A RATIONAL DEFERRED CORRECTION APPROACH TO PDE-CONSTRAINED OPTIMIZATION

STEFAN GÜTTEL* AND JOHN W. PEARSON†

Abstract. The accurate and efficient solution of time-dependent PDE-constrained optimization problems is a challenging task, in large part due to the very high dimension of the matrix systems that need to be solved. We devise a new deferred correction method for coupled systems of time-dependent PDEs, allowing one to iteratively improve the accuracy of low-order time stepping schemes. We consider two variants of our method, a splitting and a coupling version, and analyze their convergence properties. We then test our approach on a number of PDE-constrained optimization problems. We obtain solution accuracies far superior to that achieved when solving a single discretized problem, in particular in cases where the accuracy is limited by the time discretization. Our approach allows for the direct reuse of existing solvers for the resulting matrix systems, as well as state-of-the-art preconditioning strategies.

Key words. PDE-constrained optimization, deferred correction, time-dependent PDE, coupled system

AMS subject classifications. 93C20, 65B05, 34H05, 65N22

1. Introduction. The accurate and efficient solution of partial differential equations (PDEs) has been an active area of research in computational mathematics and applied science for many decades. More recently, a class of problems which has gathered considerable attention is that of *PDE-constrained optimization*: here a cost functional is sought to be minimized with one or more PDEs acting as constraints. Such problems have many interesting applications, for example, in flow control [15], medical imaging [1, 6], finance [5], and the control of chemical and biological processes [2, 11, 14]. For an overview of the field of PDE-constrained optimization, we recommend [4, 18, 42]. The efficient solution of PDE-constrained optimization problems is a highly challenging task. This is especially true for time-dependent problems, which typically require the solution of very large matrix systems arising from the discretization of time derivatives.

There are many possible forms of the cost functional to be minimized. Perhaps the most common model, and the one we examine in this article, is of the form

$$\min_{y,u} \quad \frac{1}{2}\int_0^T \int_\Omega (y - y_d)^2 \, \mathrm{d}\Omega \, \mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega u^2 \, \mathrm{d}\Omega \, \mathrm{d}t.$$

Here y denotes the *state variable* of the problem, which one wishes to be "as close as possible" in some sense to the *desired state* y_d, and u represents the *control variable*. The quantity $\beta > 0$ is the *Tikhonov regularization parameter*, which determines to what extent one wishes to achieve realization of the desired state and minimization of the control. The functional is posed on a spatial domain $\Omega \subset \mathbb{R}^d$ in $d \in \{2, 3\}$ dimensions with boundary $\partial\Omega$, and over the time interval $[0, T]$. Of course it is also possible to consider cost functionals that involve norms other than $L_2(\Omega \times [0, T])$.

For the *distributed control problems* considered here (i.e., problems where the control function u acts on the whole domain Ω), the PDE constraint is of the form $\mathcal{D}y = u$, with some differential operator $\mathcal{D}$. For instance, for the well-studied heat equation control and convection–diffusion control problems, $\mathcal{D}$ is given by

$$\mathcal{D} = \frac{\partial}{\partial t} - \nabla^2 \quad \text{and} \quad \mathcal{D} = \frac{\partial}{\partial t} - \nu\nabla^2 + \mathbf{w} \cdot \nabla, \tag{1.1}$$

respectively. Here, ν represents viscosity and $\mathbf{w}$ denotes some wind vector, which may be dependent on the spatial variable. It is also possible to consider a system of PDEs as constraints: in particular, for

*School of Mathematics, The University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom, stefan.guettel@manchester.ac.uk

†School of Mathematics, Statistics and Actuarial Science, University of Kent, Cornwallis Building (East), Canterbury, CT2 7NF, United Kingdom, j.w.pearson@kent.ac.uk

the time-dependent Stokes control problem,

$$\mathcal{D} = \begin{cases} \begin{bmatrix} \frac{\partial}{\partial t} - \nabla^2 & & \frac{\partial}{\partial x_1} \\ & \frac{\partial}{\partial t} - \nabla^2 & \frac{\partial}{\partial x_2} \\ -\frac{\partial}{\partial x_1} & -\frac{\partial}{\partial x_2} & O \end{bmatrix}, & \text{if } d = 2, \\ \begin{bmatrix} \frac{\partial}{\partial t} - \nabla^2 & & & \frac{\partial}{\partial x_1} \\ & \frac{\partial}{\partial t} - \nabla^2 & & \frac{\partial}{\partial x_2} \\ & & \frac{\partial}{\partial t} - \nabla^2 & \frac{\partial}{\partial x_3} \\ -\frac{\partial}{\partial x_1} & -\frac{\partial}{\partial x_2} & -\frac{\partial}{\partial x_3} & O \end{bmatrix}, & \text{if } d = 3, \end{cases}$$

where $\mathbf{x} = [x_1, x_2]^\top$ or $[x_1, x_2, x_3]^\top$ denote the spatial coordinates of the problem. In each case, the PDEs are accompanied by appropriate initial and boundary conditions. The range of problems which can be considered is vast, just as is the range of application areas.

Devising strategies for solving such complex time-dependent problems efficiently is an important and challenging problem. In this work, we consider the solution of PDE-constrained optimization problems using the *deferred correction* framework. Deferred correction can be interpreted as an extrapolation scheme where the accuracy of a low-order integrator is iteratively improved by repeatedly solving correction equations for the error. The idea goes back to early work by Zadundaisky [43, 44] and Pereyra [31, 32]. The important ingredient of deferred correction methods for ordinary differential equations (ODEs) is a high-order representation of the residual, often based on polynomial interpolation of the solution over the time interval $[0, T]$. An important contribution of Dutt, Greengard & Rokhlin [7] was to point out that such a polynomial scheme requires the interpolation nodes to be chosen carefully for ensuring numerical stability. The resulting class of methods is referred to as *spectral deferred correction*. These methods have been applied successfully to a wide range of PDE and ODE problems, e.g., recently for the purpose of parallel-in-time integration [20, 21].

A computational drawback of spectral deferred correction is that the interpolation nodes should not be chosen equidistant on $[0, T]$, leading to varying time steps in the low-order integrator. With Chebyshev nodes, for example, the interpolation nodes have an inverse square root density at the endpoints of the time interval. As explained in [17], this can be very inconvenient in particular with implicit integration schemes where linear matrix systems with the Jacobian have to be solved at every time step. When direct linear system solvers are used, the Jacobian matrix needs to be refactored very often and in the worst case at every time step. When iterative solvers are used, non-equal time steps are also inconvenient as a large body of literature on the preconditioning of PDE-constrained optimization problems crucially assumes *equidistant* time steps (see, e.g., [26, 27, 28, 37, 38, 39, 41, 45]).

To overcome the problems mentioned in the previous paragraph, we generalize the rational deferred correction (RDC) method presented in [17] to PDE-constrained optimization problems. The RDC method is based on the barycentric rational interpolants developed by Floater & Hormann [10]. These interpolants achieve a high rate of approximation even with equidistant interpolation nodes (provided the so-called *blending parameter* is chosen carefully). This allows for stable high-order integration of interpolants at equidistant time nodes, which is a mandatory ingredient for a practical deferred correction scheme. The resulting method solves PDE-constrained optimization problems to much higher accuracy than conventional methods, while requiring fewer time steps and less memory. Moreover, the involved linear systems are of significantly smaller size but can still be treated by existing solution techniques, including the reuse of previously developed preconditioners.

The outline of this work is as follows. In section 2 we review the optimize-then-discretize approach for PDE-constrained optimization problems, yielding systems of high-dimensional coupled ODEs and corresponding large linear systems to be solved. Section 3 then introduces two deferred correction approaches for the solution of these ODEs, a splitting-based and a coupling-based approach. We find that the coupling-based approach performs much better on linear control problems and in section 4 we give some theoretical insight into this observation. In section 5 we discuss several computational aspects of our method. Numerical experiments are given in section 6, followed by conclusions in section 7.

2. Discretization of PDE-constrained optimization problems. In this section we develop discretizations of selected PDE-constrained optimization problems. To this end, we apply a *optimize-then-discretize approach*, meaning that we derive optimality conditions on the continuous space and then select an appropriate discretization. The alternative *discretize-then-optimize approach*, which involves deriving a discrete cost functional and using it to write optimality conditions on the discrete space, is also popular within the PDE-constrained optimization community, but is less applicable for the methods presented in this paper.

We first consider the heat equation control problem (see also [42] for a more detailed derivation)

$$\begin{aligned} \min_{y,u} \quad & \frac{1}{2}\int_0^T \int_\Omega \left(y - y_d\right)^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega u^2 \,\mathrm{d}\Omega\,\mathrm{d}t \\ \text{s.t.} \quad & \frac{\partial y}{\partial t} - \nabla^2 y = u && \text{in } \Omega \times [0,T], \\ & y(\vec{x},t) = h(\vec{x},t) && \text{on } \partial\Omega \times [0,T], \\ & y(\vec{x},0) = y_0(\vec{x}) && \text{at } t = 0. \end{aligned} \tag{2.1}$$

We now wish to find the continuous optimality conditions for the Lagrangian

$$\begin{aligned} \mathcal{L} = \frac{1}{2}\int_0^T \int_\Omega \left(y - y_d\right)^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega u^2 \,\mathrm{d}\Omega\,\mathrm{d}t \\ + \int_0^T \int_\Omega \left(\frac{\partial y}{\partial t} - \nabla^2 y - u\right)\lambda_\Omega \,\mathrm{d}\Omega\,\mathrm{d}t + \int_0^T \int_{\partial\Omega} (y - g)\,\lambda_{\partial\Omega} \,\mathrm{d}s\,\mathrm{d}t, \end{aligned}$$

where the *Lagrange multiplier* (or *adjoint variable*) λ has components λ_Ω and $\lambda_{\partial\Omega}$ on the interior and boundary of Ω, respectively. Here the initial condition $y(\vec{x},0) = y_0(\vec{x})$ is absorbed into the Lagrangian.

From here, the continuous optimality conditions are obtained by differentiating $\mathcal{L}$ with respect to the adjoint, control, and state variables. Firstly, differentiating with respect to λ (on the interior and boundary in turn) returns the *forward problem*

$$\begin{aligned} \frac{\partial y}{\partial t} - \nabla^2 y = u && \text{in } \Omega \times [0,T], \\ y(\vec{x},t) = g(\vec{x},t) && \text{on } \partial\Omega \times [0,T], \\ y(\vec{x},0) = y_0(\vec{x}) && \text{at } t = 0. \end{aligned}$$

Next, differentiating with respect to u gives us the *gradient equation*

$$\beta u - \lambda = 0.$$

Finally, differentiating with respect to y gives the *adjoint problem*

$$\begin{aligned} -\frac{\partial \lambda}{\partial t} - \nabla^2 \lambda = y_d - y && \text{in } \Omega \times [0,T], \\ \lambda(\vec{x},t) = 0 && \text{on } \partial\Omega \times [0,T], \\ \lambda(\vec{x},T) = 0 && \text{at } t = 0. \end{aligned}$$

We now use the proportionality of control and adjoint, given by the gradient equation, to observe that the conditions reduce to a coupled system of PDEs

$$\frac{\partial y}{\partial t} - \nabla^2 y = \frac{1}{\beta}\lambda, \tag{2.2}$$

$$-\frac{\partial \lambda}{\partial t} - \nabla^2 \lambda = y_d - y, \tag{2.3}$$

with initial condition on y, final condition on λ, and boundary conditions on y and λ. These are referred to the *first-order optimality conditions* (or *Karush–Kuhn–Tucker conditions*) for this problem.

For this particular (self-adjoint) PDE, the Laplacian is applied to both y and λ within the coupled system. In general this is not the case and, in fact, for problems of this form the PDE in λ relates to the adjoint operator $\mathcal{D}^*$. For example, if the original PDE constraint were the time-dependent convection–diffusion problem given in (1.1), the optimality conditions would read

$$\frac{\partial y}{\partial t} - \nabla^2 y + (\vec{w}\cdot\nabla)y = \frac{1}{\beta}\lambda, \tag{2.4}$$

$$-\frac{\partial \lambda}{\partial t} - \nabla^2 \lambda - (\vec{w}\cdot\nabla)\lambda = y_d - y. \tag{2.5}$$

There is also no reason why the constraints should always arise in the form of a single PDE. For example, let us consider the following Stokes control problem:

$$\begin{aligned}
\min_{\vec{y},\vec{u}} \quad & \frac{1}{2}\int_0^T \int_\Omega \|\vec{y}-\vec{y}_d\|^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta}{2}\int_0^T\int_\Omega \|\vec{u}\|^2 \,\mathrm{d}\Omega\,\mathrm{d}t, \qquad (2.6)\\
\text{s.t.} \quad & \frac{\partial \vec{y}}{\partial t} - \nabla^2\vec{y} + \nabla p = \vec{u} + \vec{z} && \text{in } \Omega\times[0,T],\\
& -\nabla\cdot\vec{y} = 0 && \text{in } \Omega\times[0,T],\\
& \vec{y}(\vec{x},t) = \vec{h}(\vec{x},t) && \text{on } \partial\Omega\times[0,T],\\
& \vec{y}(\vec{x},0) = \vec{y}_0(\vec{x}) && \text{at } t=0.
\end{aligned}$$

In this problem setup, $\vec{y}$ denotes the velocity of a fluid over d dimensions, with p representing pressure, and $\vec{z}$ some (given) function. The variable $\vec{u}$ is the control variable over d dimensions. The continuous Lagrangian which we then seek to minimize is given by

$$\begin{aligned}
\mathcal{L} = {} & \frac{1}{2}\int_0^T \int_\Omega \|\vec{y}-\vec{y}_d\|^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta}{2}\int_0^T\int_\Omega \|\vec{u}\|^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \int_0^T\int_\Omega \left(\frac{\partial\vec{y}}{\partial t} - \nabla^2\vec{y} + \nabla p - \vec{u} - \vec{z}\right)\vec{\lambda}_\Omega \,\mathrm{d}\Omega\,\mathrm{d}t\\
& + \int_0^T\int_\Omega (\vec{y}-\vec{h})\vec{\lambda}_{\partial\Omega}\,\mathrm{d}\Omega\,\mathrm{d}t + \int_0^T\int_\Omega \mu(-\nabla\cdot\vec{y})\,\mathrm{d}\Omega\,\mathrm{d}t,
\end{aligned}$$

where $\vec{\lambda}$ (over d dimensions) and μ denote the adjoint variables for velocity and pressure, respectively. The variable $\vec{\lambda}$ is denoted as $\vec{\lambda}_\Omega$ and $\vec{\lambda}_{\partial\Omega}$ within the interior and on the boundary of Ω, respectively.

When differentiating with respect to the adjoint variables, one recovers the forward problem

$$\begin{aligned}
& \frac{\partial \vec{y}}{\partial t} - \nabla^2\vec{y} + \nabla p = \vec{u} + \vec{z} && \text{in } \Omega\times[0,T],\\
& -\nabla\cdot\vec{y} = \vec{0} && \text{in } \Omega\times[0,T],\\
& \vec{y}(\vec{x},t) = \vec{h}(\vec{x},t) && \text{on } \partial\Omega\times[0,T],\\
& \vec{y}(\vec{x},0) = \vec{y}_0(\vec{x}) && \text{at } t=0.
\end{aligned}$$

The gradient equation (obtained by differentiating with respect to the control) is given by $\beta\vec{u} - \vec{\lambda} = 0$, and differentiating with respect to the state variables gives the adjoint equations

$$\begin{aligned}
& -\frac{\partial\vec{\lambda}}{\partial t} - \nabla^2\vec{\lambda} + \nabla\mu = \vec{y}_d - \vec{y} && \text{in } \Omega\times[0,T],\\
& -\nabla\cdot\vec{\lambda} = 0 && \text{in } \Omega\times[0,T],\\
& \vec{\lambda}(\vec{x},t) = \vec{0} && \text{on } \partial\Omega\times[0,T],\\
& \vec{\lambda}(\vec{x},T) = \vec{0} && \text{at } t=0.
\end{aligned}$$

Incorporating the gradient equation with the forward problem again gives a coupled system of PDEs:

$$\frac{\partial \vec{y}}{\partial t} - \nabla^2 \vec{y} + \nabla p = \frac{1}{\beta}\vec{\lambda} + \vec{z}, \tag{2.7}$$

$$-\nabla \cdot \vec{y} = 0, \tag{2.8}$$

$$-\frac{\partial \vec{\lambda}}{\partial t} - \nabla^2 \vec{\lambda} + \nabla \mu = \vec{y}_d - y, \tag{2.9}$$

$$-\nabla \cdot \vec{\lambda} = 0, \tag{2.10}$$

with initial/final conditions on $\vec{y}/\vec{\lambda}$, and boundary conditions on both.

We note that many other systems of PDEs can be written in this form, including systems which are not self-adjoint (which the Stokes system above is).

Having posed the first-order optimality conditions for our PDE-constrained optimization problems as coupled systems of PDEs using this optimize-then-discretize approach, we now wish to consider the discretization of these PDEs. We first observe that all of the systems derived in this section can be discretized as

$$\begin{cases} M_u \mathbf{u}'(t) = K_1 \mathbf{u}(t) - K_2 \mathbf{v}(t) + \widehat{\mathbf{f}}(t), & M_u \mathbf{u}(0) = M_u \mathbf{u}_0 \in \mathbb{R}^N \text{ given}, \end{cases} \tag{2.11}$$

$$\begin{cases} M_v \mathbf{v}'(t) = K_3 \mathbf{u}(t) - K_4 \mathbf{v}(t) + \widehat{\mathbf{g}}(t), & M_v \mathbf{v}(T) = M_v \mathbf{v}_T \in \mathbb{R}^N \text{ given}, \end{cases} \tag{2.12}$$

with two vector functions $\mathbf{u}, \mathbf{v} : [0, T] \mapsto \mathbb{R}^N$, and the matrices $\{K_1, K_2, K_3, K_4, M_u, M_v\} \subset \mathbb{R}^{N \times N}$ arising, e.g., from finite difference, finite element, or spectral discretization of the spatial differential operators. For example, in the heat control problem (2.2)–(2.3) the individual terms are discretizations of the following operators:

$$\begin{aligned} &M_u \leftarrow I, \quad K_1 \leftarrow \nabla^2, \quad K_2 \leftarrow -\frac{1}{\beta} I, \\ &M_v \leftarrow I, \quad K_3 \leftarrow I, \quad K_4 \leftarrow \nabla^2, \\ &\mathbf{u}(t) \leftarrow y(t), \quad \mathbf{v}(t) \leftarrow \lambda(t), \quad \widehat{\mathbf{f}} \leftarrow 0, \quad \widehat{\mathbf{g}}(t) \leftarrow -y_d(t). \end{aligned}$$

For the convection–diffusion control problem (2.4)–(2.5) we have

$$\begin{aligned} &M_u \leftarrow I, \quad K_1 \leftarrow \nu \nabla^2 - (\vec{w} \cdot \nabla), \quad K_2 \leftarrow -\frac{1}{\beta} I, \\ &M_v \leftarrow I, \quad K_3 \leftarrow I, \quad K_4 \leftarrow \nu \nabla^2 + (\vec{w} \cdot \nabla), \\ &\mathbf{u}(t) \leftarrow y(t), \quad \mathbf{v}(t) \leftarrow \lambda(t), \quad \widehat{\mathbf{f}} \leftarrow 0, \quad \widehat{\mathbf{g}}(t) \leftarrow -y_d(t). \end{aligned}$$

Furthermore, for the 2D Stokes control problem (2.7)–(2.10), we have

$$M_u \leftarrow \begin{bmatrix} I & & \\ & I & \\ & & O \end{bmatrix}, \quad K_1 \leftarrow \begin{bmatrix} \nabla^2 & & -\frac{\partial}{\partial x_1} \\ & \nabla^2 & -\frac{\partial}{\partial x_2} \\ -\frac{\partial}{\partial x_1} & -\frac{\partial}{\partial x_2} & O \end{bmatrix}, \quad K_2 \leftarrow \begin{bmatrix} -\frac{1}{\beta} I & & \\ & -\frac{1}{\beta} I & \\ & & O \end{bmatrix},$$

$$M_v \leftarrow \begin{bmatrix} I & & \\ & I & \\ & & O \end{bmatrix}, \quad K_3 \leftarrow \begin{bmatrix} I & & \\ & I & \\ & & O \end{bmatrix} \quad K_4 \leftarrow \begin{bmatrix} \nabla^2 & & -\frac{\partial}{\partial x_1} \\ & \nabla^2 & -\frac{\partial}{\partial x_2} \\ -\frac{\partial}{\partial x_1} & -\frac{\partial}{\partial x_2} & O \end{bmatrix},$$

$$\mathbf{u}(t) \leftarrow \begin{bmatrix} y_1(t) \\ y_2(t) \\ p(t) \end{bmatrix}, \quad \mathbf{v}(t) \leftarrow \begin{bmatrix} \lambda_1(t) \\ \lambda_2(t) \\ \mu(t) \end{bmatrix}, \quad \widehat{\mathbf{f}}(t) \leftarrow \begin{bmatrix} z_1(t) \\ z_2(t) \\ 0 \end{bmatrix}, \quad \widehat{\mathbf{g}}(t) \leftarrow \begin{bmatrix} -y_{d,1}(t) \\ -y_{d,2}(t) \\ 0 \end{bmatrix},$$

with y_i, z_i, λ_i, $y_{d,i}$ denoting the components of $\vec{y}$, $\vec{z}$, $\vec{\lambda}$, $\vec{y}_d$ in the i-th spatial dimension. The 3D cases are analogous. For each problem I is the identity matrix of appropriate dimension.

One popular approach for solving (2.11)–(2.12) is to discretize both equations via the implicit Euler scheme at time points $t_0, t_1, \ldots, t_n$, forward and backward in time, giving rise to the recursions

$$\begin{aligned} M_u \frac{\mathbf{u}_j - \mathbf{u}_{j-1}}{\tau_j} &= K_1 \mathbf{u}_j - K_2 \mathbf{v}_j + \widehat{\mathbf{f}}_j, && M_u \mathbf{u}_0 \in \mathbb{R}^N \text{ given}, \\ M_v \frac{\mathbf{v}_j - \mathbf{v}_{j-1}}{\tau_j} &= K_3 \mathbf{u}_{j-1} - K_4 \mathbf{v}_{j-1} + \widehat{\mathbf{g}}_{j-1}, && M_v \mathbf{v}_n \in \mathbb{R}^N \text{ given}, \end{aligned}$$

where $\tau_j = t_j - t_{j-1}$. When rearranged, these equations can be written in the form

$$\begin{array}{llll} \tau_j K_3 \mathbf{u}_{j-1} & + (M_v - \tau_j K_4)\mathbf{v}_{j-1} & - M_v \mathbf{v}_j & = -\tau_j \widehat{\mathbf{g}}_{j-1}, \\ -M_u \mathbf{u}_{j-1} + (M_u - \tau_j K_1)\mathbf{u}_j & + \tau_j K_2 \mathbf{v}_{j-1} & & = \tau_j \widehat{\mathbf{f}}_j, \end{array}$$

for $j = 1, ..., n$. We may now gather these recursions into the linear system

$$\underbrace{\begin{bmatrix} \mathcal{K}_{3,\tau} & \mathcal{M}_v^\top - \mathcal{K}_{4,\tau} \\ \mathcal{M}_u - \mathcal{K}_{1,\tau} & \mathcal{K}_{2,\tau} \end{bmatrix}}_{\boldsymbol{\mathcal{A}}} \begin{bmatrix} \operatorname{vec}(\mathbf{U}) \\ \operatorname{vec}(\mathbf{V}) \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{g}}_\tau \\ \widehat{\mathbf{f}}_\tau \end{bmatrix}, \tag{2.13}$$

where $\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_n]$, $\mathbf{V} = [\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_n]$, and

$$\mathcal{M}_u = \begin{bmatrix} I & & & & & \\ -M_u & M_u & & & & \\ & -M_u & M_u & & & \\ & & \ddots & \ddots & & \\ & & & -M_u & M_u & \\ & & & & -M_u & M_u \end{bmatrix}, \qquad \widehat{\mathbf{f}}_\tau = \begin{bmatrix} \mathbf{u}_0 \\ \tau_1 \widehat{\mathbf{f}}_1 \\ \tau_2 \widehat{\mathbf{f}}_2 \\ \vdots \\ \tau_{n-1} \widehat{\mathbf{f}}_{n-1} \\ \tau_n \widehat{\mathbf{f}}_n \end{bmatrix},$$

$$\mathcal{M}_v = \begin{bmatrix} M_v & & & & & \\ -M_v & M_v & & & & \\ & -M_v & M_v & & & \\ & & \ddots & \ddots & & \\ & & & -M_v & M_v & \\ & & & & -M_v & I \end{bmatrix}, \qquad \widehat{\mathbf{g}}_\tau = \begin{bmatrix} -\tau_1 \widehat{\mathbf{g}}_0 \\ -\tau_2 \widehat{\mathbf{g}}_1 \\ \vdots \\ -\tau_{n-1} \widehat{\mathbf{g}}_{n-2} \\ -\tau_n \widehat{\mathbf{g}}_{n-1} \\ \mathbf{v}_T \end{bmatrix},$$

$$\mathcal{K}_i = \begin{cases} \begin{bmatrix} O & & & & & \\ & \tau_1 K_i & & & & \\ & & \tau_2 K_i & & & \\ & & & \ddots & & \\ & & & & \tau_{n-1} K_i & \\ & & & & & \tau_n K_i \end{bmatrix}, & i = 1, 2, \\ \begin{bmatrix} \tau_1 K_i & & & & & \\ & \tau_2 K_i & & & & \\ & & \ddots & & & \\ & & & \tau_{n-1} K_i & & \\ & & & & \tau_n K_i & \\ & & & & & O \end{bmatrix}, & i = 3, 4. \end{cases}$$

We have replaced the initial condition $M_u \mathbf{u}(0) = M_u \mathbf{u}_0$ in (2.11) with $I\mathbf{u}(0) = \mathbf{u}_0$, which can always be satisfied even if M_u is singular, but is computationally more convenient. We do likewise for (2.12).

A great deal of work has been devoted to the development of efficient preconditioners for linear systems of the form (2.13); see [19, 27, 28, 30, 46] for example. With this optimize-then-discretize approach, however, the accuracy is limited to that of the time-stepping scheme, i.e., first-order accuracy

in time in this case. Hence memory consumption can become very large if the number of time steps needs to be increased due to accuracy requirements. Unfortunately, the forward–backward structure of the equations does not easily allow one to split the time interval $[0, T]$ into smaller chunks without introducing additional coupling conditions. The aim of this work is to show how deferred correction can be employed to enhance the accuracy in time by repeatedly solving linear systems with the original time discretization matrices, allowing one to reuse existing preconditioners and compute more accurate solutions than previously possible.

3. Deferred correction. As we have seen in the previous section, we are required to solve coupled initial/final value problems

$$\begin{cases} M_u \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}, \mathbf{v}), & M_u \mathbf{u}(0) = M_u \mathbf{u}_0 \in \mathbb{R}^N \text{ given}, \quad (3.1) \\ M_v \mathbf{v}'(t) = \mathbf{g}(t, \mathbf{u}, \mathbf{v}), & M_v \mathbf{v}(T) = M_v \mathbf{v}_T \in \mathbb{R}^N \text{ given}, \quad (3.2) \end{cases}$$

for two vector-valued functions $\mathbf{u}, \mathbf{v} : [0, T] \mapsto \mathbb{R}^N$. Let us assume that approximations $\mathbf{u}_j$ and $\mathbf{v}_j$ at time points $0 = t_0 < t_1 < \ldots < t_n = T$ are available and consider the interpolants

$$\widetilde{\mathbf{u}}(t) = \sum_{j=0}^{n} \ell_j(t) \mathbf{u}_j \quad \text{and} \quad \widetilde{\mathbf{v}}(t) = \sum_{j=0}^{n} \ell_j(t) \mathbf{v}_j, \tag{3.3}$$

where $\ell_j(t)$ are differentiable Lagrange functions satisfying $\ell_j(t_i) = \delta_{ij}$. The concrete choice of these functions will be discussed in section 3.4. Our aim is to compute improved approximations for $\mathbf{u}(t)$ and $\mathbf{v}(t)$, and in section 3.1 we will describe the general deferred correction framework for this task. Afterwards we present two variants of deferred correction particularly tailored to our application. The first variant, which is described in section 3.2 and referred to as *splitting approach*, is applicable when $\mathbf{f}$ and $\mathbf{g}$ are nonlinear or linear functions. Although nonlinear problems are not the main focus of this paper, we include this approach for its generality. In section 3.3 we discuss a *coupling approach* which is only applicable to linear problems but is typically more efficient.

3.1. General derivation. We start by using the Picard formulations of (3.1)–(3.2), which are

$$\begin{cases} M_u \mathbf{u}(t) &= M_u \mathbf{u}_0 + \displaystyle\int_0^t \mathbf{f}(\tau, \mathbf{u}(\tau), \mathbf{v}(\tau)) \, \mathrm{d}\tau, \quad (3.4) \\ M_v \mathbf{v}(t) &= M_v \mathbf{v}_n + \displaystyle\int_T^t \mathbf{g}(\tau, \mathbf{u}(\tau), \mathbf{v}(\tau)) \, \mathrm{d}\tau, \quad (3.5) \end{cases}$$

or equivalently,

$$\begin{cases} M_u(\widetilde{\mathbf{u}}(t) + \mathbf{e}_u(t)) &= M_u \mathbf{u}_0 + \displaystyle\int_0^t \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau) + \mathbf{e}_u(\tau), \widetilde{\mathbf{v}}(\tau) + \mathbf{e}_v(\tau)) \, \mathrm{d}\tau, \\ M_v(\widetilde{\mathbf{v}}(t) + \mathbf{e}_v(t)) &= M_v \mathbf{v}_n + \displaystyle\int_T^t \mathbf{g}(\tau, \widetilde{\mathbf{u}}(\tau) + \mathbf{e}_v(\tau), \widetilde{\mathbf{v}}(\tau) + \mathbf{e}_v(\tau)) \, \mathrm{d}\tau, \end{cases}$$

with some unknown error functions $\mathbf{e}_u(t)$ and $\mathbf{e}_v(t)$. Using (3.4)–(3.5) to define the residuals

$$\begin{cases} \mathbf{r}_u(t) &:= M_u \mathbf{u}_0 + \displaystyle\int_0^t \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau)) \, \mathrm{d}\tau - M_u \widetilde{\mathbf{u}}(t), \quad (3.6) \\ \mathbf{r}_v(t) &:= M_v \mathbf{v}_n + \displaystyle\int_T^t \mathbf{g}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau)) \, \mathrm{d}\tau - M_v \widetilde{\mathbf{v}}(t), \quad (3.7) \end{cases}$$

we immediately find

$$\begin{aligned} M_u \mathbf{e}_u(t) &= \mathbf{r}_u(t) + \int_0^t \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau) + \mathbf{e}_u(\tau), \widetilde{\mathbf{v}}(\tau) + \mathbf{e}_v(\tau)) - \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau)) \, \mathrm{d}\tau \quad (3.8) \\ &= \mathbf{r}_u(t) + \int_0^t \mathbf{h}_f(\tau, \mathbf{e}_u(\tau), \mathbf{e}_v(\tau)) \, \mathrm{d}\tau, \end{aligned}$$

with $\mathbf{h}_f(\tau, \mathbf{e}_u(\tau), \mathbf{e}_v(\tau)) := \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau) + \mathbf{e}_u(\tau), \widetilde{\mathbf{v}}(\tau) + \mathbf{e}_v(\tau)) - \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau))$. This is a Picard-type formulation for the error $\mathbf{e}_u(t)$, and a completely analogous relation involving $\mathbf{h}_g(\tau, \mathbf{e}_u(\tau), \mathbf{e}_v(\tau))$ can be derived for $\mathbf{e}_v(t)$.

A high-order accurate representation of the residual function $\mathbf{r}_u(t)$ in (3.6) is obtained by integrating a smooth interpolant for $\mathbf{f}_j := \mathbf{f}(t_j, \mathbf{u}_j, \mathbf{v}_j)$, i.e.,

$$\mathbf{r}_u(t) \approx M_u \mathbf{u}_0 + \sum_{j=0}^{n} \mathbf{f}_j \left(\int_0^t \ell_j(\tau)\, \mathrm{d}\tau \right) - M_u \widetilde{\mathbf{u}}(t).$$

Denoting by $\mathbf{r}_{u,j}$ the approximations to the residuals $\mathbf{r}_u(t_j)$ at times t_j, we can write

$$[\mathbf{r}_{u,0}, \mathbf{r}_{u,1}, \ldots, \mathbf{r}_{u,n}] = M_u \mathbf{u}_0 [1, 1, \ldots, 1] + [\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_n] C_u - M_u [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_n],$$

where $C_u = [c_{i,j}] \in \mathbb{R}^{(n+1)\times(n+1)}$ is a collocation matrix for cumulative integration, i.e.,

$$c_{i,j} = \int_0^{t_j} \ell_i(\tau)\, \mathrm{d}\tau, \quad i, j = 0, 1, \ldots, n, \tag{3.9}$$

with $c_{0,0}$ being the top-left entry of C_u. An analogous representation for the residual approximations $\mathbf{r}_{v,j}$ can be easily derived. As the integrand in (3.8) can be expected to be smooth, we may apply a low-order quadrature rule to the integral and thereby obtain a time-stepping method for the error $\mathbf{e}_u(t)$ at the time points t_j. The right-endpoint rectangular rule gives rise to an implicit Euler scheme, starting with $\mathbf{e}_{u,0} = \mathbf{0}$,

$$M_u \mathbf{e}_{u,j} = M_u \mathbf{e}_{u,j-1} + (\mathbf{r}_{u,j} - \mathbf{r}_{u,j-1}) + (t_j - t_{j-1}) \cdot \mathbf{h}_f(t_j, \mathbf{e}_{u,j}, \mathbf{e}_{v,j}), \quad j = 1, \ldots, n. \tag{3.10}$$

Analogously, the time-stepping scheme for $\mathbf{e}_v(t)$, with $\mathbf{e}_{v,n} = \mathbf{0}$, reads

$$M_v \mathbf{e}_{v,j-1} = M_v \mathbf{e}_{v,j} - (\mathbf{r}_{v,j} - \mathbf{r}_{v,j-1}) - (t_j - t_{j-1}) \cdot \mathbf{h}_g(t_{j-1}, \mathbf{e}_{u,j-1}, \mathbf{e}_{v,j-1}), \quad j = n, \ldots, 1. \tag{3.11}$$

Finally the current approximations $\widetilde{\mathbf{u}}$ and $\widetilde{\mathbf{v}}$ are updated via $\widetilde{\mathbf{u}}_{\text{new}} = \widetilde{\mathbf{u}} + \mathbf{e}_u$ and $\widetilde{\mathbf{v}}_{\text{new}} = \widetilde{\mathbf{v}} + \mathbf{e}_v$, which concludes one deferred correction sweep.

3.2. Splitting approach. To solve the coupled time-stepping recursions for $\mathbf{e}_u$ and $\mathbf{e}_v$ in (3.10)–(3.11), one way is to "freeze" the component $\widetilde{\mathbf{v}}$, while computing a correction $\mathbf{e}_u$ for $\widetilde{\mathbf{u}}$, and vice versa. In order to give a precise description, let us again denote by $\mathbf{u}_j$ approximations for $\mathbf{u}(t_j)$ on a time grid $t_0, t_1, \ldots, t_n$; analogously we denote by $\mathbf{v}_j$ the approximations for $\mathbf{v}(t_j)$. Here is a complete description of our procedure.

Algorithm 1: Deferred correction for optimal control via splitting.

1. Initialize $\mathbf{u}_j := \mathbf{u}_0$ and $\mathbf{v}_j := \mathbf{v}_T$ for all $j = 0, 1, \ldots, n$.
2. Compute residuals $\mathbf{r}_{u,j}$ $(j = 0, 1, \ldots, n)$ associated with $\widetilde{\mathbf{u}}(t)$ via quadrature of
$$\mathbf{r}_u(t) = \mathbf{u}_0 + \int_0^t \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau))\, \mathrm{d}\tau - \widetilde{\mathbf{u}}(t).$$
3. Compute errors $\mathbf{e}_{u,j}$ via (3.10).
4. Update approximations $\mathbf{u}_j := \mathbf{u}_j + \mathbf{e}_{u,j}$ $(j = 0, 1, \ldots, n)$.
5. Compute residuals $\mathbf{r}_{v,j}$ $(j = 0, 1, \ldots, n)$ associated with $\widetilde{\mathbf{v}}(t)$ via quadrature of
$$\mathbf{r}_v(t) = \mathbf{v}_T + \int_T^t \mathbf{g}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau))\, \mathrm{d}\tau - \widetilde{\mathbf{v}}(t).$$
6. Compute errors $\mathbf{e}_{v,j}$ via (3.11).
7. Update approximations $\mathbf{v}_j := \mathbf{v}_j + \mathbf{e}_{v,j}$ $(j = 0, 1, \ldots, n)$.
8. If error criterion is satisfied, stop. Otherwise go to step 2.

3.3. Coupling approach for linear problems. Consider the coupled system of linear initial/final value problems (2.11)–(2.12). It is possible to write down a global system for the errors in $\widetilde{\mathbf{u}}$ and $\widetilde{\mathbf{v}}$, and to solve for both error components simultaneously. The equations for the errors simplify due to linearity (cf. (3.8)):

$$\begin{cases} M_u \mathbf{e}_u(t) &= \mathbf{r}_u(t) + \displaystyle\int_0^t K_1 \mathbf{e}_u(\tau) - K_2 \mathbf{e}_v(\tau)\,\mathrm{d}\tau, \\ M_v \mathbf{e}_v(t) &= \mathbf{r}_v(t) - \displaystyle\int_T^t K_3 \mathbf{e}_u(\tau) - K_4 \mathbf{e}_v(\tau)\,\mathrm{d}\tau. \end{cases}$$

Discretizing both equations using the implicit Euler method we obtain

$$\begin{aligned} M_u \mathbf{e}_{u,j} &= M_u \mathbf{e}_{u,j-1} + (\mathbf{r}_{u,j} - \mathbf{r}_{u,j-1}) + (t_j - t_{j-1}) \cdot (K_1 \mathbf{e}_{u,j} - K_2 \mathbf{e}_{v,j}), \qquad M_u \mathbf{e}_{u,0} = \mathbf{0}, \\ M_v \mathbf{e}_{v,j-1} &= M_v \mathbf{e}_{v,j} - (\mathbf{r}_{v,j} - \mathbf{r}_{v,j-1}) - (t_j - t_{j-1}) \cdot (K_3 \mathbf{e}_{u,j-1} - K_4 \mathbf{e}_{v,j-1}), \qquad M_v \mathbf{e}_{v,n} = \mathbf{0}. \end{aligned}$$

We can write these relations in form of the following matrix system:

$$\underbrace{\left[\begin{array}{c|c} \mathcal{K}_{3,\tau} & \mathcal{M}_v^\top - \mathcal{K}_{4,\tau} \\ \hline \mathcal{M}_u - \mathcal{K}_{1,\tau} & \mathcal{K}_{2,\tau} \end{array}\right]}_{\boldsymbol{\mathcal{A}}} \left[\begin{array}{c} \mathbf{e}_{u,0} \\ \mathbf{e}_{u,1} \\ \vdots \\ \mathbf{e}_{u,n-1} \\ \mathbf{e}_{u,n} \\ \hline \mathbf{e}_{v,0} \\ \mathbf{e}_{v,1} \\ \vdots \\ \mathbf{e}_{v,n-1} \\ \mathbf{e}_{v,n} \end{array}\right] = \left[\begin{array}{c} \mathbf{r}_{v,0} - \mathbf{r}_{v,1} \\ \mathbf{r}_{v,1} - \mathbf{r}_{v,2} \\ \vdots \\ \mathbf{r}_{v,n-1} - \mathbf{r}_{v,n} \\ \mathbf{0} \\ \hline \mathbf{0} \\ \mathbf{r}_{u,1} - \mathbf{r}_{u,0} \\ \vdots \\ \mathbf{r}_{u,n-1} - \mathbf{r}_{u,n-2} \\ \mathbf{r}_{u,n} - \mathbf{r}_{u,n-1} \end{array}\right]. \tag{3.12}$$

Note that the matrix $\boldsymbol{\mathcal{A}}$ is exactly the same as that in (2.13). This is a major advantage of the deferred correction approach as we can reuse existing solvers and preconditioners for the solution of (3.12). Only the right-hand side changes from one sweep to the next, incorporating the latest residuals $\mathbf{r}_{u,j}$ and $\mathbf{r}_{v,j}$. We obtain the following algorithm.

Algorithm 2: Deferred correction for optimal control via coupling.

1. Get initial approximations $\mathbf{u}_j$ and $\mathbf{v}_j$ $(j = 0, 1, \ldots, n)$ by solving the linear system (2.13).
2. Compute residuals $\mathbf{r}_{u,j}$ $(j = 0, 1, \ldots, n)$ associated with $\widetilde{\mathbf{u}}(t)$ via quadrature of
$$\mathbf{r}_u(t) = \mathbf{u}_0 + \int_0^t K_1 \widetilde{\mathbf{u}}(\tau) - K_2 \widetilde{\mathbf{v}}(\tau) + \widehat{\mathbf{f}}(\tau)\,\mathrm{d}\tau - \widetilde{\mathbf{u}}(t).$$
3. Compute residuals $\mathbf{r}_{v,j}$ $(j = 0, 1, \ldots, n)$ associated with $\widetilde{\mathbf{v}}(t)$ via quadrature of
$$\mathbf{r}_v(t) = \mathbf{v}_T + \int_T^t K_3 \widetilde{\mathbf{u}}(\tau) - K_4 \widetilde{\mathbf{v}}(\tau) + \widehat{\mathbf{g}}(\tau)\,\mathrm{d}\tau - \widetilde{\mathbf{v}}(t).$$
4. Compute errors $\mathbf{e}_{u,j}$ and $\mathbf{e}_{v,j}$ by solving the linear system (3.12).
5. Update approximations $\mathbf{u}_j := \mathbf{u}_j + \mathbf{e}_{u,j}$ and $\mathbf{v}_j := \mathbf{v}_j + \mathbf{e}_{v,j}$ $(j = 0, 1, \ldots, n)$.
6. If error criterion is satisfied, stop. Otherwise go to step 2.

3.4. Rational interpolation scheme. We now draw attention to the choice of Lagrange basis functions ℓ_j ($j = 0, 1, \ldots, n$) in (3.3). Ideally we want these functions to be smooth so that they can be integrated to high accuracy, but at the same time we wish to flexible in the spacing of the time points t_j. In most cases, we would like to collocate our approximations $\mathbf{u}_j$ and $\mathbf{v}_j$ at *equidistant* time points on $[0, T]$. As polynomial interpolation through equidistant data abscissae exhibits exponential instability with increasing degree n, we prefer here to use the rational barycentric interpolants proposed in [10]. More precisely, given a *blending parameter* b such that $0 \le b \le n$, we define as in [10, eq. (11)] the index set

$$J_k = \{0, 1, \ldots, n-b\} \cap \{k-b, k-b+1, \ldots, k\}.$$

and the weights

$$w_k = (-1)^{k-b} \sum_{i \in J_k} \prod_{j=i, j \neq k}^{i+b} \frac{1}{|t_k - t_j|}.$$

One can easily verify that the barycentric formula

$$\ell_j(t) = \frac{w_j}{t - t_j} \Big/ \sum_{k=0}^{n} \frac{w_k}{t - w_k}$$

represents a rational function satisfying $\ell_j(t_i) = \delta_{ij}$ as required, and has no poles on the real axis (hence is smooth). A deferred correction scheme for initial value problems based on these rational interpolants has been previously proposed in [17]. The choice of the parameter b influences the order of smoothness of the interpolation scheme and it has to be balanced with the growing Lebesgue constant; see, e.g., [10, 16, 17]. In our numerical experiments we use $b = \min(n, 10)$, and find this to be a good compromise between high approximation accuracy and low condition number of the interpolation scheme.

4. Convergence analysis. In this section we analyze the splitting and the coupling approach for a *scalar* linear system

$$\begin{cases} u'(t) &= k_1 u(t) - k_2 v(t), & u(0) \in \mathbb{R} \text{ given}, & (4.1) \\ v'(t) &= k_3 u(t) - k_4 v(t), & v(T) \in \mathbb{R} \text{ given}. & (4.2) \end{cases}$$

We will show that both approaches are related to power iteration for a certain eigenvalue problem, and how the spectral properties of the iteration matrices relate to the convergence of each method. For basic convergence results on the power iteration we refer to [34, Section 4.1]. A similar connection to iterative linear algebra methods, in this case the Gauss–Seidel method, has been made in [21] for the single-ODE deferred correction scheme.

4.1. Splitting approach. Assume we are given $\widetilde{\mathbf{U}} = [u_0, u_1, \ldots, u_n]$, a vector which collects the values of a deferred correction approximation $\widetilde{u}(t)$ at time points $t_0, t_1, \ldots, t_n$. Likewise, we define $\widetilde{\mathbf{V}} = [v_0, v_1, \ldots, v_n]$. The residual vector $\mathbf{R}_u = [r_{u,0}, r_{u,1}, \ldots, r_{u,n}]$ associated with $\widetilde{\mathbf{U}}$ is defined as

$$\mathbf{R}_u = [u_0, u_0, \ldots, u_0] + \left[\int_0^{t_j} [\mathbb{I}\,(k_1 \widetilde{\mathbf{U}} - k_2 \widetilde{\mathbf{V}})](\tau)\, \mathrm{d}\tau \right]_{j=0,1,\ldots,n} - \widetilde{\mathbf{U}}, \tag{4.3}$$

where the interpolation operator $\mathbb{I}$ maps a row vector with $n+1$ entries to its interpolating function. The cumulative integration of $\mathbb{I}\,\widetilde{\mathbf{u}}$ is a linear operation and can hence be represented as a matrix product with a collocation matrix $C_u \in \mathbb{R}^{(n+1)\times(n+1)}$ defined in (3.9), i.e.,

$$\mathbf{R}_u = [u_0, u_0, \ldots, u_0] + (k_1 \widetilde{\mathbf{U}} - k_2 \widetilde{\mathbf{V}}) C_u - \widetilde{\mathbf{U}}. \tag{4.4}$$

We now turn our attention to the implicit Euler recursion (3.11), which for the linear test problem simplifies to $e_{u,0} = 0$ and

$$e_{u,j} = e_{u,j-1} + (r_{u,j} - r_{u,j-1}) + \tau_j k_1 e_{u,j}, \quad j = 1, \ldots, n, \tag{4.5}$$

where $\tau_j = t_j - t_{j-1}$. Defining the vector $\mathbf{E}_u = [e_{u,0}, e_{u,1}, \ldots, e_{u,n}]$ and the $(n+1) \times (n+1)$ matrices

$$D_u = \begin{bmatrix} 0 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & \ddots & \\ & & & \ddots & -1 \\ & & & & 1 \end{bmatrix}, \quad E_u = \begin{bmatrix} 1 & -1 & & & & \\ & 1-\tau_1 k_1 & -1 & & & \\ & & 1-\tau_2 k_1 & \ddots & & \\ & & & \ddots & -1 & \\ & & & & 1-\tau_n k_1 \end{bmatrix},$$

we can write the implicit Euler recursion (4.5) in the form

$$\mathbf{E}_u E_u = \mathbf{R}_u D_u.$$

Combining this with formula (4.4) for $\mathbf{R}_u$, and using that $[u_0, u_0, \ldots, u_0]D_u = [0, 0, \ldots, 0]$, we obtain

$$\mathbf{E}_u = \widetilde{\mathbf{U}}(k_1 C_u D_u - D_u)E_u^{-1} - \widetilde{\mathbf{V}} k_2 C_u D_u E_u^{-1}.$$

Finally, in deferred correction the next iterate is obtained via $\widetilde{\mathbf{U}}_{\text{new}} = \widetilde{\mathbf{U}} + \mathbf{E}_u$, and hence

$$\widetilde{\mathbf{U}}_{\text{new}} = \widetilde{\mathbf{U}}(E_u + k_1 C_u D_u - D_u)E_u^{-1} - \widetilde{\mathbf{V}} k_2 C_u D_u E_u^{-1} = \widetilde{\mathbf{U}} M_u + \widetilde{\mathbf{V}} N_u,$$

with $M_u = (E_u + k_1 C_u D_u - D_u)E_u^{-1}$ and $N_u = -k_2 C_u D_u E_u^{-1}$. A similar derivation for $\widetilde{\mathbf{V}}$ yields

$$\widetilde{\mathbf{V}}_{\text{new}} = \widetilde{\mathbf{U}}_{\text{new}} M_v + \widetilde{\mathbf{V}} N_v,$$

where $M_v = k_3 C_v D_v E_v^{-1}$, $N_v = (E_v - k_4 C_v D_v - D_v)E_v^{-1}$, and

$$D_v = \begin{bmatrix} 1 & & & & \\ -1 & \ddots & & & \\ & \ddots & 1 & & \\ & & -1 & 1 & 0 \\ & & & -1 & 0 \end{bmatrix}, \quad E_v = \begin{bmatrix} 1-\tau_1 k_4 & & & & \\ -1 & \ddots & & & \\ & \ddots & 1-\tau_{n-1}k_4 & & \\ & & -1 & 1-\tau_n k_4 & \\ & & & -1 & 1 \end{bmatrix}.$$

Combining these two nested recursions for $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{V}}$ into one gives

$$[\widetilde{\mathbf{U}}_{\text{new}}, \widetilde{\mathbf{V}}_{\text{new}}] = [\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}] \underbrace{\begin{bmatrix} M_u & O \\ N_u & N_v \end{bmatrix} \begin{bmatrix} I & M_v \\ O & I \end{bmatrix}}_{\mathcal{S}}. \tag{4.6}$$

This finally shows that our deferred correction approach with splitting applied to a linear problem is equivalent to a power iteration with the matrix $\mathcal{S}$. Due to the special structure of the matrices M_u and N_v is it easy to see that $\mathcal{S}$ has an eigenvalue equal to 1 of geometric multiplicity at least 2. Together with the conditions for u_0 and v_n the power iteration is then guaranteed to converge to a unique solution (a linear combination of the two eigenvectors associated with the eigenvalue 1) if all other eigenvalues of the iteration matrix have modulus strictly less than 1. The following theorem summarizes our findings.

THEOREM 4.1. *The matrix $\mathcal{S}$ defined in* (4.6) *has an eigenvalue $\lambda = 1$ of geometric multiplicity at least* 2. *The splitting deferred correction method given in Algorithm 1 applied to the linear test problem* (4.1)–(4.2) *is guaranteed to converge to an eigenvector of $\mathcal{S}$ if the eigenvalues of $\mathcal{S}$ sorted by non-increasing modulus satisfy*

$$1 = \lambda_1 = \lambda_2 > |\lambda_3| \geq |\lambda_4| \geq \cdots.$$

We will illustrate this convergence result in section 4.3 and at a heat control problem in section 6.

4.2. Coupling approach. We now analyze the coupling approach for the same linear test problem (4.1)–(4.2). The scalar form of the coupled time-stepping scheme reads

$$\left[\begin{array}{ccc|ccc} \tau_1 k_3 & & & & & \\ & \tau_2 k_3 & & & E_v^\top & \\ & & \ddots & & & \\ & & \tau_n k_3 & & & \\ & & 0 & & & \\ \hline & & & 0 & & \\ & & & & \tau_1 k_2 & \\ & E_u^\top & & & & \ddots \\ & & & & & \tau_{n-1} k_2 \\ & & & & & \tau_n k_2 \end{array}\right] \left[\begin{array}{c} e_{u,0} \\ e_{u,1} \\ \vdots \\ e_{u,n-1} \\ e_{u,n} \\ \hline e_{v,0} \\ e_{v,1} \\ \vdots \\ e_{v,n-1} \\ e_{v,n} \end{array}\right] = \left[\begin{array}{c} r_{v,0} - r_{v,1} \\ r_{v,1} - r_{v,2} \\ \vdots \\ r_{v,n-1} - r_{v,n} \\ 0 \\ \hline 0 \\ r_{u,1} - r_{u,0} \\ \vdots \\ r_{u,n-1} - r_{u,n-2} \\ r_{u,n} - r_{u,n-1} \end{array}\right],$$

or equivalently (written in transposed form),

$$[\mathbf{E}_u, \mathbf{E}_v] \left[\begin{array}{cc} T_u & E_u \\ E_v & T_v \end{array}\right] = [\mathbf{R}_u, \mathbf{R}_v] \left[\begin{array}{cc} & D_u \\ D_v & \end{array}\right], \tag{4.7}$$

where $\mathbf{E}_u$, $\mathbf{E}_v$, E_u, E_v, D_u, D_v are defined as in the previous subsection, $T_u = k_3 \cdot \mathrm{diag}(\tau_1, \tau_2, \ldots, \tau_n, 0)$ and $T_v = k_2 \cdot \mathrm{diag}(0, \tau_1, \ldots, \tau_{n-1}, \tau_n)$. Now, using the linear forms of the residuals (cf. (4.3)) we have

$$\begin{aligned} [\mathbf{R}_u, \mathbf{R}_v] \left[\begin{array}{cc} & D_u \\ D_v & \end{array}\right] &= [\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}] \left[\begin{array}{cc} k_3 C_v D_v & k_1 C_u D_u - D_u \\ -k_4 C_v D_v - D_v & -k_2 C_u D_u \end{array}\right] \\ &= [\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}] \left[\begin{array}{cc} k_1 C_u - I & k_3 C_v \\ -k_2 C_u & -k_4 C_v - I \end{array}\right] \left[\begin{array}{cc} & D_u \\ D_v & \end{array}\right]. \end{aligned}$$

Combining this relation with formula (4.7) for the error $[\mathbf{E}_u, \mathbf{E}_v]$ we arrive at

$$\begin{aligned} [\widetilde{\mathbf{U}}_{\mathrm{new}}, \widetilde{\mathbf{V}}_{\mathrm{new}}] &= [\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}] + [\mathbf{E}_u, \mathbf{E}_v] \\ &= [\widetilde{\mathbf{U}}, \widetilde{\mathbf{V}}] \underbrace{\left(I + \left[\begin{array}{cc} k_1 C_u - I & k_3 C_v \\ -k_2 C_u & -k_4 C_v - I \end{array}\right] \left[\begin{array}{cc} & D_u \\ D_v & \end{array}\right] \left[\begin{array}{cc} T_u & E_u \\ E_v & T_v \end{array}\right]^{-1} \right)}_{\mathcal{C}}. \end{aligned} \tag{4.8}$$

Again, this is a power iteration applied with the matrix $\mathcal{C}$ and the following theorem is immediate.

THEOREM 4.2. *The matrix $\mathcal{C}$ defined in* (4.8) *has an eigenvalue $\lambda = 1$ of geometric multiplicity at least* 2. *The coupling deferred correction method given in Algorithm 2 applied to the linear test problem* (4.1)–(4.2) *is guaranteed to converge to an eigenvector of $\mathcal{C}$ if the eigenvalues sorted by non-increasing modulus satisfy*

$$1 = \lambda_1 = \lambda_2 > |\lambda_3| \geq |\lambda_4| \geq \cdots .$$

4.3. Numerical illustration and the non-scalar case. In Figure 4.1 we illustrate the above results for the scalar test problem

$$\begin{aligned} u' &= -4.935u + 20v, \quad u(0) = 1, \\ v' &= u + 4.935v, \qquad\quad v(1) = 0. \end{aligned}$$

The parameters are chosen to match those of the (non-scalar) heat equation example in section 6.1. The time interval $[0, 1]$ is discretized by equispaced time points $t_j = j/n$, $j = 0, 1, \ldots, n = 10$, using

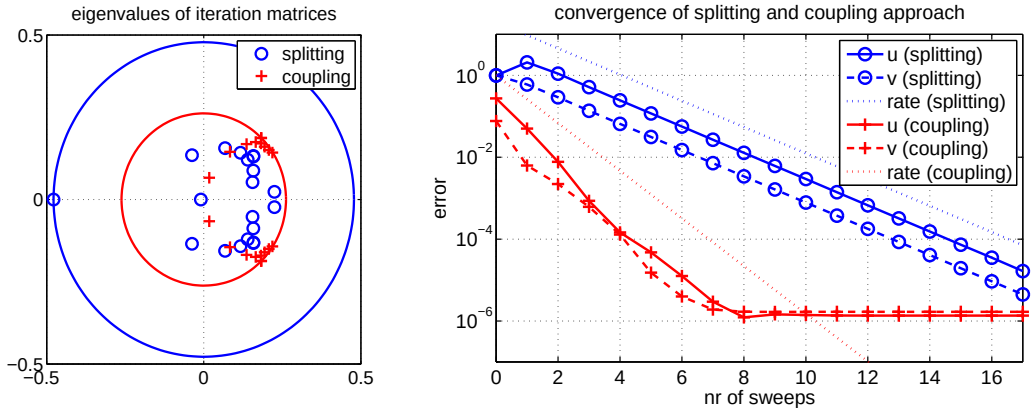

Fig. 4.1: Illustration of our convergence analysis of the splitting and coupling approaches applied to a linear test problem. Left: Eigenvalues of the iteration matrices associated with the splitting (blue circles) and coupling (red stars) approaches. The eigenvalues $\lambda_1 = \lambda_2 = 1$ are outside the visible region. Right: Convergence curves for both components u and v of the solutions after each sweep. Sweep 0 corresponds to the initial guess (which is constant in the splitting approach and obtained from a single implicit Euler run in the coupling approach).

the barycentric rational interpolation scheme with blending parameter $b = 10$ (as $b = n$ this actually amounts to polynomial interpolation; see [10]). On the left of Figure 4.1 we plot the eigenvalues of the splitting and coupling iteration matrices, $\mathcal{S}$ (blue circles) and $\mathcal{C}$ (red pluses), respectively. The enclosing circles are of radius $|\lambda_3|$. On the right of Figure 4.1 we show the error of the approximate solutions obtained via the two deferred correction approaches. We observe geometric convergence at rate $|\lambda_3|$ (indicated by the dotted line), as expected from a power iteration. The convergence factor of the coupling approach, $|\lambda_3| = 0.26$, is significantly better than that of the splitting approach, $|\lambda_3| = 0.47$.

Let us also comment on non-scalar problems, in particular, the heat control problem. The homogeneous form of (2.2)–(2.3) with finite difference or spectral discretization (i.e., $M_u = M_v = I$) is

$$\begin{cases} \mathbf{u}'(t) = L\mathbf{u}(t) + \dfrac{1}{\beta}\mathbf{v}(t), & \mathbf{u}(0) = \mathbf{u}_0 \in \mathbb{R}^N \text{ given}, \\ \mathbf{v}'(t) = \mathbf{u}(t) - L\mathbf{v}(t), & \mathbf{v}(T) = \mathbf{v}_T \in \mathbb{R}^N \text{ given}, \end{cases}$$

with $L \in \mathbb{R}^{N\times N}$ corresponding to a discretization of ∇^2. Assuming that $L = XDX^{-1}$ is diagonalizable with $D = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$, the above system decouples into N scalar equations of the studied form. In section 6.1 we will indeed find that the smallest eigenvalue of the discretization matrix for the diffusion operator ∇^2 dictates the convergence behavior of our deferred correction scheme.

In general it does not seem to be possible to relate the convergence of our deferred correction method to a simple scalar problem, in particular, when dealing with differential-algebraic equations where time derivatives are pre-multiplied by singular matrices M_u, M_v as, for example, in the Stokes problem. A thorough convergence analysis for such problems will be subject of future work.

5. Computational considerations. We discuss a number of computational aspects which play a role when implementing the deferred correction methods presented in this article. In particular, there are three main sources of numerical error in our method:

- error from the spatial discretization,
- error from the time discretization,
- linear algebra error from inner solves of (2.13).

These errors need to be balanced to obtain an efficient method, and such considerations become particularly relevant when solving large-scale and very complex problems. We will discuss this in some detail in the following sections. However, as the main focus of this paper is on the algorithmic presentation and analysis of our method, we keep this discussion brief.

5.1. Discretization in space. The methods proposed in section 3 are presented in a general form, and should be implementable with any method of spatial discretization (for example spectral methods, finite elements, finite differences, meshless methods).

The choice of the "best" method for spatial discretization will depend on the problem being solved, and the domain on which the solution is sought. One of the determining factors is the size of the matrix systems being solved and their conditioning. For example, in many scenarios a spectral method will generate a much more accurate solution than a finite element method with a comparable number of mesh points, however, the conditioning of the discretization matrices is likely to be much worse (to illustrate, for the Laplacian operator the conditioning of a finite element matrix can grow in inverse proportion to the square of the mesh size [9, Chapter 2], whereas the conditioning of a spectral collocation matrix is more severe [33, Chapter 6]). This trade-off between a method's theoretical accuracy and the conditioning of the resulting matrix is an important consideration when solving large-scale PDE-constrained optimization problems.

To illustrate the high solution accuracy that can be achieved with our deferred correction approach, we predominantly use spectral space discretizations for the numerical experiments in section 6. However, we will also demonstrate that a high-order time discretization can significantly reduce the computational cost even with low-order spatial discretizations (such as finite elements) because the number of required time steps is smaller.

5.2. Discretization in time. Along with the spatial discretization, consideration needs to be given to the discretization in time. In this paper we exclusively used an implicit Euler time stepping scheme with constant time steps $\tau_j = t_j - t_{j-1} = \text{constant}$. In principle we could use any quadrature rule (time stepping scheme) to the error equation (3.8), however, we found the right-endpoint rectangular rule (implicit Euler method) effective and particularly easy to implement within the deferred correction framework. In particular, with equidistant time steps the matrices $M_u - \tau_j K_1$ and $M_v - \tau_j K_4$ implicitly appearing in (2.13) will be identical across all time steps. When a direct linear system solver is employed, only a single matrix factorization needs to be computed across all time steps. If the systems are solved iteratively, we can benefit from constant time steps by employing one of the many preconditioners that have been developed for such a situation (see, e.g., [26, 27, 28, 37, 38, 39, 41, 45]).

5.3. Inexact inner solves. Another key aspect of our deferred correction method is that of the required accuracy when solving the matrix system (2.13). Note that this system is solved repeatedly with different right-hand sides at each deferred correction sweep. For early sweeps in particular, where the accuracy in the approximate solution is relatively poor, it seems unnecessary to solve this system to high accuracy. We are therefore presented with the option of constructing preconditioned iterative methods, of the form discussed in [19, 27, 28, 30, 46], for the matrix systems (2.13) (as opposed to applying direct methods). Of course the viability and effectiveness of such an approach will depend on the PDE which we wish to solve, and the resulting complexity of the matrix system. Once a fast and robust iterative method has been constructed for the problem at hand, it can easily be applied within a deferred correction method. A reasonable heuristic would be to solve the matrix system to a tolerance a fraction of the size of the expected update (e.g., if the update is of $\mathcal{O}(10^{-3})$ there is little point solving the system to a much lower error tolerance).

The choice of whether to apply a direct or iterative method is likely to be influenced by the method of spatial discretization. For instance, if a finite element discretization is applied, the matrix (2.13) will be large and sparse, and hence a preconditioned iterative method will be the preferred option (provided a suitable preconditioner can be constructed). By contrast when a spectral space discretization is used, the matrix system will be denser and of lower dimension, and hence a direct method should be used.

In addition, the choice of solver will also depend on whether the coupling or splitting approach is used. For the coupling approach, the systems to be solved are much larger and an iterative solution method may be the only option. For the splitting approach, one is solely required to compute solutions to (smaller) block triangular systems and direct solution methods may be attractive. To summarize, the solver for (2.13) should be tailored to the PDE-constrained optimization problem at hand, the space and time discretization approach being used, and the accuracy requirements.

6. Numerical experiments. The development of test problems for PDE-constrained optimization is a highly non-trivial task. A contribution of the following two subsections is to derive test problems with analytic solutions. In order to evaluate the accuracy of the deferred correction solutions we discretize these problems using spectral methods. The third subsection considers a convection–diffusion problem without a known analytic solution, and the fourth problem is of a larger scale using a finite element discretization and a preconditioned iterative solver.

The MATLAB codes for all tests are available online at `http://guettel.com/dccontrol`.

6.1. 2D heat control problem with spectral space discretization. The components of our first test problem, for the heat equation control problem (2.1), are stated below. It can be shown that this is an exact solution of the continuous optimality conditions (2.2) and (2.3), along with all initial and boundary conditions imposed on y and λ,

$$\begin{aligned}
y &= \left(\frac{4}{d\pi^2\beta}e^T - \frac{4}{(4+d\pi^2)\beta}e^t\right) \prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right), \\
\lambda &= (e^T - e^t) \prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right), \qquad \left[u = \frac{1}{\beta}\lambda\right] \\
y_d &= \left(\left[\frac{d\pi^2}{4} + \frac{4}{d\pi^2\beta}\right]e^T + \left[1 - \frac{d\pi^2}{4} - \frac{4}{(4+d\pi^2)\beta}\right]e^t\right) \prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right), \\
y_0 &= \left(\frac{4}{d\pi^2\beta}e^T - \frac{4}{(4+d\pi^2)\beta}\right) \prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right), \\
h &= 0,
\end{aligned}$$

where $d \in \{2,3\}$ is the dimension of the problem, which is solved on the space domain $[-1,1]^d$ and the time interval $[0,T]$. For the purpose of this test we choose $d = 2$, $T = 1$, and $\beta = 0.05$. We are then able to input the above y_d, y_0, and h into the heat equation control problem (2.1), and compare the solutions obtained with the explicit expressions for y and λ. For the space discretization we use a spectral collocation scheme with 11 Chebyshev points on $[-1,1]$ in each coordinate direction. The time interval $[0,1]$ is discretized by equidistant time points $t_j = j/n$, $j = 0,1,\ldots,n = 10$, using the rational barycentric interpolation scheme with blending parameter $b = 10$.

The numerical results are shown in Figure 6.1. On the left, we plot the relative error in the solutions $(\mathbf{u},\mathbf{v})$ (which are the discretization vectors of (y,λ)) computed by the splitting and coupling approaches after $0,1,\ldots$ sweeps. Here, 0 sweeps corresponds to the error of the initializations of $\mathbf{u}_j$ and $\mathbf{v}_j$ in both cases, i.e., a constant initialization $\mathbf{u}_j = \mathbf{u}_0$ and $\mathbf{v}_j = \mathbf{v}_T$ $(j = 0,1,\ldots,n)$ in the splitting approach, and for the coupling approach the implicit Euler approximations obtained by solving (2.13) once. The relative error of the approximation $\widetilde{\mathbf{u}}(t)$ defined in (3.3) is measured after each sweep as

$$\texttt{relerr} = \frac{\max_{j=0,\ldots,n}\|\mathbf{u}(t_j) - \widetilde{\mathbf{u}}(t_j)\|_\infty}{\max_{j=0,\ldots,n}\|\mathbf{u}(t_j)\|_\infty},$$

where $\mathbf{u}(t)$ is the vector function that corresponds to the evaluation of the analytic solution at the spatial grid points. An analogous error measure is used for $\widetilde{\mathbf{v}}(t)$.

We also show in Figure 6.1 (left) the convergence rate predicted by the scalar analysis in section 4. The parameters are chosen as in section 4.3, with $k_1 = k_4$ corresponding to the eigenvalue of smallest modulus of the matrix $K_1 = K_4$, the discretization matrix of ∇^2, $k_2 = -1/\beta = -20$, and $k_3 = 1$. We find that the convergence behavior is very well described by this scalar approximation. The stagnation of the error curves at level $\approx$ 4e-10 indicates that the spatial discretization is of that accuracy (note that we always compare to the analytic solution).

On the right of Figure 6.1 we show the relative error in $\mathbf{u}$ after $0,1,2,5$ and 10 deferred correction sweeps with the coupling approach, when the number of time steps n is varied. This test verifies

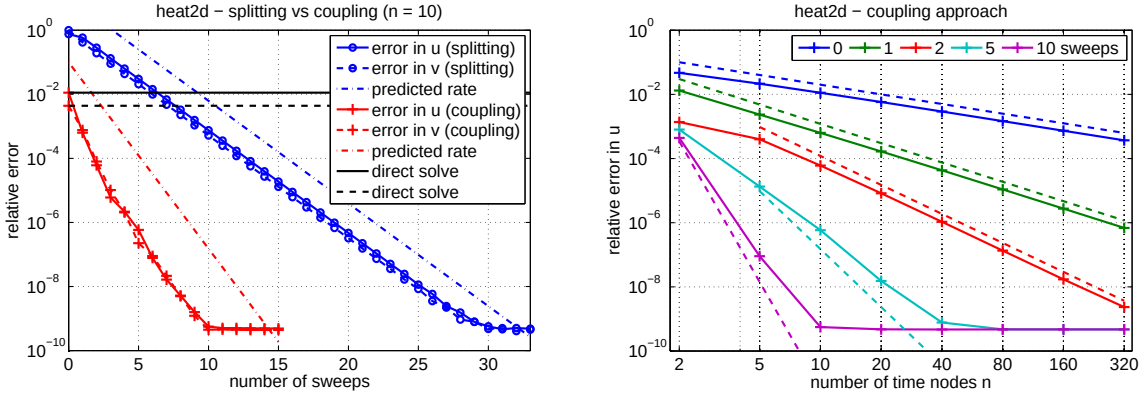

Fig. 6.1: Convergence of deferred correction methods for a 2D heat control problem with analytic solution, discretized by a spectral method. Left: The problem is solved using the splitting and coupling approach with $n = 10$ time steps. Right: Relative error in $\mathbf{u}$ achieved by the coupling approach after $0, 1, 2, 5, 10$ sweeps, respectively, when the number of time steps n is varied. The dashed curves indicate algebraic convergence of order $1, 2, 3, 6, 11$, respectively.

numerically that ℓ sweeps of the coupling deferred correction scheme yield the accuracy of an $(\ell+1)$-th order time-stepping scheme; i.e., $\texttt{relerr} \approx C/n^{\ell+1}$ for some constant C. This is remarkable as these high accuracies are obtained merely by running the first-order implicit Euler scheme $\ell + 1$ times.

Additional information is given in Table 6.1. For example, we can read off this table that with only one deferred correction sweep over $n = 10$ time steps we can achieve a relative error of 6.29e-4 in $\mathbf{u}$, whereas the plain implicit Euler discretization requires $n = 160$ time steps to achieve a comparable accuracy of 7.36e-4. The table gives evidence of how deferred correction can significantly reduce memory requirements and solution time (using MATLAB's backslash in all cases) due to the smaller number n of time steps required for a desired accuracy.

6.2. 2D Stokes control problem with spectral space discretization. Our second test problem relates to the Stokes control problem (2.6). We can verify that the functions and vector fields stated for $\vec{y}$, p, $\vec{\lambda}$, μ satisfy the optimality conditions (2.7)–(2.10) (along with associated initial/boundary conditions) with $\vec{y}_d$, $\vec{z}$, $\vec{y}_0$, $\vec{h}$ as presented:

$$
\begin{aligned}
\vec{y} &= \big[(c_1 + c_2 e^t)\sin^2(\pi x_1)\sin(2\pi x_2),\ \ -(c_1 + c_2 e^t)\sin(2\pi x_1)\sin^2(\pi x_2)\big]^\top, \\
p &= -\pi(c_1 + c_2 e^t)\sin(2\pi x_1)\sin(2\pi x_2) + \text{constant}, \\
\vec{\lambda} &= \big[-(e^T - e^t)\sin^2(\pi x_1)\sin(2\pi x_2),\ (e^T - e^t)\sin(2\pi x_1)\sin^2(\pi x_2)\big]^\top, \qquad \Big[\vec{u} = \frac{1}{\beta}\vec{\lambda}\Big] \\
\mu &= x_1 + x_2 + \text{constant}, \\
\vec{y}_d &= \big[1 + \big(-e^t + (c_1 + c_2 e^t)\big)\sin^2(\pi x_1)\sin(2\pi x_2) + 2\pi^2(e^T - e^t)\big(1 - 4\sin^2(\pi x_1)\big)\sin(2\pi x_2), \\
&\qquad 1 + \big(e^t - (c_1 + c_2 e^t)\big)\sin(2\pi x_1)\sin^2(\pi x_2) + 2\pi^2(e^T - e^t)\sin(2\pi x_1)\big(4\sin^2(\pi x_2) - 1\big)\big]^\top, \\
\vec{z} &= \big[-4\pi^2(c_1 + c_2 e^t)\cos(2\pi x_1)\sin(2\pi x_2),\ 0\big]^\top, \\
\vec{y}_0 &= \big[(c_1 + c_2)\sin^2(\pi x_1)\sin(2\pi x_2),\ \ -(c_1 + c_2)\sin(2\pi x_1)\sin^2(\pi x_2)\big]^\top, \\
\vec{h} &= [0,\ 0]^\top,
\end{aligned}
$$

with $c_1 = -e^T/(4\pi^2\beta)$ and $c_2 = 1/((1 + 4\pi^2)\beta)$. The regularization parameter is chosen as $\beta = 0.01$.

The problem is solved over the time interval $[0, T = 1]$ on the spatial domain $[-1, 1]^2$ discretized using a spectral collocation scheme with 21 Chebyshev points in each coordinate direction. The numerical results are shown in Figure 6.2. The left plot shows the relative error in the solution vectors $(\mathbf{u}, \mathbf{v})$

Table 6.1: Accuracies, memory requirements, and timings for the 2D heat control problem without (top) and with (bottom) deferred correction.

direct solution via MATLAB backslash (i.e., 0 sweeps)

time steps n	`relerr` in $\mathbf{u}$	`relerr` in $\mathbf{v}$	system size	memory (MB)	solution time (ms)
2	4.68e-02	3.04e-02	486	0.1	3
5	2.14e-02	9.90e-03	972	0.3	13
10	1.12e-02	4.38e-03	1782	0.5	26
20	5.75e-03	2.02e-03	3402	1.0	52
40	2.91e-03	9.61e-04	6,642	2.1	108
80	1.47e-03	4.68e-04	13,122	4.2	228
160	7.36e-04	2.31e-04	26,082	8.3	452
320	3.69e-04	1.15e-04	52,002	16.6	900
∞	4.55e-10	4.24e-10	—	—	—

coupling approach with $n = 10$ time steps

sweeps	`relerr` in $\mathbf{u}$	`relerr` in $\mathbf{v}$	system size	memory (MB)	solution time (ms)
0	1.12e-02	4.38e-03	1,782	0.5	26
1	6.29e-04	7.58e-04	1,782	0.5	55
2	6.05e-05	7.94e-05	1,782	0.5	83
3	5.99e-06	1.02e-05	1,782	0.5	110
4	2.16e-06	2.05e-06	1,782	0.5	139
5	5.84e-07	2.24e-07	1,782	0.5	167

after each sweep, with the time interval being discretized by $n = 20$ time steps. When computing these relative errors we have only taken into account the components of $(\mathbf{u}, \mathbf{v})$ which correspond to $(\vec{y}, \vec{\lambda})$. The remaining components correspond to the functions (p, μ) which are only determined up to additive constants. In the right plot of Figure 6.2 we show the relative error in $\mathbf{u}$ after $\ell = 0, 1, 2, 5$ and 10 deferred correction sweeps with the coupling approach, when the number of time steps n is varied. We find that the accuracy monotonically improves as n is increased and more sweeps are being performed. In contrast to the heat equation example, however, the accuracy does not improve algebraically with $(\ell + 1)$-th order in n.

Table 6.2 gives some additional information about the coupling deferred correction method compared to the plain implicit Euler solution. Similarly to the heat equation example we find that only one deferred correction sweep improves the solution accuracy significantly without the need of an increased number of time steps n. The benefit is a much reduced system size, memory requirement, and solution time (again using MATLAB's backslash in all cases).

6.3. A convection–diffusion problem. We now consider a 2D control problem with

$$\mathcal{D} = \frac{\partial}{\partial t} - \nu\nabla^2 + \mathbf{w} \cdot \nabla.$$

Although we do not have an analytic solution for this problem, we can use as a measure of convergence the contraction of the residuals $\mathbf{r}_u$ and $\mathbf{r}_v$ defined in (3.6)–(3.7). On the top of Figure 6.3 we show plots of the residual norm for each of 20 deferred correction sweeps. Below we show plots for the state variable y and the adjoint variable λ. We use a 20-point spectral method in two spatial dimensions, the time domain is chosen as $[0, T = 1]$, and we select $\nu = 0.01$ and $y_d = te^{x_1+x_2}\sin(\pi x_1)\sin(\pi x_2)$. The two experiments shown in Figure 6.3 involve different values of the regularization parameter β and wind vector $\mathbf{w}$.

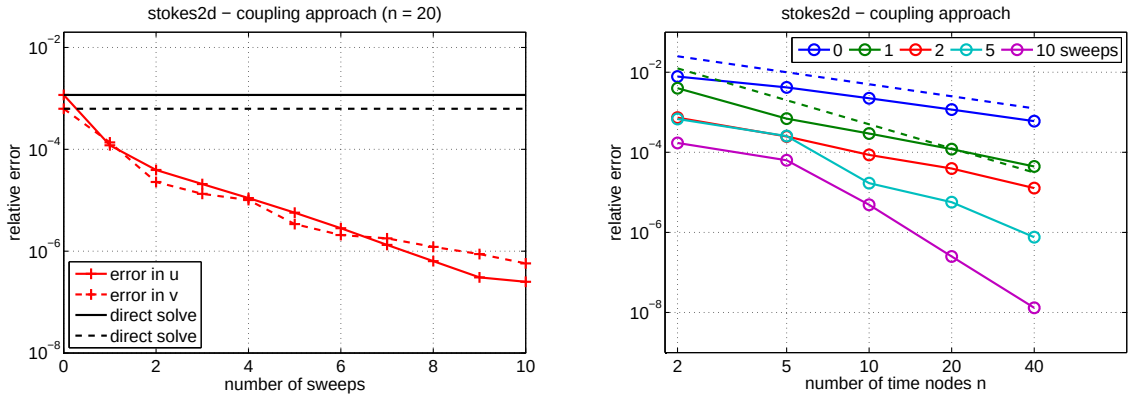

Fig. 6.2: Convergence of the coupling deferred correction method for a 2D Stokes control problem with analytic solution, discretized by a spectral method. Left: Error dependent on the number of sweeps with a fixed time discretization ($n = 20$). Right: Convergence of the $\mathbf{u}$ *component with a fixed number of sweeps over a varying number of time steps n.*

Table 6.2: Accuracies, memory requirements, and timings for the 2D Stokes control problem without (top) and with (bottom) deferred correction.

direct solution via MATLAB backslash (i.e., 0 sweeps)

time steps n	`relerr` in $\mathbf{u}$	`relerr` in $\mathbf{v}$	system size	memory (MB)	solution time (s)
2	7.78e-03	5.08e-03	7,932	5.0	0.7
5	4.17e-03	2.31e-03	15,864	12.3	2.5
10	2.24e-03	1.21e-03	29,084	24.5	4.9
20	1.17e-03	6.26e-04	55,524	48.9	9.9
40	6.01e-04	3.20e-04	108,404	97.6	54.4

coupling approach with $n = 20$ time steps

sweeps	`relerr` in $\mathbf{u}$	`relerr` in $\mathbf{v}$	system size	memory (MB)	solution time (s)
0	1.17e-03	6.26e-04	55,524	48.9	9.9
1	1.20e-04	1.37e-04	55,524	48.9	19.8
2	3.93e-05	2.28e-05	55,524	48.9	29.8
5	5.70e-06	3.40e-06	55,524	48.9	59.6
10	2.50e-07	5.75e-07	55,524	48.9	108.4

6.4. 2D heat control problem with finite elements. So far we have considered numerical experiments involving spectral discretization in space. A more widely used approach, however, is to apply a finite element discretization in space, and this experiment tests our coupling deferred correction method in this context. Let us consider the matrix system (2.13) for the heat equation control problem

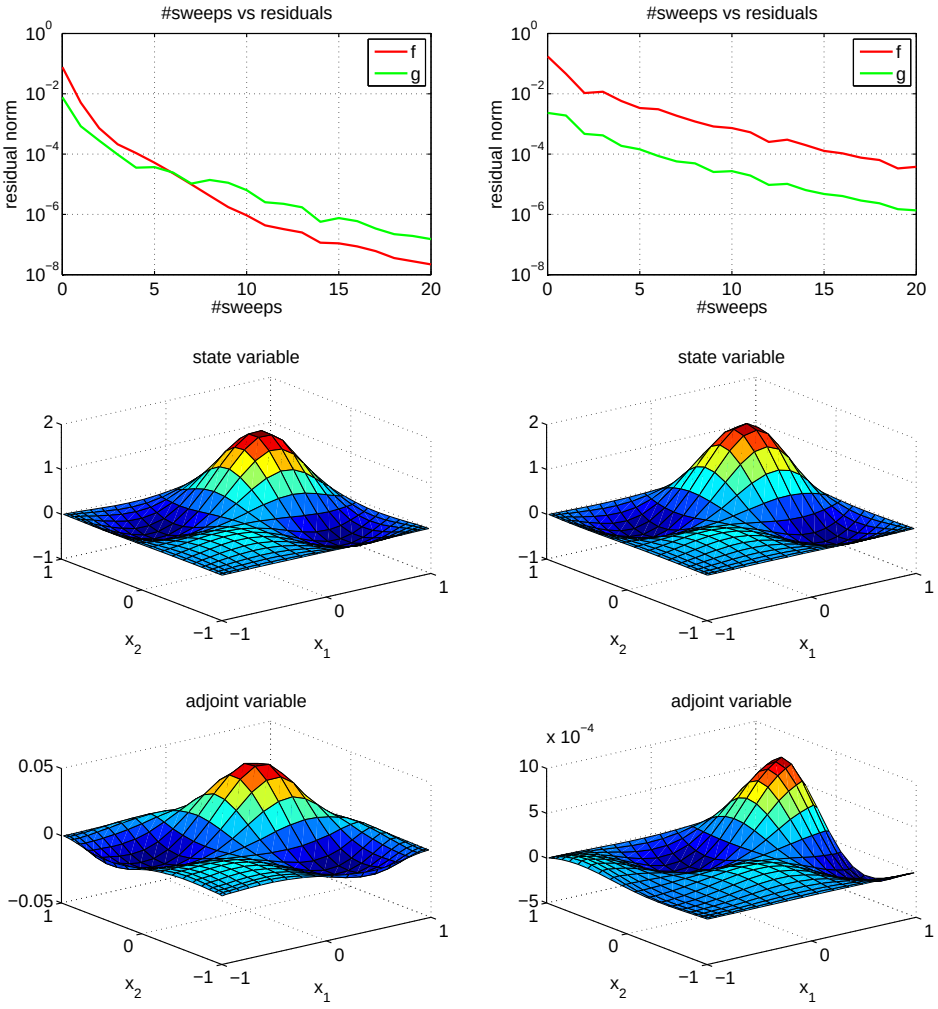

Fig. 6.3: Top: Residual norms after each deferred correction sweep for a 2D convection–diffusion control problem, discretized by a spectral method. Middle: Plots of the state $y(t)$ at time $t = 0.5$. Bottom: Plots of the adjoint $\lambda(t)$ at time $t = 0.5$. For each experiment, $y_d = te^{x_1+x_2}\sin(\pi x_1)\sin(\pi x_2)$, $\nu = 0.01$, $T = 1$, and zero initial and boundary conditions are taken. For the left plots, the wind vector is $\mathbf{w} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^\top$ and the regularization parameter is $\beta = 10^{-2}$; for the right plots, $\mathbf{w} = [-1, 0]^\top$ and $\beta = 10^{-4}$.

(2.2)–(2.3) using a finite element discretization. The relevant blocks of $\boldsymbol{\mathcal{A}}$ are given as follows:

$$\mathcal{M}_u = \mathcal{M}_v = \begin{bmatrix} M & & & & \\ -M & M & & & \\ & \ddots & \ddots & & \\ & & -M & M & \\ & & & -M & M \end{bmatrix},$$

$$\mathcal{K}_{1,\tau} = \text{blkdiag}\big(O, -\tau K, ..., -\tau K, -\tau K\big), \qquad \mathcal{K}_{2,\tau} = \text{blkdiag}\left(O, -\frac{\tau}{\beta}M, ..., -\frac{\tau}{\beta}M, -\frac{\tau}{\beta}M\right),$$

$$\mathcal{K}_{3,\tau} = \text{blkdiag}\big(\tau M, \tau M, ..., \tau M, O\big), \qquad \mathcal{K}_{4,\tau} = \text{blkdiag}\big(-\tau K, -\tau K, ..., -\tau K, O\big),$$

where M and K denote finite element *mass* and *stiffness matrices*, respectively, and τ is the constant time step. The right-hand side incorporates boundary/initial conditions and the desired state y_d [28, 40].

The matrix system (2.13) is of extremely large dimension, even for very coarse finite element discretizations in space. In order to solve this system efficiently, it is necessary to employ preconditioned iterative methods. Fortunately, because the matrix (2.13) is similar to that used for previously developed solvers which do not incorporate deferred correction, we can apply existing preconditioning techniques. In this case of heat equation control, we follow a similar strategy as in [28, 29] by constructing a *saddle point preconditioner*

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} \widehat{\mathcal{K}}_{3,\tau} & 0 \\ 0 & \widehat{S} \end{bmatrix},$$

where $\widehat{\mathcal{K}}_{3,\tau}$ and $\widehat{S}$ denote (invertible) approximations of $\mathcal{K}_{3,\tau}$ and the *approximate Schur complement*[1]

$$\widetilde{S} := -\mathcal{K}_{2,\tau} + \big(\mathcal{M}_u - \mathcal{K}_{1,\tau}\big)\widehat{\mathcal{K}}_{3,\tau}^{-1}\big(\mathcal{M}_v^\top - \mathcal{K}_{4,\tau}\big).$$

To construct an invertible matrix $\widehat{\mathcal{K}}_{3,\tau}$, we 'perturb' $\mathcal{K}_{3,\tau}$ and take $\widehat{\mathcal{K}}_{3,\tau} := \mathrm{blkdiag}(\tau M, \tau M, ..., \tau M, \gamma\tau M)$, for a small parameter γ of our choosing (we use $\gamma = 10^{-6}$ for our experiments). We now approximate $\widetilde{S}$ using a 'matching strategy' (as in [28]) to arrive at

$$\widehat{S} := \big(\mathcal{M}_u - \mathcal{K}_{1,\tau} + \widehat{\mathcal{M}}\big)\widehat{\mathcal{K}}_{3,\tau}^{-1}\big(\mathcal{M}_v^\top - \mathcal{K}_{4,\tau} + \widehat{\mathcal{M}}\big),$$

where $\widehat{\mathcal{M}}$ is chosen so that additional (outer) term $\widehat{\mathcal{M}}\widehat{\mathcal{K}}_{3,\tau}^{-1}\widehat{\mathcal{M}}$ 'matches' the term $-\mathcal{K}_{2,\tau}$ in $\widetilde{S}$. This is achieved by selecting $\widehat{\mathcal{M}} := \mathrm{blkdiag}\left(O, \frac{\tau}{\sqrt{\beta}}M, ..., \frac{\tau}{\sqrt{\beta}}M, \tau\sqrt{\frac{\gamma}{\beta}}M\right)$.

For our experiments, we consider the heat control test problem from section 6.1, but now on an L-shaped domain and for the time interval $[0, T = 5]$. We discretize in space using piecewise quadratic ($Q2$) finite elements. We solve the relevant matrix system with a preconditioned GMRES [35] method using the Incompressible Flow and Iterative Solver Software (IFISS) [8, 36]. Within the preconditioner we apply Chebyshev semi-iteration [12, 13] to approximately invert mass matrices, and the Aggregation-Based Algebraic Multigrid (AGMG) software [22, 23, 24, 25] for sums of stiffness and mass matrices. (For other PDE-constrained optimization problems one will need to incorporate different preconditioners.)

Figure 6.4 shows the convergence of our coupling deferred correction method when the regularization parameter is chosen as $\beta = 10^{-2}$, along with the solution obtained for the state variable. Table 6.3 compares the achievable accuracy without and with deferred correction sweeps for a range of mesh parameters h. We observe in the upper table that the accuracy is limited by the time discretization, as a decrease in h does not result in a more accurate solution. In such cases, our deferred correction approach is very attractive. For example, when $h = 2e-2$ and $n = 20$ time steps are taken, then only 4 deferred correction sweeps can improve the accuracy from 1.43e-2 to 2.67e-4. Extrapolating the accuracies in Table 6.3, we estimate that plain implicit Euler would require approximately $n = 950$ time steps to achieve a similar accuracy. This would be prohibitive.

We further observe in Table 6.3 that increasing the number of time steps n decreases the number of deferred correction iterations required until stagnation. We consider the method to have stagnated if the accuracy of the solution $\mathbf{u}$ does not exceed by a factor of 1.1 the best accuracy achieved within the first 60 sweeps. It is natural that an increase in n will make each deferred correction sweep become "more accurate" and thereby reduce the number of sweeps required until stagnation. On the other hand, the dimension of the matrix systems to be solved increases linearly with n and it is advisable to choose n merely depending on accuracy requirements. Decreasing the mesh parameter h also results in larger linear systems, but is only worthwhile when deferred correction is used to improve the accuracy in time.

Table 6.4 shows the average number of GMRES iterations required to solve the matrix systems with $\boldsymbol{\mathcal{A}}$ to a relative residual tolerance of 10^{-6}, averaged over the first ten deferred correction sweeps,

[1]The exact Schur complement of $\boldsymbol{\mathcal{A}}$ does not exist in this case, due to the non-invertibility of $\mathcal{K}_{3,\tau}$; we therefore instead consider $\widetilde{S}$ by incorporating our approximation $\widehat{\mathcal{K}}_{3,\tau}$ of $\mathcal{K}_{3,\tau}$. See [3] for a comprehensive review of numerical methods for saddle point systems.

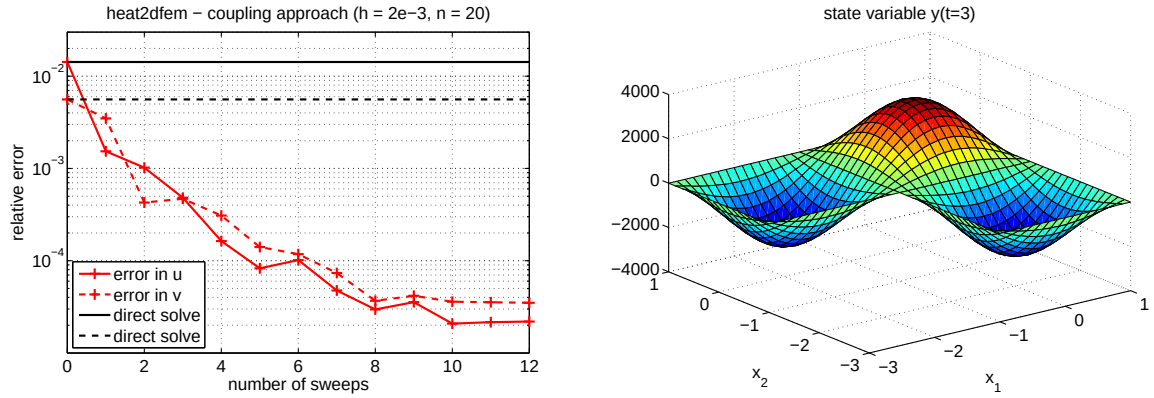

Fig. 6.4: Convergence of deferred correction method for a 2D heat control problem with analytic solution, discretized using Q2 finite elements on an L-shaped domain. Left: The problem is solved using the coupling approach with $n = 20$ time steps. Right: A plot of the solution $y(t)$ at time $t = 3$. The finite element mesh with parameter $h = 2^{-3}$ is also shown.

Table 6.3: Achievable accuracies for the 2D heat equation with (top) and without (bottom) deferred correction. The problem is posed on an L-shaped domain discretized with finite elements depending on the mesh parameter h and the number of time steps n.

implicit Euler accuracy for **u** (i.e., 0 sweeps)

grid parameter h	$n = 10$	$n = 15$	$n = 20$	$n = 25$
2e-2	2.9123e-02	1.9340e-02	1.4285e-02	1.1245e-02
2e-3	2.9074e-02	1.9293e-02	1.4241e-02	1.1203e-02
2e-4	2.9071e-02	1.9290e-02	1.4238e-02	1.1201e-02
2e-5	2.9071e-02	1.9290e-02	1.4238e-02	1.1200e-02

stagnation accuracy for **u** with the coupling approach (number of sweeps in brackets)

grid parameter h	$n = 10$	$n = 15$	$n = 20$	$n = 25$
2e-2	3.1194e-04 (10)	2.7623e-04 (5)	2.6719e-04 (4)	2.8705e-04 (3)
2e-3	1.9819e-05 (31)	2.2406e-05 (16)	2.0833e-05 (10)	2.1649e-05 (7)
2e-4	1.4231e-06 (45)	1.5591e-06 (26)	1.4830e-06 (18)	1.3932e-06 (13)
2e-5	1.4619e-07 (51)	1.0604e-07 (38)	9.3860e-08 (26)	9.1240e-08 (19)

for a range of h and β values. The robustness of the linear solver matches the observations in [28, 29], and we highlight that the computational cost per iteration scales linearly with the system size, i.e., we have employed an *optimal solver*.

The performance of the deferred correction scheme, as well as that of the inner iterative solver, leads us to conclude that the method presented here can readily be applied within existing finite element schemes for PDE-constrained optimization problems with minor code changes. The resulting combination has the proven potential to give highly accurate solutions within a small number of deferred correction iterations and with low memory consumption.

7. Concluding remarks. We have presented a rational deferred correction framework for solving time-dependent PDE-constrained optimization problems. This framework enabled us to solve a range of such problems to much higher accuracy than conventional discretize-in-time-and-solve schemes. The reduced number of required time steps resulted in much smaller matrix systems to be solved, and fewer solution vectors to be stored. Our deferred correction approach is beneficial in particular when the

Table 6.4: Matrix system size for a 2D heat equation example on an L-shaped domain, for different values of the grid parameter h, and the average number of GMRES iterations required to reduce the residual norm by a factor of 10^{-6}.

		coupling approach with $n = 20$ time steps								
		regularization parameter β								
grid parameter h	system size	10^2	10	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
2e-2	9,450	9.9	10.4	12.8	15.1	14.6	14.3	13.9	12.0	10.1
2e-3	34,986	9.3	11.3	13.7	15.8	15.6	16.4	12.6	11.4	12.3
2e-4	134,442	9.1	11.1	13.7	16.0	15.0	15.8	12.8	11.9	11.9
2e-5	526,890	9.2	11.0	13.7	15.6	15.8	14.9	12.8	10.9	11.6
2e-6	2,085,930	9.5	11.2	14.4	16.4	14.6	14.9	12.8	10.9	11.0

desired accuracy is limited by the time discretization. An important feature of our approach is that it can be implemented with minimal effort alongside existing linear system solvers and preconditioners.

We believe there is much future research which may be spawned by this work. There would be great value in applying a similar approach to time-dependent problems governed by nonlinear PDEs, as well as optimization problems with additional box constraints on the state and/or control variables. While the splitting approach presented here has been found to be relatively inefficient for linear problems (compared to the coupling approach), it may well be attractive in the nonlinear case due to its simplicity. A detailed convergence analysis of deferred correction methods for coupled problems, beyond the scalar analysis we have offered, will also be a subject of future research.

Acknowledgements. JWP gratefully acknowledges support from the Engineering and Physical Sciences Research Council (EPSRC) Fellowship EP/M018857/1.

REFERENCES

[1] S. R. Arridge. Optical tomography in medical imaging. *Inverse Problems*, 15:R41–R93, 1999.
[2] W. Barthel, C. John, and F. Tröltzsch. Optimal boundary control of a system of reaction diffusion equations. *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)*, 90:966–982, 2000.
[3] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
[4] A. Borzi and V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations*. SIAM, 2012.
[5] I. Bouchouev and V. Isakov. Uniqueness, stability and numerical methods for the inverse problem that arises in financial markets. *Inverse Problems*, 15:R95–R116, 1999.
[6] M. Cheney, D. Isaacson, and J. C. Newell. Electrical impedance tomography. *SIAM Review*, 41:85–101, 1999.
[7] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numerical Mathematics*, 40(2):241–266, 2000.
[8] H. C. Elman, A. Ramage, and D. J. Silvester. IFISS: a computational laboratory for investigating incompressible flow problems. *SIAM Review*, 56:261–273, 2014.
[9] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press, 2005.
[10] M. S. Floater and K. Hormann. Barycentric rational interpolation with no poles and high rates of approximation. *Numerische Mathematik*, 107:315–331, 2007.
[11] M. R. Garvie, P. K. Maini, and C. Trenchea. An efficient and robust numerical algorithm for estimating parameters in Turing systems. *Journal of Computational Physics*, 229:7058–7071, 2010.
[12] G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods, Part I. *Numerische Mathematik*, 3:147–156, 1961.
[13] G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods, Part II. *Numerische Mathematik*, 3:157–168, 1961.
[14] R. Griesse and S. Volkwein. A primal-dual active set strategy for optimal boundary control of a nonlinear reaction-diffusion system. *SIAM Journal on Control and Optimization*, 44:467–494, 2005.
[15] M. Gunzberger. *Perspectives in Flow Control and Optimization*. SIAM, 2010.
[16] S. Güttel and G. Klein. Convergence of linear barycentric rational interpolation for analytic functions. *SIAM Journal on Numerical Analysis*, 50(5):2560–2580, 2012.

[17] S. Güttel and G. Klein. Efficient high-order rational integration and deferred correction with equispaced data. *Electronic Transactions on Numerical Analysis*, 41:443–464, 2014.
[18] K. Ito and K. Kunisch. *Lagrange Multiplier Approach to Variational Problems and Applications*. Vol. 15 of Advances in Design and Control, SIAM, 2008.
[19] W. Krendl, V. Simoncini, and W. Zulehner. Stability estimates and structural spectral properties of saddle point problems. *Numerische Mathematik*, 124(1):183–213, 2013.
[20] M. Minion. A hybrid parareal spectral deferred corrections method. *Communications in Applied Mathematics and Computational Science*, 5(2):265–301, 2011.
[21] M. L. Minion, R. Speck, M. Bolten, M. Emmett, and D. Ruprecht. Interweaving PFASST and parallel multigrid. *SIAM Journal on Scientific Computing*, 37:S244–S264, 2015.
[22] A. Napov and Y. Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM Journal on Scientific Computing*, 34:A1079–A1109, 2012.
[23] Y. Notay. AGMG software and documentation; see `http://homepages.ulb.ac.be/~ynotay/AGMG`.
[24] Y. Notay. An aggregation-based algebraic multigrid method. *Electronic Transactions on Numerical Analysis*, 37:123–146, 2010.
[25] Y. Notay. Aggregation-based algebraic multigrid for convection-diffusion equations. *SIAM Journal on Scientific Computing*, 34:A2288–A2316, 2012.
[26] J. W. Pearson. Fast iterative solvers for large matrix systems arising from time-dependent Stokes control problems. *submitted to Applied Numerical Mathematics*, 2015.
[27] J. W. Pearson and M. Stoll. Fast iterative solution of reaction-diffusion control problems arising from chemical processes. *SIAM Journal on Scientific Computing*, 35:B987–B1009, 2013.
[28] J. W. Pearson, M. Stoll, and A. J. Wathen. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1126–1152, 2012.
[29] J. W. Pearson, M. Stoll, and A. J. Wathen. Robust iterative solution of a class of time-dependent optimal control problems. *PAMM*, 12(1):1–4, 2012.
[30] J. W. Pearson and A. J. Wathen. A new approximation of the Schur complement in preconditioners for PDE-constrained optimization. *Numerical Linear Algebra with Applications*, 19(5):816–829, 2012.
[31] V. Pereyra. On improving an approximate solution of a functional equation by deferred corrections. *Numerische Mathematik*, 8:376–391, 1966.
[32] V. Pereyra. Iterated deferred correction for nonlinear boundary value problems. *Numerische Mathematik*, 11:111–125, 1968.
[33] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 1994.
[34] Y. Saad. *Numerical Methods for Large Eigenvalue Problems. Revised Edition*. SIAM, 2011.
[35] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 7:856–869, 1986.
[36] D. J. Silvester, H. C. Elman, and A. Ramage. Incompressible Flow and Iterative Solver Software (IFISS), Version 3.3, `http://www.manchester.ac.uk/ifiss`. 2014.
[37] M. Stoll. One-shot solution of a time-dependent time-periodic PDE-constrained optimization problem. *IMA Journal of Numerical Analysis*, 34(4):1554–1577, 2014.
[38] M. Stoll and T. Breiten. A low-rank in time approach to PDE-constrained optimization. *SIAM Journal on Scientific Computing*, 37(1):B1–B29, 2015.
[39] M. Stoll, J. W. Pearson, and P. K. Maini. Fast solvers for optimal control problems from pattern formation. *Journal of Computational Physics*, 304:27–45, 2016.
[40] M. Stoll and A. Wathen. All-at-once solution of time-dependent PDE-constrained optimization problems. *Oxford Centre for Collaborative Applied Mathematics Technical Report 10/47*, 2010.
[41] M. Stoll and A. Wathen. All-at-once solution of time-dependent Stokes control. *Journal of Computational Physics*, 232(1):498–515, 2013.
[42] F. Tröltzsch. *Optimal Control of Partial Differential Equations – Theory, Methods and Applications*. Graduate Studies in Mathematics, Vol. 112. American Mathematical Society, 2010.
[43] P. E. Zadunaisky. A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations, The theory of orbits in the solar system and in stellar systems (G. Contopoulos, ed.), Proceedings of the Symposium of the International Astronomical Union, no. 25, Academic Press, London, 1966.
[44] P. E. Zadunaisky. On the estimation of errors propagated in the numerical integration of ordinary differential equations. *Numerische Mathematik*, 27:21–39, 1976.
[45] M. Zeng and H. Zhang. A new preconditioning strategy for solving a class of time-dependent PDE-constrained optimization problems. *Journal of Computational Mathematics*, 32:215–232, 2014.
[46] W. Zulehner. Nonstandard norms and robust estimates for saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 32(2):526–560, 2011.