{pmatrix}$$
$$+ \begin{pmatrix} 2e^{-i\theta_t} - \frac{e^{-2i\theta_t}}{2} & 0 \\ 0 & 2e^{i\theta_t} - \frac{e^{2i\theta_t}}{2} \end{pmatrix} \begin{pmatrix} \tilde{Y}^{(1)}_{\boldsymbol{\theta}} \\ \tilde{P}^{(1)}_{\boldsymbol{\theta}} \end{pmatrix} = \begin{pmatrix} -\sigma\gamma e^{i\theta_x} & 0 \\ 0 & -\sigma\gamma e^{i\theta_x} \end{pmatrix} \begin{pmatrix} \tilde{Y}^{(0)}_{\boldsymbol{\theta}} \\ \tilde{P}^{(0)}_{\boldsymbol{\theta}} \end{pmatrix}.$$

Hence, we obtain the Fourier symbol of the TS-CGS scheme as follows

$$\hat{s}(\boldsymbol{\theta})$$
$$= \begin{pmatrix} -(\frac{3}{2}+2\sigma\gamma)+\sigma\gamma e^{-i\theta_x}+2e^{-i\theta_t}-\frac{e^{-2i\theta_t}}{2} & -\frac{\delta t}{\nu} \\ \alpha\delta t & -(\frac{3}{2}+2\sigma\gamma)+\sigma\gamma e^{-i\theta_x}+2e^{i\theta_t}-\frac{e^{2i\theta_t}}{2} \end{pmatrix}^{-1}$$
$$\times \begin{pmatrix} -\sigma\gamma e^{i\theta_x} & 0 \\ 0 & -\sigma\gamma e^{i\theta_x} \end{pmatrix}.$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

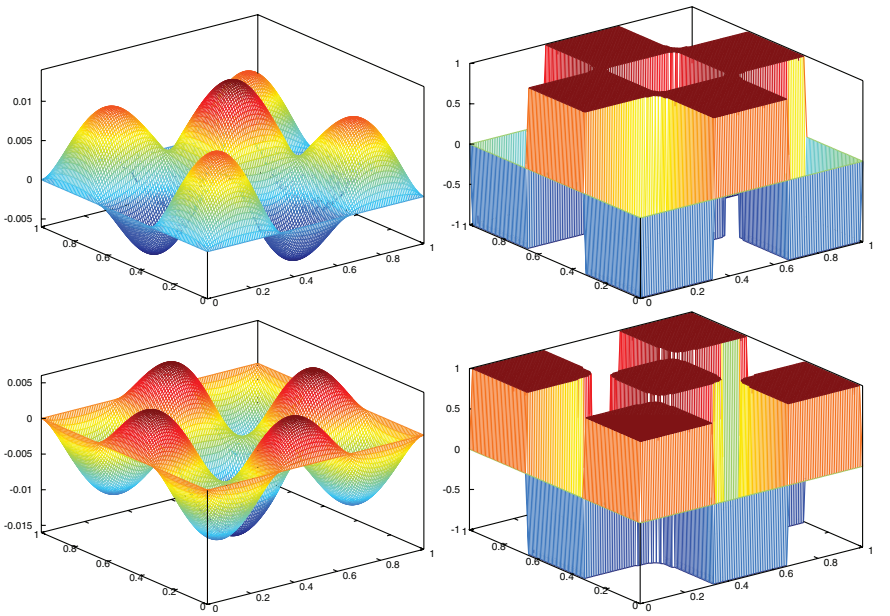

Figure 5.19. *Numerical bang-bang control solutions with $\nu = 0$ at $t = T/4$ (top) and $t = 3T/4$ (bottom). The state (left) and the control (right); $128 \times 128 \times 128$ mesh. Reprinted with permission from S. Gonzalez Andrade and A. Borzì, Multigrid second-order accurate solution of parabolic control-constrained problems, Computational Optimization and Applications, to appear.*

The numerical evaluation of the smoothing factor $\mu(S_k)$ of the TS-CGS scheme demonstrates that the TS-CGS scheme for the BDF2 discretization is mesh independent and optimization-parameter independent for a very large range of meshes and of values of the weight of the cost of the control.

Next, consider the case of TL-CGS relaxation. In this case, the Fourier symbol of the smoothing operator is given by the following 2×2 matrix

$$\hat{s}(\boldsymbol{\theta}) = -(A + B\,e^{-2i\theta_t} + C\,e^{-i\theta_t} + D\,e^{i\theta_t} + E\,e^{2i\theta_t} + \tilde{I}\,e^{-i\theta_x})^{-1}(\tilde{I}\,e^{i\theta_x}),$$

where

$$A = \begin{bmatrix} -(\frac{3}{2} + 2\sigma\gamma) & -\frac{\delta t}{\nu} \\ \alpha\delta t & -(\frac{3}{2} + 2\sigma\gamma) \end{bmatrix}, B_m = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix},$$

$$C_m = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}, D_m = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}, E_m = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{2} \end{bmatrix}, \text{ and } \tilde{I} = \begin{bmatrix} \sigma\gamma & 0 \\ 0 & \sigma\gamma \end{bmatrix}.$$

Since the system is solved for all m at once, we have that y_{im-1}, y_{im-2}, p_{im+1}, and p_{im+2} have superindex (0). We obtain that the TL-CGS scheme has smoothing properties similar to those of TS-CGS scheme. This fact appears also from results of numerical experiments.

Next, we investigate the action of the coarse-grid correction to obtain TG convergence estimates. We consider the full-weighting restriction operator (5.195) and the symbol

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

of the fine-grid operator is as in (5.196), where

$$\begin{aligned} a_y(\boldsymbol{\theta}) &= 2\sigma\gamma\cos(\theta_x) + 2e^{-i\theta_t} - \tfrac{1}{2}e^{-2i\theta_t} - 2\sigma\gamma - \tfrac{3}{2}, \\ a_p(\boldsymbol{\theta}) &= 2\sigma\gamma\cos(\theta_x) + 2e^{i\theta_t} - \tfrac{1}{2}e^{2i\theta_t} - 2\sigma\gamma - \tfrac{3}{2}. \end{aligned}$$

The symbol of the coarse-grid operator follows

$$\widehat{A}_{k-1}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\sigma\gamma\cos(\theta_x)}{2} + 2e^{-i\theta_t} - \frac{1}{2}e^{-2i\theta_t} - \frac{\sigma\gamma+3}{2} & -\delta t/\nu \\ \alpha\delta t & \frac{\sigma\gamma\cos(\theta_x)}{2} + 2e^{i\theta_t} - \frac{1}{2}e^{2i\theta_t} - \frac{\sigma\gamma+3}{2} \end{bmatrix}.$$

Notice that on the coarser grid δt remains unchanged, since we do not apply coarsening in the time direction, while $\gamma \to \gamma/4$ by coarsening.

Based on the representation on TG_k^{k-1} by a 4×4 matrix $\widehat{TG}_k^{k-1}(\boldsymbol{\theta})$ we can calculate the convergence factor given by $\rho(TG_k^{k-1})$ that requires us to determine the spectral radius of a 4×4 matrix.

We obtain that the convergence factor ρ is almost independent of the value of the weight ν and of the discretization parameter γ for both choices of the smoothing scheme. Notice that our result predicts convergence factors that improve for smaller values of the optimization parameter. This is a unique feature of the CSMG multigrid approach. The estimates obtained with TG local Fourier analysis are sharp, and in order to facilitate comparison with values of convergence factors obtained with numerical experiments, we report in Tables 5.22 and 5.23 the local Fourier analysis quantitative estimates of the smoothing factor and the convergence factor of TL-CGS- and TS-CGS-multigrid schemes.

Table 5.22. *Smoothing factor $\mu(S_k)$ and convergence factor $\rho(TG_k^{k-1})$ for the TS-CGS multigrid scheme ($\nu_1 = \nu_2 = 1$). Parameters: $\delta t = 1/64$, $\sigma = 1$, $\alpha = 1$, and $\beta = 0$.*

TS-CGS	γ \ ν	10^{-8}	10^{-6}	10^{-4}	10^{-2}
$\mu(S_k)$	32	0.2289	0.4843	0.4516	0.4493
	48	0.3317	0.4737	0.4502	0.4486
	64	0.4056	0.4677	0.4494	0.4483
$\rho(TG_k^{k-1})$	32	0.0427	0.1317	0.1361	0.1347
	48	0.0822	0.1352	0.1354	0.1344
	64	0.1147	0.1368	0.1350	0.1342

5.7.5 Projected Collective Smoothing Schemes and the Semismooth Newton Method

The CSMG collective smoothing schemes are formulated based on the idea of obtaining at the grid-point (grid-block) level the mappings $y = y(u)$ and $p = p(u)$ and using these mappings in the optimality condition to compute the unconstrained control. In the presence of constraints the projection of this control on the given bounds gives the required solution. We show that the control obtained using this procedure can be equally obtained applying a local or block semismooth Newton (SSN) methods [191, 281, 343].

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.23. *Smoothing factor $\mu(S_k)$ and convergence factor $\rho(TG_k^{k-1})$ for the TL-CGS multigrid schemes ($\nu_1 = \nu 2 = 1$). Parameters: $\delta t = 1/64$, $\sigma = 1$, $\alpha = 1$, and $\beta = 0$.*

TL-CGS	γ \ ν	10^{-8}	10^{-6}	10^{-4}	10^{-2}
$\mu(S_k)$	32	0.2289	0.4843	0.4516	0.4493
	48	0.3317	0.4737	0.4502	0.4486
	64	0.4056	0.4677	0.4494	0.4483
$\rho(TG_k^{k-1})$	32	0.0427	0.1300	0.1282	0.1266
	48	0.0822	0.1330	0.1303	0.1290
	64	0.1147	0.1345	0.1313	0.1301

Now, we analyze the application of a local SSN method applied to (5.204) to show that the resulting iterative scheme is equivalent to the P-TS-CGS scheme. Recall that (5.204c) is equivalent to the following (see [245])

$$u(\mathbf{x},t) = \max\left\{\underline{u}(\mathbf{x},t), \min\left\{\overline{u}(\mathbf{x},t), \frac{1}{\nu}p(\mathbf{x},t)\right\}\right\} \text{ a.e. in } Q \text{ and for } \nu > 0. \tag{5.221}$$

We consider (5.221) at a grid point $(i\ j\ m)$ and, for the ease of illustration, we study the system (5.204) with $G(y) = 0$. Further, we denote (5.204) as the following operator equation

$$\Phi(y_{ijm}, p_{ijm}, u_{ijm}) := \begin{bmatrix} -a\, y_{ijm} + S_{ijm} - \delta t u_{ijm} \\ -a\, p_{ijm} + R_{ijm} + \alpha\delta t y_{ijm} \\ u_{ijm} - \max\left\{\underline{u}_{ijm}, \min\left\{\overline{u}_{ijm}, \frac{1}{\nu}p_{ijm}\right\}\right\} \end{bmatrix} = 0. \tag{5.222}$$

We can state that both the max and the min functions involved in (5.222) are semismooth. Indeed, it is well known (see Lemma 3.1 in [191]) that the mappings $y \to \max(0, y)$ and $y \to \min(0, y)$, from $\mathbb{R}^n$ to $\mathbb{R}^n$, $n \in \mathbb{N}$, are Newton differentiable with Newton derivatives given by the diagonal matrices

$$(\Gamma_{max})_{ii} := \begin{cases} 1 & \text{if } y_i \geq 0, \\ 0 & \text{if } y_i < 0, \end{cases} \quad \text{and} \quad (\Gamma_{min})_{ii} := \begin{cases} 1 & \text{if } y_i \leq 0, \\ 0 & \text{if } y_i > 0, \end{cases} \quad i = 1, \ldots, n, \tag{5.223}$$

respectively. As a result, from Theorem 4.6 in [322] it is implied that the real function $\max\{\underline{u}_{ijm}, \min\{\overline{u}_{ijm}, \frac{1}{\nu}p_{ijm}\}\}$ is Newton differentiable, with respect to p_{ijm}, and its Newton derivative is given by

$$\Gamma_p := \chi_{\mathcal{A}_+}\, \chi_{\mathcal{A}_-}\, \frac{1}{\nu},$$

where $\chi_{\mathcal{A}_+}$ and $\chi_{\mathcal{A}_-}$ are defined by

$$\chi_{\mathcal{A}_+} := \begin{cases} 1 & \text{if } \min\{\overline{u}_{ijm}, \frac{1}{\nu}p_{ijm}\} \geq \underline{u}_{ijm}, \\ 0 & \text{if } \min\{\overline{u}_{ijm}, \frac{1}{\nu}p_{ijm},\} < \underline{u}_{ijm}, \end{cases} \quad \text{and} \quad \chi_{\mathcal{A}_-} := \begin{cases} 1 & \text{if } \frac{1}{\nu}p_{ijm} \leq \overline{u}_{ijm}, \\ 0 & \text{if } \frac{1}{\nu}p_{ijm} > \overline{u}_{ijm}, \end{cases} \tag{5.224}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

respectively. Consequently, we obtain the SSN step applied to the operator equation (5.222) as follows

$$\begin{bmatrix} -a & 0 & -\delta t \\ \delta t\alpha & -a & 0 \\ 0 & -\chi_{\mathcal{A}_+}\chi_{\mathcal{A}_-}\frac{1}{\nu} & 1 \end{bmatrix}_{ijm} \begin{pmatrix} \delta_y \\ \delta_p \\ \delta_u \end{pmatrix}_{ijm} = \begin{pmatrix} r_y \\ r_p \\ r_u \end{pmatrix}_{ijm}. \tag{5.225}$$

Since (5.225) is an uncoupled system of equations for the residual $(\delta_u)_{ijm}$, we have

$$u^{(1)}_{ijm} = \max\left\{\underline{u}_{ijm}, \min\left\{\overline{u}_{ijm}, \frac{1}{\nu}p^{(0)}_{ijm}\right\}\right\} + \chi_{\mathcal{A}_+}\chi_{\mathcal{A}_-}\frac{1}{\nu}(\delta_p)_{ijm}, \tag{5.226}$$

which yields that

$$\begin{aligned} \begin{pmatrix} y \\ p \end{pmatrix}^{(1)}_{ijm} &= \begin{pmatrix} y \\ p \end{pmatrix}^{(0)}_{ijm} \\ &+ \begin{bmatrix} -a & -\delta t\,\chi_{\mathcal{A}_+}\chi_{\mathcal{A}_-}\frac{1}{\nu} \\ \alpha\,\delta t & -a \end{bmatrix}^{(0)-1}_{ijm} \begin{pmatrix} a\,y - S + \delta t U \\ a\,p - R - \alpha\delta t\,y \end{pmatrix}^{(0)}_{ijm}, \end{aligned} \tag{5.227}$$

where $U^{(0)}_{ijm} := \max\{\underline{u}_{ijm}, \min\{\overline{u}_{ijm}, \frac{1}{\nu}p^{(0)}_{ijm}\}\}$.

Now, we show that one iteration step given by (5.226)–(5.227) is equivalent to one iteration of the algorithm P-TS-CGS in the sense that the two methods compute the same update for the control, the state, and the adjoint variables. Indeed, the local SSN iteration must be performed in the forward time direction to calculate the updates for y_{ijm} and u_{ijm} and in the backward time direction to calculate the updates for p_{ijm}.

Consider the three possible occurrences given by (5.209):

(i) $\frac{1}{\nu}p_{ijm} > \overline{u}_{ijm}$. Here, we have that $\chi_{\mathcal{A}_-} = 0$, and we obtain that $U_{ijm} := \overline{u}_{ijm}$. Therefore, from (5.226), we obtain that $u^{(1)}_{ijm} = \overline{u}_{ijm}$ and, from (5.227), the following updates for y_{ijm} and p_{ijm}:

$$\begin{aligned} y^{(1)}_{ijm} &= y^{(0)}_{ijm} + \frac{(r_y)_{ijm}}{-a}, \\ p^{(1)}_{ijm} &= p^{(0)}_{ijm} + \frac{-\alpha\,\delta t\,(r_y)_{ijm} - a\,(r_p)_{ijm}}{a^2} \end{aligned} \tag{5.228}$$

with $(r_y)_{ijm} := a\,y^{(0)}_{ijm} - S_{ijm} + \delta t\overline{u}_{ijm}$ and $(r_p)_{ijm} := a\,p^{(0)}_{ijm} - R_{ijm} - \alpha\delta t\,y_{ijm}$.

(ii) $\frac{1}{\nu}p_{ijm} < \underline{u}_{ijm}$. In this case, we have that $\chi_{\mathcal{A}_+} = 0$, since $\min\{\overline{u}_{ijm}, \frac{1}{\nu}p_{ijm}\} < \underline{u}_{ijm}$. Hence, we have that $U_{ijm} = \underline{u}_{ijm}$ and (5.226) implies that $u^{(1)}_{ijm} = \underline{u}_{ijm}$. Further, (5.227) gives the following updates for y_{ijm} and p_{ijm}:

$$\begin{aligned} y^{(1)}_{ijm} &= y^{(0)}_{ijm} + \frac{(r_y)_{ijm}}{-a}, \\ p^{(1)}_{ijm} &= p^{(0)}_{ijm} + \frac{-\alpha\,\delta t\,(r_y)_{ijm} - a\,(r_p)_{ijm}}{a^2} \end{aligned} \tag{5.229}$$

with $(r_y)_{ijm} := a\,y^{(0)}_{ijm} - S_{ijm} + \delta t\underline{u}_{ijm}$ and $(r_p)_{ijm} := a\,p^{(0)}_{ijm} - R_{ijm} - \alpha\delta t\,y_{ijm}$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

(iii) $\underline{u}_{ijm} \leq \frac{1}{\nu} p_{ijm} \leq \overline{u}_{ijm}$. In this case, we have that $\chi_{\mathcal{A}_+} = \chi_{\mathcal{A}_-} = 1$. Thus, $U_{ijm} = \frac{1}{\nu} p^{(0)}_{ijm}$ and (5.226) yields that

$$\begin{aligned} u^{(1)}_{ijm} &= \tfrac{1}{\nu} p^{(0)}_{ijm} + \tfrac{1}{\nu}(\delta_p)_{ijm} \\ &= \tfrac{1}{\nu} p^{(0)}_{ijm} + \tfrac{1}{\nu}(p^{(1)}_{ijm} - p^{(0)}_{ijm}) = \tfrac{1}{\nu} p^{(1)}_{ijm}. \end{aligned} \tag{5.230}$$

Moreover, by solving the system (5.227) we obtain the following updates for y_{ijm} and p_{ijm}

$$y^{(1)}_{ijm} = \frac{\nu a\, S_{ijm} - \nu\, \delta t\, R_{ijm}}{\nu a^2 + \alpha\, \delta t^2} \quad \text{and} \quad p^{(1)}_{ijm} = \frac{\nu a\, R_{ijm} - \nu \alpha \delta t\, S_{ijm}}{\nu\, a^2 + \alpha \delta t^2}. \tag{5.231}$$

The equivalence between the SSN iteration and the P-TS-CGS iteration is clear in the cases (i) and (ii). Furthermore, in the case (iii), since $\widetilde{u}_{ijm}$ is given by

$$\widetilde{u}_{ijm} = \frac{a\, R_{ijm} + \delta t \alpha\, S_{ijm}}{a^2\, \delta t + \alpha\, \delta t^2},$$

we obtain the same expressions (5.231), by plugging $\widetilde{u}_{ijm}$ in $(r_y)_{ijm}$. Therefore, the equivalence between the P-TS-CGS and the SSN iteration (5.226)–(5.227) is totally established.

We complete this section analyzing the equivalence between the P-TL-CGS approach and an SSN scheme. We require that the controls u_{ijm} be available to construct the residuals $(r_y)_{ijm}$ and $(r_p)_{ijm}$, prior to the calculation of the generalized Newton step, when solving the residual equations. We consider that the controls u_{ijm} are given, for $i = 1, \ldots, N_x$, $j = 1, \ldots, N_y$, and $m = 1, \ldots, N_t + 1$, by

$$u_{ijm} := \max\left\{\underline{u}_{ijm}, \min\left\{\overline{u}_{ijm}, \frac{1}{\nu} p_{ijm}\right\}\right\}. \tag{5.232}$$

Further, for ease of illustration, we consider the linear case $G(y) = 0$. Therefore, due to the fact that function (5.232) is Newton differentiable, we calculate an SSN step to solve the residual equations as follows

$$\begin{pmatrix} \delta_y \\ \delta_p \end{pmatrix}_{ij} = -J_E^{-1} \begin{pmatrix} r_y \\ r_p \end{pmatrix}_{ij} \quad \text{for } i = 1, \ldots, N_x, \text{ and } j = 1, \ldots, N_y, \tag{5.233}$$

where J_E stands for the generalized Jacobian of E_{ij}. We immediately observe that this Jacobian can be written as the following block pentadiagonal matrix

$$-J_E = \begin{bmatrix} A_1 & D_1 & E_1 & & & & & \\ \widetilde{C}_2 & \widetilde{A}_2 & D_2 & E_2 & & & & \\ B_3 & C_3 & \widetilde{A}_3 & D_3 & E_3 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & B_{N_t-1} & C_{N_t-1} & \widetilde{A}_{N_t-1} & D_{N_t-1} & E_{N_t-1} \\ & & & & B_{N_t} & C_{N_t} & \widetilde{A}_{N_t} & D_{N_t} \\ & & & & & B_{N_t+1} & C_{N_t+1} & \widetilde{A}_{N_t+1} \end{bmatrix}, \tag{5.234}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva

where the diagonal block entries are given by

$$\widetilde{A}_m = \begin{bmatrix} -(\frac{3}{2}+4\sigma\gamma) & -\delta t\, \chi_{\mathcal{A}_+}\, \chi_{\mathcal{A}_-}\, \frac{1}{\nu} \\ \delta t\, \alpha & -(\frac{3}{2}+4\sigma\gamma) \end{bmatrix} \quad \text{for } m = 3, \ldots, N_t - 1.$$

Next, due to the combination of CN and BDF2 schemes for $m = 2$ and $m = N_t$, we have that

$$\widetilde{A}_2 = \begin{bmatrix} -(1+2\sigma\gamma) & -\delta t\, \chi_{\mathcal{A}_+}\, \chi_{\mathcal{A}_-}\, \frac{1}{2\nu} \\ \delta t\, \alpha & -(\frac{3}{2}+4\sigma\gamma) \end{bmatrix} \text{ and } \widetilde{A}_{N_t} = \begin{bmatrix} -(\frac{3}{2}+4\sigma\gamma) & -\delta t\, \chi_{\mathcal{A}_+}\, \chi_{\mathcal{A}_-}\, \frac{1}{\nu} \\ \frac{\delta t}{2}\, \alpha & -(1+2\sigma\gamma) \end{bmatrix}.$$

The terminal block is given by

$$\widetilde{A}_{N_t+1} = \begin{bmatrix} -(\frac{3}{2}+4\sigma\gamma) & -\delta t\, \chi_{\mathcal{A}_+}\, \chi_{\mathcal{A}_-}\, \frac{1}{\nu} \\ \beta & 1 \end{bmatrix},$$

and the matrix $\widetilde{C}_2$ takes the following form

$$\widetilde{C}_2 = \begin{bmatrix} -(1-2\sigma\gamma) & -\delta t\, \chi_{\mathcal{A}_+}\, \chi_{\mathcal{A}_-}\, \frac{1}{2\nu} \\ 0 & 0 \end{bmatrix}.$$

As before, the functions within the brackets [ ] are evaluated at the corresponding t_m. Further, all the other constituent block matrices of $-J_E$ are defined in the same way as the corresponding constituent blocks of the matrix M. Furthermore, notice that the function $\chi_{\mathcal{A}_+}\, \chi_{\mathcal{A}_-}$ can be rewritten as

$$\chi_{ijm} := \begin{cases} 1 & \text{if } \underline{u}_{ijm} \leq \frac{1}{\nu}\, p_{ijm} \leq \overline{u}_{ijm}, \\ 0 & \text{otherwise,} \end{cases}$$

which is, by construction, equivalent to the indicator function (5.212). Therefore, the matrix M and the generalized Jacobian $-J_E$ are equivalent. Thanks to this argumentation, we conclude that the algorithm P-TL-CGS is equivalent to the SSN strategy given by the generalized Newton step (5.233).

5.7.6 Multigrid Receding-Horizon Approach

It is possible to combine multigrid schemes with receding-horizon techniques [208] to develop an efficient optimal control algorithm for tracking a desired trajectory over very long time intervals. In the following, we sketch the implementation of the multigrid receding-horizon scheme. For an application of this scheme in physiology see [46].

Consider the optimal control problem of tracking y_d for $t \geq 0$. Define time windows of size Δt. In each time window, an optimal control problem with tracking ($\alpha = 1$) and terminal observation ($\beta = 1$) is solved with desired trajectory given by $y_d(x,t)$, $t \in$

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

$(n\Delta t,(n+1)\Delta t)$, and terminal state given by $y_T(x) = y_d(x,(n+1)\Delta t)$. The resulting optimal state at $n\Delta t$ defines the initial condition for the next optimal control problem defined in $(n\Delta t,(n+1)\Delta t)$. The following algorithm results.

ALGORITHM 5.15. Multigrid receding-horizon scheme (MG-RH).

1. Set $y(\mathbf{x},0) = y_0(\mathbf{x})$ and $n = 0$.
2. Set $y_T(\mathbf{x}) = y_d(\mathbf{x},(n+1)\Delta t)$.
3. Apply the CSMG scheme to solve the optimal control problem in $(n\Delta t,(n+1)\Delta t)$.
4. Update $n := n+1$, set $y_0(\mathbf{x}) = y(\mathbf{x},n\Delta t)$, and goto 2.

To show the ability of the CSMG multigrid receding-horizon approach to track over long time intervals, we consider the following desired trajectory

$$y_d(x_1,x_2,t) := t\,\sin(2\pi t)(x_1 - x_1^2)(x_2 - x_2^2)$$

over the time interval $(0,T)$ with $T = 5$, using 10 time windows of size $\Delta t = 0.5$. We test the receding-horizon algorithm considering a control-unconstrained and a control-constrained problem. In both cases, we take $\sigma = 0.01$ and we use the CSMG algorithm with the P-TL-CGS smoothing schemes because this smoothing scheme is efficient and robust with small values of the diffusion coefficient.

For the control-unconstrained receding-horizon case, we take $\alpha = 1$, $\beta = 1$, and $\nu = 10^{-6}$. We obtain that in each time window the optimal control problem is solved to the required tolerance by 3 STMG-$V(2,2)$-cycles on a grid with $\gamma = 64$. In Figure 5.20(left) the time evolution of the state variable compared to the desired trajectory is depicted and shows accurate tracking ($\|y_h - R_h\, y_d\| \approx 10^{-6}$). Also in Figure 5.20(right), the control function is depicted that appears to be smooth across time windows.

For the control-constrained receding-horizon case, we study the tracking of the given trajectory, considering the following constraints for the control $\underline{u}(\mathbf{x},t) = -1$ and $\overline{u}(\mathbf{x},t) = 1$. Our experience shows that fast accurate tracking is obtained taking $\alpha = 1$, $\beta = 0.1$, and $\nu = 10^{-4}$. In this case, the optimal control problem is solved to the required tolerance,

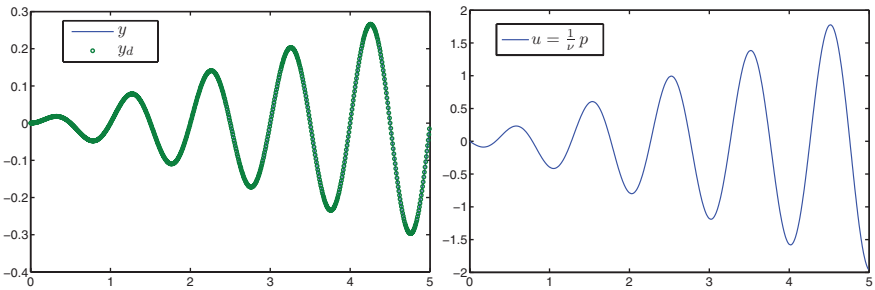

Figure 5.20. *Receding-horizon solution for the control-unconstrained tracking problem. Left: time evolution of the state y (solid line) and the desired trajectory y_d (dots) at $(x_1,x_2) = (0.5,0.5)$. Right: optimal control $u = \frac{1}{\nu}\,p$ at $(x_1,x_2) = (0.5,0.5)$.*

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

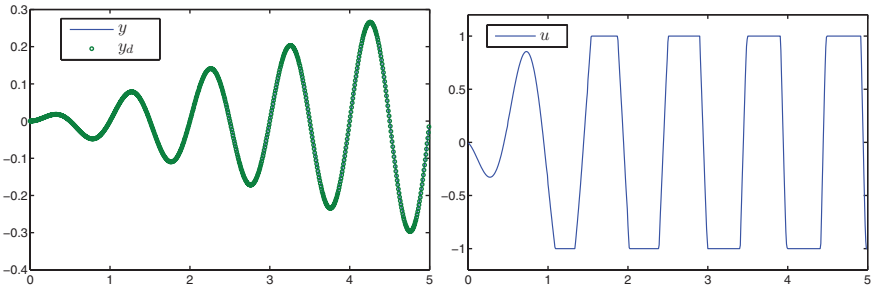

Figure 5.21. *Receding-horizon solution for the control-constrained tracking problem. Left: time evolution of the state y (solid line) and the desired trajectory y_d (dots) at $(x_1,x_2) = (0.5,0.5)$. Right: optimal control u at $(x_1,x_2) = (0.5,0.5)$. Reprinted with permission from S. Gonzalez Andrade and A. Borzì, Multigrid second-order accurate solution of parabolic control-constrained problems, Computational Optimization and Applications, to appear.*

on a grid with $\gamma = 64$, by 5 STMG-$W(2,2)$-cycles, in average in each time window. In Figure 5.21(left), the time evolution of the state variable compared to the desired trajectory is depicted that shows accurate tracking ($\|y_h - R_h\, y_d\| \approx 10^{-4}$). In Figure 5.21(right), the control function is depicted. We observe that initially the control constraints appear to be nonactive but as the time increases, the constraints become active.

For a challenging application of the MG-RH scheme to control the reaction-diffusion model of cardiac arrhythmia [3, 272], see [46].

5.7.7 A CSMG Scheme for Fredholm Control Problems

We discuss the CSMG solution of an optimal control problem governed by a Fredholm integral equation of the second kind. Specifically, we consider the Fredholm integral equations of the second kind with linear distributed control mechanism introduced in Chapter 3. We have

$$\min J(y,u) := \frac{1}{2}\|y - z\|^2_{L^2(\Omega)} + \frac{\nu}{2}\|u\|^2_{L^2(\Omega)}, \tag{5.235}$$

$$y = f(y) + u + g \qquad \text{in } \Omega, \tag{5.236}$$

where $g, z \in L^2(\Omega)$ are given and $\nu > 0$. The term $f(y)$ is supposed to be symmetric and is given by

$$f(y)(x) = \int_\Omega K(x,t)\, y(t)\, dt. \tag{5.237}$$

Existence and uniqueness of solution for (5.235)–(5.237) and related discretization issues are discussed in Section 3.4.

Without constraints on the control u, we have the scalar equation $\nu u - p = 0$, which we use to eliminate $u = p/\nu$ and obtain the following equivalent system

$$\left\{ \begin{array}{lcl} y - f(y) - p/\nu &=& g, \\ p - f(p) + y &=& z. \end{array} \right. \tag{5.238}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

This system corresponds to two coupled integral equations. As discussed in Chapter 3, we use the Nyström method [14, 176] to discretize this problem. We obtain the following discrete optimality system

$$\begin{aligned} y_i - h\sum_{j=-N}^{N} w_{ij}\, y_j - p_i/\nu &= g_i, \\ p_i - h\sum_{j=-N}^{N} w_{ij}\, p_j + y_i &= z_i. \end{aligned} \tag{5.239}$$

Now, we define an iterative procedure on Ω_h that belongs to the class of CGS schemes. It results from a sequential update of the optimization variables at each grid point. The update of the variables (y_i, p_i) at grid point x_i is obtained by solving exactly the discrete optimality system with respect to (y_i, p_i) and considering the remaining variables as constant. This iterative method is given by the following algorithm. Let an initial approximation $(y^{(0)}, p^{(0)})$ be given. Here, tol is the required tolerance on the L^2-norm of the residual of the constraint equation.

ALGORITHM 5.16. CGS iteration scheme.

1. For $m = 0, 1, 2, \ldots, \ell$ do
2. If $\|g_h - y_h^m + f_h(y_h^m) - p_h^m/\nu\|_{L_h^2} < tol$ then stop.
3. For $i = -N, -N+1, \ldots, 0, 1, \ldots, N-1, N$ (lexicographic order) do

$$\begin{pmatrix} y_i \\ p_i \end{pmatrix}^{(m+1)} = \begin{pmatrix} 1 - hw_{ii} & -1/\nu \\ 1 & 1 - hw_{ii} \end{pmatrix}^{-1} \times \left[\begin{pmatrix} g_i \\ z_i \end{pmatrix} + h\sum_{j<i} w_{ij} \begin{pmatrix} y_j \\ p_j \end{pmatrix}^{(m+1)} + h\sum_{j>i} w_{ij} \begin{pmatrix} y_j \\ p_j \end{pmatrix}^{(m)} \right]. \tag{5.240}$$

4. End.

Notice that the summation in (5.240) can be implemented in a fast procedure with recursive subtraction and addition of the quantity being updated. In [6], this smoothing procedure is implemented in a CSMG scheme obtaining a fast and robust multigrid solver for Fredholm integral control problems. This result is in agreement with estimates of local Fourier analysis presented in [6].

Next, we report results of numerical experiments to validate the convergence performance of the CGS and CSMG schemes. We consider an application corresponding to a kernel that represents the covariance function of an Ornstein–Uhlenbeck stochastic process at the equilibrium that arises in statistical communication theory [213, 279]. We have $K(x,t) = -e^{-\alpha|x-t|}/2$, where $\alpha > 0$ represents the characteristic correlation time of the process. Here, $y(x)$ represents a signal and $u(x)$ a control for the signal. In this case, the norm of the kernel $K(x,t)$ as defined in (3.37) is approximately equal to 0.142, so that we can state existence and uniqueness of solution for a given control.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

In addition, we take $g(x) = 2/\pi, u(x) = \sin(\pi x)$ and a target function which is discontinuous as typical in the modeling of signals. We choose

$$z(x) = \lfloor 5[x(x-1)/\pi + (1+1/\pi^2)\sin(\pi x)]\rfloor/5,$$

where $\lfloor\ \rfloor$ is the floor function.

In Table 5.24, results obtained with the CGS iteration are reported. We see that tracking improves as ν decreases. We obtain robust tracking despite z being discontinuous. Next, we consider the same setting and apply the CSMG multigrid scheme with the smoother given by Algorithm 5.16. Results with the multigrid scheme are reported in Table 5.25. We obtain that both the CGS and the multigrid scheme converge efficiently to the solution and the observed convergence rate is weakly dependent on the mesh size. As ν becomes smaller, convergence rates improve, thus showing robustness. We see that multigrid convergence rates are two orders of magnitude better than the CGS convergence rates. Moreover, the observed rates are in good agreement with the corresponding estimates by local Fourier analysis. In Figure 5.22, we depict the computed optimal state and control solutions.

Table 5.24. *Results with the CGS scheme;* $\alpha = 1$.

ν	N	$\|y-z\|$	$\|\|res\|\|_2$	ρ	N_{iter}
1e-03	512	6.66e-04	9.03e-015	7.02e-04	6
1e-03	256	6.66e-04	8.89e-015	7.01e-04	6
1e-03	128	6.66e-04	8.60e-015	6.99e-04	6
1e-04	512	6.67e-05	1.32e-013	4.10e-05	5
1e-04	256	6.67e-05	1.30e-013	4.09e-05	5
1e-04	128	6.66e-05	1.26e-013	4.06e-05	5
1e-05	512	6.67e-06	8.33e-014	4.10e-06	4
1e-05	256	6.67e-06	8.27e-014	4.09e-06	4
1e-05	128	6.66e-06	8.12e-014	4.06e-06	4
1e-06	512	6.67e-07	8.56e-016	4.21e-07	4
1e-06	256	6.67e-07	8.32e-016	4.12e-07	4
1e-06	128	6.66e-07	8.08e-016	4.04e-07	4

Local Fourier Analysis of a CSMG Solver of Integral Control Problems

In this section, we discuss the local Fourier analysis of the CSMG scheme for solving Fredholm control problems. We consider the case of two levels with a fine-grid level with mesh size $h = h_k$ and the coarse-grid problem is constructed on the grid with mesh size $H = h_{k-1}$. Recall the optimality system

$$y_h - f_h(y_h) - p_h/\nu = g_h, \tag{5.241}$$

$$p_h - f_h^T(p_h) + y_h = z_h. \tag{5.242}$$

Let $(e_y(j), e_p(j)) = \sum_\theta W_\theta \phi(j,\theta)$ denote the errors for the state and the adjoint variables on the grid points $x_j = jh$. Here, $W_\theta = (Y_\theta, P_\theta)$ are the corresponding Fourier coefficients. The action of one smoothing step on the errors can be expressed by $W_\theta^{(1)} = \hat{S}(\theta) W_\theta^{(0)}$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Table 5.25. *Results with the CSMG scheme; $\alpha = 1$ and $m_1 = 1, m_2 = 1$ pre- and postsmoothing sweeps.*

ν	N	$\|y - z\|$	$\|\|res\|\|_2$	ρ	N_{cycle}
1e-03	512	6.66e-04	3.74e-016	1.07e-05	3
1e-03	256	6.66e-04	3.65e-016	1.04e-05	3
1e-03	128	6.66e-04	3.48e-016	1.00e-05	3
1e-04	512	6.67e-05	3.52e-013	1.04e-06	2
1e-04	256	6.67e-05	3.51e-013	1.03e-06	2
1e-04	128	6.66e-05	3.48e-013	1.03e-06	2
1e-05	512	6.67e-06	3.52e-015	1.03e-07	2
1e-05	256	6.67e-06	3.52e-015	1.03e-07	2
1e-05	128	6.66e-06	3.48e-015	1.02e-07	2
1e-06	512	6.67e-07	1.62e-016	4.76e-08	2
1e-06	256	6.67e-07	1.27e-016	3.75e-08	2
1e-06	128	6.66e-07	1.08e-016	3.18e-08	2

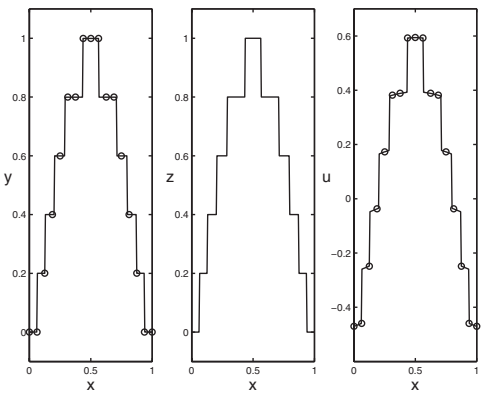

Figure 5.22. *Optimal solution y_N (left), target z, and control u_N (right) for $\alpha = 1$, $\nu = 10^{-3}$, $N = 8$, with the Nyström formula. Circles are the interpolation points. Reprinted with permission from M. Annunziato and A. Borzì, Fast solvers of Fredholm optimal control problems, Numer. Math. Theory Methods Appl.,* 3(4) (2010), 431–448.

Now, consider applying the CGS step for solving our distributed control problem. We assume that the kernel is symmetric, i.e., $w_{i-j} = w_{|i-j|}$, and that it is decaying sufficiently fast such that we can truncate the sum approximating the integral, $\sum_{k=-\ell}^{\ell} \phi(k) \approx \sum_{k=-N}^{N} \phi(k)$. Substituting $(e_y(j), e_p(j))$ in (5.239) and applying the CGS Algorithm 5.16, we obtain

$$\begin{pmatrix} (1 - h\sum_{k=-\ell}^{0} w_{|k|} e^{i\theta k}) & -\frac{1}{\nu} \\ 1 & (1 - h\sum_{k=-\ell}^{0} w_{|k|} e^{i\theta k}) \end{pmatrix} \begin{pmatrix} Y_\theta^{(1)} \\ P_\theta^{(1)} \end{pmatrix} = \begin{pmatrix} h\sum_{k=1}^{\ell} w_{|k|} e^{i\theta k} & 0 \\ 0 & h\sum_{k=1}^{\ell} w_{|k|} e^{i\theta k} \end{pmatrix} \begin{pmatrix} Y_\theta^{(0)} \\ P_\theta^{(0)} \end{pmatrix}.$$

Downloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Hence

$$\hat{S}(\theta) = \begin{pmatrix} (1 - h\sum_{k=-\ell}^{0} w_{|k|} e^{i\theta k}) & -\frac{1}{\nu} \\ 1 & (1 - h\sum_{k=-\ell}^{0} w_{|k|} e^{i\theta k}) \end{pmatrix}^{-1} \times \begin{pmatrix} h\sum_{k=1}^{\ell} w_{|k|} e^{i\theta k} & 0 \\ 0 & h\sum_{k=1}^{\ell} w_{|k|} e^{i\theta k} \end{pmatrix}. \tag{5.243}$$

With local Fourier analysis, we can estimate the smoothing factor μ. It appears that μ is almost independent of the value of the discretization parameter h and increases by increasing the value of the weight ν. Also in this case, we can perform a TG Fourier analysis to estimate the convergence factor of the multigrid iteration.

Consider the optimality system (5.241)–(5.242). We have that the symbol of the fine-grid operator is as follows

$$\widehat{A}_h(\theta) = \begin{bmatrix} a_y^h(\theta) & -1/\nu & 0 & 0 \\ 1 & a_p^h(\theta) & 0 & 0 \\ 0 & 0 & a_y^h(\bar{\theta}) & -1/\nu \\ 0 & 0 & 1 & a_p^h(\bar{\theta}) \end{bmatrix},$$

where

$$a_y^h(\theta) = 1 - h\sum_{k=-\ell}^{\ell} w_{|k|} e^{i\theta k}, \qquad a_p^h(\theta) = a_y^h(\theta).$$

Similarly, for the frequency represented on the coarse grid, the symbol of the coarse-grid operator is as follows

$$\widehat{A}_H(\theta) = \begin{bmatrix} a_y^H(2\theta) & -1/\nu \\ 1 & a_p^H(2\theta) \end{bmatrix}.$$

In Table 5.26, we report theoretical estimates of $\rho(TG_h^H)$ resulting from the TG convergence analysis. Comparison with results of numerical experiments show that these estimates are sharp; see Table 5.25 for results of numerical experiments. We can see that the local Fourier analysis predicts mesh-independent smoothing factors and convergence factors and these factors improve as ν becomes smaller.

Table 5.26. *Estimates for $\rho(TG_h^H)$ for the case of $m_1 = m_2 = 1$ smoothing steps; $w_{|i-j|} = -\frac{1}{2}\exp(-|i-j|h)$.*

ν	$N = 64$	$N = 128$	$N = 256$	$N = 512$
1.0e-01	3.31e-03	3.35e-03	3.36e-03	3.37e-03
1.0e-02	3.74e-04	3.77e-04	3.79e-04	3.80e-04
1.0e-03	3.78e-05	3.82e-05	3.84e-05	3.85e-05
1.0e-04	3.79e-06	3.83e-06	3.85e-06	3.86e-06
1.0e-05	3.79e-07	3.83e-07	3.85e-07	3.86e-07
1.0e-06	3.80e-08	3.83e-08	3.85e-08	3.86e-08

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

5.7.8 Optimization Properties of the CSMG Scheme

The CSMG strategy aims at solving optimality systems that represent first-order necessary conditions for PDE-based optimization problems. We notice that intermediate steps of the CSMG solution process may well be unfeasible or result in increasing values of the objective to be minimized. These phenomena can be observed in the case of nonconvex and nonlinear optimization problems. In fact, in the construction of the CSMG scheme it is not required to provide a minimizing sequence, but to solve the optimality system. Nevertheless, as discussed in [60], the CSMG scheme can be defined in such a way to provide a minimizing sequence as in the case of the MGOPT method.

For this discussion consider the following optimization problem

$$\begin{cases} \min J(y,u) := h(y) + \nu\, g(u), \\ \qquad\qquad c(y,u) = 0 \quad \text{in } \Omega. \end{cases} \tag{5.244}$$

Here g and h are required to be continuously differentiable, bounded from below, and such that $g(u) \to \infty$ as $||u|| \to \infty$. Allowing g and h to be locally nonconvex and e to be possibly nonlinear, (5.244) may have multiple extremals including minima, maxima, and saddle points. Local minima satisfy the first-order necessary conditions given by the optimality system

$$\begin{aligned} c(y,u) &= 0, \\ c_y(y,u)^* p &= -h'(y), \\ \nu\, g'(u) + c_u^* p &= 0. \end{aligned} \tag{5.245}$$

To guarantee a CSMG cycle that is minimizing the objective, we should define the smoothing process based on the gradient of the reduced cost functional and show that the coarse-grid correction step provides a descent update.

In order to update the control function in the smoothing process, we could use the following descent scheme (as in [10])

$$u_h^{new} = u_h - \beta\,(\nu\, g'(u_h) + c_u^*\, p_h(u_h)), \tag{5.246}$$

where optimal choice of the scaling factor $\beta > 0$ may be done using linesearch methods. Alternatively, one could use a smoothing step based on subspace correction methods [334]; see also [60]. However, notice that local Newton updates are not a suitable approach in the case of nonconvex problems because the Newton scheme may well converge to a maximum instead. In this case, in [60] the search for possible negative eigenvalues of the reduced Hessian considered at the coarsest grid of the multigrid process is discussed. If negative eigenvalues are detected, a globalization step in the direction of negative curvature is performed to escape undesired maxima or saddle points.

Now, let us discuss the coarse-grid correction procedure. Consider the case where $c(y,u) = -\Delta y - u$ and $h(y) = ||y - z||^2_{L^2(\Omega)}/2$, and assume that g' is Lipschitz continuous with Lipschitz constant $\gamma > 0$ and satisfies the following monotonicity requirement

$$(g'(u) - g'(v), u - v) \geq \delta\, \|u - v\|^2_{L^2(\Omega)}$$

for some $\delta > 0$. We require that this property hold after discretization and we show that the CSMG coarse-grid correction provides a descent direction in the sense that

$$(\nu\, g'(u_h) - p_h, I_H^h(u_H - \hat{I}_h^H u_h))_h < 0,$$

unless $u_H = \hat{I}_h^H u_h$, occurring at convergence.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Starting from an initial approximation and after a few presmoothing steps the resulting triple (y_h, u_h, p_h) satisfies the optimality system up to residuals (d_h^1, d_h^2, d_h^3), that is,

$$\begin{aligned} -\Delta_h y_h - u_h &= d_h^1, \\ -\Delta_h p_h + y_h - z_h &= d_h^2, \\ \nu g'(u_h) - p_h &= d_h^3. \end{aligned} \tag{5.247}$$

For the coarse-grid process, we take $\hat{I}_h^h = I_h^H$, where I_h^H is the full-weighting restriction operator. For I_H^h we choose bilinear interpolation, i.e., $(I_h^H u_h, v_H)_H = (u_h, I_H^h v_H)_h$. We define $z_H = I_h^H z_h$. With this setting, we obtain the following coarse-grid equations

$$\begin{aligned} -\Delta_H y_H - u_H &= I_h^H \Delta_h y_h - \Delta_H I_h^H y_h, \\ -\Delta_H p_H + y_H - z_H &= I_h^H \Delta_h p_h - \Delta_H I_h^H p_h, \\ \nu g'(u_H) - p_H &= 0. \end{aligned} \tag{5.248}$$

As usual in TG convergence analysis, we assume that this coarse system of equations is solved exactly. From the first equation of (5.248), and using the corresponding equation in (5.247), we obtain

$$u_H - I_h^H u_h = -\Delta_H (y_H - I_h^H y_h) + I_h^H d_h^1. \tag{5.249}$$

Combining the fine and coarse adjoint equations we have

$$p_H - I_h^H p_h = \Delta_H^{-1}(y_H - I_h^H y_h) + \Delta_H^{-1} I_h^H d_h^2. \tag{5.250}$$

Let us assume that

$$(g'(v_H) - I_h^H g'(v_h), v_H - I_h^H v_h)_H \geq \delta' \| v_H - I_h^H v_h \|_H^2 \tag{5.251}$$

for some $\delta' > 0$ independent of v_h and v_H. Note that (5.251) is satisfied, for example, if g' is linear or if I_h^H is strict injection and g is strictly convex.

With these preparations we are ready to show that the update step of the CSMG coarse-grid correction follows a descent direction

$$\begin{aligned} &(\nu g'(u_h) - p_h, I_H^h(u_H - I_h^H u_h))_h = (I_h^H(\nu g'(u_h) - p_h), u_H - I_h^H u_h)_H \\ &= (\nu I_h^H g'(u_h) - I_h^H p_h, u_H - I_h^H u_h)_H \\ &= (\nu I_h^H g'(u_h) - p_H + \Delta_H^{-1}(y_H - I_h^H y_h) + \Delta_H^{-1} I_h^H d_h^2, u_H - I_h^H u_h)_H \\ &= -\nu (g'(u_H) - I_h^H g'(u_h), u_H - I_h^H u_h)_H \\ &\quad + (\Delta_H^{-1}(y_H - I_h^H y_h) + \Delta_H^{-1} I_h^H d_h^2, -\Delta_H (y_H - I_h^H y_h) + I_h^H d_h^1)_H \\ &= -\nu (g'(u_H) - I_h^H g'(u_h), u_H - I_h^H u_h)_H - (y_H - I_h^H y_h, y_H - I_h^H y_h)_H \\ &\quad + (\Delta_H^{-1}(y_H - I_h^H y_h), I_h^H d_h^1)_H - (\Delta_H^{-1} I_h^H d_h^2, \Delta_H (y_H - I_h^H y_h))_H \\ &\quad + (\Delta_H^{-1} I_h^H d_h^2, I_h^H d_h^1)_H \\ &\leq -\nu (g'(u_H) - I_h^H g'(u_h), u_H - I_h^H u_h)_H \\ &\quad + \frac{1}{2}(\| \Delta_H^{-1} I_h^H d_h^1 \|_H^2 + \| \Delta_H^{-1} I_h^H d_h^2 \|_H^2 + \| I_h^H d_h^1 \|_H^2 + \| I_h^H d_h^2 \|_H^2) \\ &\leq -\nu \delta' \| u_H - I_h^H u_h \|_H^2 \\ &\quad + \frac{1}{2}(\| \Delta_H^{-1} I_h^H d_h^1 \|_H^2 + \| \Delta_H^{-1} I_h^H d_h^2 \|_H^2 + \| I_h^H d_h^1 \|_H^2 + \| I_h^H d_h^2 \|_H^2). \end{aligned}$$

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privac

Therefore

$$(\nu\, g'(u_h) - p_h, I_H^h(u_H - I_h^H u_h))_h < 0$$

if (5.251) holds and the residuals d_h^1 and d_h^2 are sufficiently small.

Finally we show that the coarse-grid correction step does not produce overshooting in the sense that $(\hat{J}'(u_h)_h, \hat{J}'(u_h^{new})_h)_h \geq 0$. We consider the case where $g'(u) = u$. We have the following

$$\begin{aligned}
&(\hat{J}'(u_h)_h, \hat{J}'(u_h^{new})_h)_h \\
&= (\nu u_h - p_h, \nu(u_h + I_H^h(u_H - I_h^H u_h)) - (p_h + I_H^h(p_H - I_h^H p_h))_h \\
&= \|\nu u_h - p_h\|_h^2 + (\nu u_h - p_h, I_H^h[\nu(u_H - I_h^H u_h) - (p_H - I_h^H p_h)])_h \\
&= \|\nu u_h - p_h\|_h^2 - \|I_h^H(\nu u_h - p_h)\|_H^2 \geq 0,
\end{aligned}$$

where we use $\|I_h^H\| \leq 1$.

ownloaded 05/28/24 to 129.161.43.217 by Vignesh Ramakrishnan (ramakv3@rpi.edu). Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-priva