A spectral-in-time Newton-Krylov method for nonlinear PDE-constrained optimization

Güttel, Stefan and Pearson, John W.

2020

ISSN 1749-9097

A SPECTRAL-IN-TIME NEWTON–KRYLOV METHOD FOR NONLINEAR PDE-CONSTRAINED OPTIMIZATION

STEFAN GÜTTEL* AND JOHN W. PEARSON†

Abstract. We devise a method for nonlinear time-dependent PDE-constrained optimization problems that uses a spectral-in-time representation of the residual, combined with a Newton–Krylov method to drive the residual to zero. We also propose a preconditioner to accelerate this scheme. Numerical results indicate that this method can achieve fast and accurate solution of nonlinear problems for a range of mesh sizes and problem parameters, the numbers of outer Newton and inner Krylov iterations required to reach the attainable accuracy of a spatial discretization are robust with respect to the number of collocation points in time, and also do not change substantially when other problem parameters are varied.

Key words. PDE-constrained optimization, time-dependent PDE, residual-based method, Newton–Krylov method, coupled systems of PDEs, preconditioner

AMS subject classifications. 93C20, 65N22, 49M15, 65F08

1. Introduction.

PDE-constrained optimization is an important class of mathematical problems [19, 23, 30], with a wide range of applications across science and engineering (see [1, 5, 11, 12], for instance). The fast and accurate solution of the first-order optimality conditions resulting from such problems is a significant challenge for researchers in these communities. For example, when an 'all-at-once' approach is applied to solve such conditions, one is faced with coupled linear systems of huge scale, particularly when standard finite difference or finite element schemes are used for the discretization procedure in the spatial coordinates. For time-dependent problems, there is an additional challenge of how to appropriately discretize in the time variable. When fast solvers for the resulting linear system are required, in particular preconditioned iterative methods, it has been popular to utilize a backward Euler scheme for this purpose (see for example [26, 27, 29]); however this requires a very small step size in order to obtain an accurate numerical solution, and thus very large linear systems to solve.

Motivated by this, in [14] the authors devised deferred correction methods for linear PDE-constrained optimization problems, where equations for the errors and residuals at each deferred correction step were constructed in order to successively increase the order of the time-stepping scheme. As a result one is required to solve a sequence of much smaller linear systems in order to achieve the same accuracy of the numerical solution, compared to a method without a deferred correction acceleration. This approach was found to be highly effective for linear PDE-constrained optimization problems, however a significant question remained how to devise a related strategy for nonlinear PDE-constrained optimization. These problems possess a vastly increased level of difficulty, compared to linear problems, as the matrices describing the spatial behavior of the physical system are different at every time step.

In this paper we devise a residual-based approach for nonlinear PDE-constrained optimization problems; in particular this is based on using a spectral-in-time representation of the residual which is then linearized and solved by a Newton method. The use of a spectral method in the time variable means that high accuracy solutions can be obtained with only a small number of time steps, keeping the linearized Newton system of relatively small dimension. We implement our approach using a Newton–Krylov method, of the form described in [21, Chapter 3] and [22]. To make such a method numerically viable we suggest a general preconditioning strategy, which is found to substantially accelerate the Newton–Krylov scheme for the problems examined. Care has to be taken with the implementation of the preconditioner as the spectral time integration matrix is dense. We hence transform the Jacobian arising in the Newton system into a form that can be easily approximated by a Kronecker product, allowing for the application of a preconditioner whose cost is approximately proportional to the number of spatial degrees of freedom. Our approach mitigates a key difficulty encountered with alternative discretization techniques for nonlinear PDE-constrained optimization problems, specifically being required to solve

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

†School of Mathematics, The University of Edinburgh, James Clerk Maxwell Building, The King's Buildings, Edinburgh, EH9 3FD, United Kingdom, j.pearson@ed.ac.uk

linear systems arising from very large numbers of grid points in time to obtain even modest discretization error properties, and our new method is found to be an effective strategy for a number of examples, mesh sizes, and problem parameters.

We highlight that there has been a substantial amount of research undertaken with the goal of achieving numerical solutions to time-dependent PDE-constrained optimization problems, with low discretization error and within reasonable computation time. Among many valuable references, we refer to [6, 29] for low-rank solution methods, to [26, 27] for preconditioned iterative solvers, to [3, 16] for a discussion of multigrid approaches, to [15] for a multiple shooting strategy, to [24, 25] for parareal approaches, and to [10] for a recently-developed time-parallel method with the computation of the adjoint gradient information performed using the PFASST framework. We also point to the body of work on Krylov deferred correction methods for initial value problems developed in [18, 20]. Indeed, for linear initial value problems, deferred correction can be interpreted as a preconditioned Newton-like method for solving the time collocation system [17, 31].

While the use of a spectral-in-time residual function in this work is inspired by the spectral deferred correction approach in [8], we found in extensive numerical tests that the direct Newton-based minimization of this function without the outer deferred correction loop is both conceptually simpler and indeed more efficient for nonlinear problems. This is the approach we would like to explore in this paper. Compared to our previous work [14] on linear PDE-constrained optimization, the method described here is not a deferred correction scheme. Our aim is to provide a general approach for nonlinear time-dependent problems which, as we will demonstrate, can lead to very accurate solutions while being applicable with any discretization in the spatial variables of the user's choice, and which has the potential to be combined with a number of the approaches listed above. Our focus is on problems with a squared L^2-norm regularization term applied to the control variable in the objective function, along with first-order derivatives in time and a linear term involving the control within the PDE constraint, as such a formulation allows us to eliminate the control variable a-priori and solve a coupled state–adjoint system. We believe that, with suitable modifications which we will describe, our method is more generalizable still.

This paper is structured as follows. In Section 2 we present the residual-based method for nonlinear PDE-constrained optimization. Specifically, in Section 2.1 we state the coupled systems of PDEs arising from the PDE-constrained optimization problems considered in this paper, in Section 2.2 we derive the residual-in-time representation and introduce the Newton–Krylov method we apply, then in Section 2.3 we present the preconditioner which is embedded within the Newton–Krylov scheme. Sections 2.4 and 2.5 further discuss the implementation and the stopping condition used. Section 3 presents the results of a range of numerical experiments, using test problems for which analytic solutions are given in Appendix A. In Section 4 we provide our concluding remarks.

2. Residual-based Newton–Krylov method.

In this section we present our residual-based method for nonlinear time-dependent PDE-constrained optimization problems. To focus our discussion, we will present the methodology using two specific examples, although this approach is much more general. Firstly, motivated by research in literature such as [4, 13], we consider the optimal control of the Schlögl equation given by:

$$\begin{aligned} \min_{y,c} \quad & \frac{1}{2}\int_0^T \int_\Omega \left(y-\widehat{y}\right)^2 \, \mathrm{d}\Omega \, \mathrm{d}t + \frac{\beta}{2}\int_0^T \int_\Omega c^2 \, \mathrm{d}\Omega \, \mathrm{d}t \\ \text{s. t.} \quad & \frac{\partial y}{\partial t} - \nabla^2 y - y + y^3 = c + f \qquad \text{in } \Omega \times (0,T), \\ & \qquad y(\vec{x},t) = h(\vec{x},t) \qquad \text{on } \partial\Omega \times (0,T), \\ & \qquad y(\vec{x},0) = y_0(\vec{x}) \qquad \text{at } t = 0. \end{aligned} \tag{2.1}$$

Here, y, $\widehat{y}$, and c denote the *state*, *desired state*, and *control variables*, respectively, f and h are given functions in space and time, y_0 is a given initial condition, $\beta > 0$ denotes a *regularization parameter* (or *Tikhonov parameter*), and $T > 0$ is the final time. The PDE constraint is posed on a space–time domain, with $(\vec{x}, t) \in \Omega \times (0, T)$, and with $\partial\Omega$ denoting the boundary of Ω. We highlight that the formulation (2.1) is also similar in structure to the optimal control problem constrained by the Nagumo equation considered in [10, Section 5.2].

Secondly, we consider a reaction–diffusion control problem, based on the formulation in [2]:

$$\begin{aligned}
\min_{y,z,c} \quad & \frac{\beta_y}{2}\int_0^T\int_\Omega \left(y-\widehat{y}\right)^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta_z}{2}\int_0^T\int_\Omega \left(z-\widehat{z}\right)^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta_c}{2}\int_0^T\int_{\partial\Omega} c^2 \,\mathrm{d}s\,\mathrm{d}t \qquad (2.2)\\
\text{s.t.} \quad & \frac{\partial y}{\partial t} - D_1\nabla^2 y + k_1 y = -\gamma_1 yz + f \qquad \text{in } \Omega\times(0,T),\\
& \frac{\partial z}{\partial t} - D_2\nabla^2 z + k_2 z = -\gamma_2 yz + g \qquad \text{in } \Omega\times(0,T),\\
& D_1\frac{\partial y}{\partial n} = c \qquad \text{on } \partial\Omega\times(0,T),\\
& D_2\frac{\partial z}{\partial n} = 0 \qquad \text{on } \partial\Omega\times(0,T),\\
& y(\vec{x},0) = y_0(\vec{x}) \qquad \text{at } t=0,\\
& z(\vec{x},0) = z_0(\vec{x}) \qquad \text{at } t=0.
\end{aligned}$$

In this formulation, y and z denote state variables with corresponding desired states $\widehat{y}$ and $\widehat{z}$, c is again a control variable, f and g are given functions in space and time, y_0 and z_0 are given initial conditions, and $T>0$ is again a final time. The parameters β_c, D_1, and D_2 are positive, with β_y, β_z, k_1, k_2, γ_1, and γ_2 non-negative (and generally positive). In particular, at least one of β_y and β_z must be positive. We highlight that a key structural difference between the two examples is that (2.1) is a *distributed control problem*, whereas (2.2) is a *boundary control problem*. In both examples, the source terms f and g would often be zero in practice, but we allow non-zero functions so that test problems with analytic solutions are more readily available.

2.1. Derivation of coupled systems of PDEs. In order to apply our residual-based approach, the first step is to describe the solution of our time-dependent PDE-constrained optimization problems using a coupled system of PDEs, which define the *first-order optimality conditions.* A general approach, an outline of which is provided in [30, Chapter 3] for time-dependent (parabolic) PDE-constrained optimization problems, is to apply an *optimize-then-discretize* strategy. In that strategy one seeks the stationary points of a continuous Lagrangian involving the cost functional being minimized, as well as the PDE constraints and boundary conditions enforced by an *adjoint variable* (or *Lagrange multiplier*). One may then take Fréchet derivatives in the direction of the adjoint, control, and state variables, and test the conditions that the derivatives must be equal to zero using functions with appropriate continuity and differentiability properties and which satisfy suitable boundary conditions. Taking the derivatives in this order gives rise to *state equations*, *gradient equations*, and *adjoint equations*, respectively.

For instance, to briefly outline how this may be applied to the optimal control problem involving the Schlögl equation (2.1), the continuous Lagrangian reads

$$\begin{aligned}
\mathcal{L}(y,c,p) = {} & \frac{1}{2}\int_0^T\int_\Omega \left(y-\widehat{y}\right)^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta}{2}\int_0^T\int_\Omega c^2 \,\mathrm{d}\Omega\,\mathrm{d}t\\
& + \int_0^T\int_\Omega p_\Omega\left(\frac{\partial y}{\partial t} - \nabla^2 y - y + y^3 - c - f\right) \mathrm{d}\Omega\,\mathrm{d}t + \int_0^T\int_{\partial\Omega} p_{\partial\Omega}\left(y-h\right)\,\mathrm{d}s\,\mathrm{d}t,
\end{aligned}$$

where the adjoint variable is split into its components in the interior of Ω and boundary $\partial\Omega$, denoted p_Ω and $p_{\partial\Omega}$. Here, we have omitted the initial condition from the definition of the Lagrangian for ease of presentation. Using integration by parts and applying the Divergence Theorem allows us to rearrange the Lagrangian to obtain that

$$\begin{aligned}
\mathcal{L}(y,c,p) = {} & \frac{1}{2}\int_0^T\int_\Omega \left(y-\widehat{y}\right)^2 \,\mathrm{d}\Omega\,\mathrm{d}t + \frac{\beta}{2}\int_0^T\int_\Omega c^2 \,\mathrm{d}\Omega\,\mathrm{d}t\\
& + \int_0^T\int_\Omega y\left(-\frac{\partial p_\Omega}{\partial t} - \nabla^2 p_\Omega - p_\Omega + y^2 p_\Omega\right) \mathrm{d}\Omega\,\mathrm{d}t - \int_0^T\int_\Omega p_\Omega\left(c+f\right)\,\mathrm{d}\Omega\,\mathrm{d}t
\end{aligned}$$

$$+\int_{\Omega}\left[p_{\Omega}y\right]_{t=0}^{t=T}\,\mathrm{d}\Omega+\int_0^T\int_{\partial\Omega}\left(y\frac{\partial p_{\Omega}}{\partial n}-p_{\Omega}\frac{\partial y}{\partial n}\right)\,\mathrm{d}s\,\mathrm{d}t+\int_0^T\int_{\partial\Omega}p_{\partial\Omega}\left(y-h\right)\,\mathrm{d}s\,\mathrm{d}t.$$

Applying the Fréchet derivatives in the directions of p_{Ω}, c, and y, testing with appropriate functions, and relabelling the adjoint variable p_{Ω} as p, gives the state, gradient, and adjoint equations, respectively. Here,

$$\left.\begin{array}{rl} \frac{\partial y}{\partial t}-\nabla^2 y-y+y^3=c+f & \text{in } \Omega\times(0,T)\\ y(\vec{x},t)=h(\vec{x},t) & \text{on } \partial\Omega\times(0,T)\\ y(\vec{x},0)=y_0(\vec{x}) & \text{at } t=0 \end{array}\right\}\ \text{state equation} \tag{2.3}$$

$$\begin{array}{rl} \beta c-p=0 & \text{in } \Omega\times(0,T) \end{array}\ \Big\}\ \text{gradient equation} \tag{2.4}$$

$$\left.\begin{array}{rl} -\frac{\partial p}{\partial t}-\nabla^2 p-p+3y^2p=\widehat{y}-y & \text{in } \Omega\times(0,T)\\ p(\vec{x},t)=0 & \text{on } \partial\Omega\times(0,T)\\ p(\vec{x},T)=0 & \text{at } t=T \end{array}\right\}\ \text{adjoint equation} \tag{2.5}$$

Applying a similar approach to the reaction–diffusion control problem (2.2), the first-order optimality conditions are as follows (see also [2]):

$$\left.\begin{array}{rl} \frac{\partial y}{\partial t}-D_1\nabla^2 y+k_1y=-\gamma_1yz+f & \text{in } \Omega\times(0,T)\\ \frac{\partial z}{\partial t}-D_2\nabla^2 z+k_2z=-\gamma_2yz+g & \text{in } \Omega\times(0,T)\\ D_1\frac{\partial y}{\partial n}=c & \text{on } \partial\Omega\times(0,T)\\ D_2\frac{\partial z}{\partial n}=0 & \text{on } \partial\Omega\times(0,T)\\ y(\vec{x},0)=y_0(\vec{x}) & \text{at } t=0\\ z(\vec{x},0)=z_0(\vec{x}) & \text{at } t=0 \end{array}\right\}\ \text{state equations} \tag{2.6}$$

$$\begin{array}{rl} \beta_c c-p=0 & \text{on } \partial\Omega\times(0,T) \end{array}\ \Big\}\ \text{gradient equation} \tag{2.7}$$

$$\left.\begin{array}{rl} -\frac{\partial p}{\partial t}-D_1\nabla^2 p+k_1p+\gamma_1zp+\gamma_2zq=\beta_y\big(\widehat{y}-y\big) & \text{in } \Omega\times(0,T)\\ -\frac{\partial q}{\partial t}-D_2\nabla^2 q+k_2q+\gamma_1yp+\gamma_2yq=\beta_z\big(\widehat{z}-z\big) & \text{in } \Omega\times(0,T)\\ D_1\frac{\partial p}{\partial n}=0 & \text{on } \partial\Omega\times(0,T)\\ D_2\frac{\partial q}{\partial n}=0 & \text{on } \partial\Omega\times(0,T)\\ p(\vec{x},T)=0 & \text{at } t=T\\ q(\vec{x},T)=0 & \text{at } t=T \end{array}\right\}\ \text{adjoint equations} \tag{2.8}$$

Note how in both cases the first-order optimality conditions take the form of coupled time-dependent PDEs, with the state equations equipped with initial conditions, and the adjoint equations possessing final-time conditions.

2.2. Derivation of the Newton system. The formulations derived in the previous section, specifically (2.3)–(2.5) for the Schlögl problem and (2.6)–(2.8) for the reaction–diffusion control problem, may be discretized in space and result in coupled initial/final value problems in time, of the form

$$\begin{cases} M_u\mathbf{u}'(t)=\mathbf{f}(t,\mathbf{u},\mathbf{v}), & M_u\mathbf{u}(0)=M_u\mathbf{u}_0\in\mathbb{R}^N \text{ given}, \quad (2.9)\\ M_v\mathbf{v}'(t)=\mathbf{g}(t,\mathbf{u},\mathbf{v}), & M_v\mathbf{v}(T)=M_v\mathbf{v}_T\in\mathbb{R}^N \text{ given}, \quad (2.10) \end{cases}$$

for two vector-valued functions $\mathbf{u},\mathbf{v}:[0,T]\mapsto\mathbb{R}^N$ and with matrices $M_u,M_v\in\mathbb{R}^{N\times N}$. This system may be thought of as a coupled system of ordinary differential equations in the time variable. We highlight the key structures that allow us to re-write the optimization problem in the form (2.9)–(2.10): (i) the spatial derivatives are separate to the first-order time derivatives within the PDE constraints; (ii) a squared L^2-norm term measures the control within the objective function and a linear term incorporates the control within the constraints, allowing us to derive a linear relation between control and adjoint variables (as above). We believe the second structure could be relaxed in order to allow a nonlinear relation between control and adjoint, which may be incorporated into (2.9)–(2.10) through an additional algebraic constraint (i.e., not involving a time derivative). In certain cases, higher-order PDEs in time could also be written as (2.9)–(2.10) by rewriting these as a system of first-order equations.

In the case of the Schlögl example, substituting the gradient equation (2.4) into (2.3) leads to a coupled system of the form (2.9)–(2.10) with

$$\begin{aligned} \mathbf{u}(t) &:= u(t) \leftarrow y(t), \quad M_u \leftarrow \mathrm{Id}, \quad \mathbf{f}(t,\mathbf{u},\mathbf{v}) \leftarrow \nabla^2 u + u - u^3 + \frac{1}{\beta}v + f, \\ \mathbf{v}(t) &:= v(t) \leftarrow p(t), \quad M_v \leftarrow \mathrm{Id}, \quad \mathbf{g}(t,\mathbf{u},\mathbf{v}) \leftarrow -\nabla^2 v - v + 3u^2 v + u - \widehat{y}. \end{aligned}$$

We have used the notation "←", with the quantities on the left-hand side being the result of spatial discretization. The symbol Id refers to the discretized identity operator.

For the reaction–diffusion example, we again have a system of the form (2.9)–(2.10) with

$$\mathbf{u}(t) := \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \leftarrow \begin{bmatrix} y(t) \\ z(t) \end{bmatrix}, \quad M_u \leftarrow \begin{bmatrix} \mathrm{Id} & 0 \\ 0 & \mathrm{Id} \end{bmatrix}, \quad \mathbf{f}(t,\mathbf{u},\mathbf{v}) \leftarrow \begin{bmatrix} D_1\nabla^2 u_1 - k_1 u_1 - \gamma_1 u_1 u_2 + f \\ D_2\nabla^2 u_2 - k_2 u_2 - \gamma_2 u_1 u_2 + g \end{bmatrix},$$

$$\mathbf{v}(t) := \begin{bmatrix} v_1(t) \\ v_2(t) \end{bmatrix} \leftarrow \begin{bmatrix} p(t) \\ q(t) \end{bmatrix}, \quad M_v \leftarrow \begin{bmatrix} \mathrm{Id} & 0 \\ 0 & \mathrm{Id} \end{bmatrix},$$

$$\mathbf{g}(t,\mathbf{u},\mathbf{v}) \leftarrow \begin{bmatrix} -D_1\nabla^2 v_1 + k_1 v_1 + \gamma_1 u_2 v_1 + \gamma_2 u_2 v_2 + \beta_y\big(u_1 - \widehat{y}\big) \\ -D_2\nabla^2 v_2 + k_2 v_2 + \gamma_1 u_1 v_1 + \gamma_2 u_1 v_2 + \beta_z\big(u_2 - \widehat{z}\big) \end{bmatrix}.$$

Having derived systems of the above forms, we need to equip the relevant operators with boundary conditions. For example, in the reaction–diffusion example and with the notation above, using the gradient equation (2.7) gives that the first Neumann boundary condition reads $D_1 \frac{\partial u_1}{\partial n} = \frac{1}{\beta_c} v_1$.

Given a system of the form (2.9)–(2.10), let us assume that approximations $\widetilde{\mathbf{u}}_j$ and $\widetilde{\mathbf{v}}_j$ at time points $0 = t_0 < t_1 < \cdots < t_n = T$ are available and consider the interpolants

$$\widetilde{\mathbf{u}}(t) = \sum_{j=0}^{n} \ell_j(t)\widetilde{\mathbf{u}}_j \quad \text{and} \quad \widetilde{\mathbf{v}}(t) = \sum_{j=0}^{n} \ell_j(t)\widetilde{\mathbf{v}}_j, \tag{2.11}$$

where $\ell_j(t)$ are differentiable Lagrange functions satisfying $\ell_j(t_i) = \delta_{ij}$. We consider the associated residual functions

$$\begin{cases} \mathbf{r}_u(t) := \displaystyle\int_0^t \mathbf{f}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau))\,\mathrm{d}\tau - M_u\widetilde{\mathbf{u}}(t) + M_u\widetilde{\mathbf{u}}(0), & (2.12) \\ \mathbf{r}_v(t) := \displaystyle\int_0^t \mathbf{g}(\tau, \widetilde{\mathbf{u}}(\tau), \widetilde{\mathbf{v}}(\tau))\,\mathrm{d}\tau - M_v\widetilde{\mathbf{v}}(t) + M_v\widetilde{\mathbf{v}}(0). & (2.13) \end{cases}$$

It is important to note that the residuals are zero for *all* solutions that satisfy the system (2.9) and (2.10), *irrespective* of the initial/final condition for $\mathbf{u}/\mathbf{v}$. Hence, we still need to impose $M_u\widetilde{\mathbf{u}}(0) = M_u\mathbf{u}_0$ and $M_v\widetilde{\mathbf{v}}(T) = M_v\mathbf{v}_T$ explicitly.

Denoting by $\mathbf{r}_{u,j}$ and $\mathbf{r}_{v,j}$ the approximations to the residuals $\mathbf{r}_u(t_j)$ and $\mathbf{r}_v(t_j)$ at time points t_j, respectively, we can write

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

and

$$[\mathbf{r}_{v,0}, \mathbf{r}_{v,1}, \ldots, \mathbf{r}_{v,n}] = [\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_n]Q + M_v[\widetilde{\mathbf{v}}_0 - \widetilde{\mathbf{v}}_0, \widetilde{\mathbf{v}}_0 - \widetilde{\mathbf{v}}_1, \ldots, \widetilde{\mathbf{v}}_0 - \widetilde{\mathbf{v}}_n].$$

Here, the $(n+1) \times (n+1)$ collocation matrix Q corresponds to cumulative integration, i.e.,

$$q_{ij} = \int_0^{t_j} \ell_i(\tau)\,\mathrm{d}\tau, \quad i,j = 0,1,\ldots,n, \tag{2.14}$$

with q_{00} being the top-left entry of Q. (The name of that matrix can be remembered by thinking of "quadrature".) The motivation for using forward integration for both (2.12) and (2.13), as opposed to using coupled forward–backward integration as in our previous work [14], is to avoid the introduction of separate time-collocation matrices $Q_{\mathbf{f}}$ and $Q_{\mathbf{g}}$ (in [14] these matrices were called C_u and C_v, respectively). We will see that using a common collocation matrix for both $\mathbf{f}$ and $\mathbf{g}$ results in a regular matrix structure which we can exploit for the construction of a computationally efficient preconditioner.

We define the *global residual function* $\mathbf{R} : \mathbb{R}^{2N(n+1)} \to \mathbb{R}^{2N(n+1)}$ as

$$\mathbf{R} : \underbrace{\begin{bmatrix} \widetilde{\mathbf{u}}_0 \\ \widetilde{\mathbf{v}}_0 \\ \widetilde{\mathbf{u}}_1 \\ \widetilde{\mathbf{v}}_1 \\ \widetilde{\mathbf{u}}_2 \\ \widetilde{\mathbf{v}}_2 \\ \vdots \\ \widetilde{\mathbf{u}}_n \\ \widetilde{\mathbf{v}}_n \end{bmatrix}}_{=:\,\mathbf{w}} \mapsto \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \sum_{i=0}^n q_{i1}\mathbf{f}_i \\ \sum_{i=0}^n q_{i1}\mathbf{g}_i \\ \sum_{i=0}^n q_{i2}\mathbf{f}_i \\ \sum_{i=0}^n q_{i2}\mathbf{g}_i \\ \vdots \\ \sum_{i=0}^n q_{in}\mathbf{f}_i \\ \sum_{i=0}^n q_{in}\mathbf{g}_i \end{bmatrix} + \begin{bmatrix} \widetilde{\mathbf{u}}_0 - \mathbf{u}_0 \\ \widetilde{\mathbf{v}}_n - \mathbf{v}_T \\ M_u(\widetilde{\mathbf{u}}_0 - \widetilde{\mathbf{u}}_1) \\ M_v(\widetilde{\mathbf{v}}_0 - \widetilde{\mathbf{v}}_1) \\ M_u(\widetilde{\mathbf{u}}_0 - \widetilde{\mathbf{u}}_2) \\ M_v(\widetilde{\mathbf{v}}_0 - \widetilde{\mathbf{v}}_2) \\ \vdots \\ M_u(\widetilde{\mathbf{u}}_0 - \widetilde{\mathbf{u}}_n) \\ M_v(\widetilde{\mathbf{v}}_0 - \widetilde{\mathbf{v}}_n) \end{bmatrix}. \tag{2.15}$$

We aim to apply the Newton method $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - [\mathcal{D}\mathbf{R}(\mathbf{w}^{(k)})]^{-1}\mathbf{R}(\mathbf{w}^{(k)})$ in order to approximately solve $\mathbf{R}\mathbf{w} = \mathbf{0}$. The Jacobian $\mathbf{J} = \mathcal{D}\mathbf{R}(\mathbf{w})$ is readily computed as:

$$\mathbf{J} = \left[\begin{array}{cc|ccccc} I_N & O & \cdots & \cdots & \cdots & O & O \\ O & O & \cdots & \cdots & \cdots & O & I_N \\ \hline q_{01}\mathcal{D}_u\mathbf{f}_0 + M_u & q_{01}\mathcal{D}_v\mathbf{f}_0 & q_{11}\mathcal{D}_u\mathbf{f}_1 - M_u & q_{11}\mathcal{D}_v\mathbf{f}_1 & \cdots & q_{n1}\mathcal{D}_u\mathbf{f}_n & q_{n1}\mathcal{D}_v\mathbf{f}_n \\ q_{01}\mathcal{D}_u\mathbf{g}_0 & q_{01}\mathcal{D}_v\mathbf{g}_0 + M_v & q_{11}\mathcal{D}_u\mathbf{g}_1 & q_{11}\mathcal{D}_v\mathbf{g}_1 - M_v & \cdots & q_{n1}\mathcal{D}_u\mathbf{g}_n & q_{n1}\mathcal{D}_v\mathbf{g}_n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ q_{0n}\mathcal{D}_u\mathbf{f}_0 + M_u & q_{0n}\mathcal{D}_v\mathbf{f}_0 & q_{1n}\mathcal{D}_u\mathbf{f}_1 & q_{1n}\mathcal{D}_v\mathbf{f}_1 & \cdots & q_{nn}\mathcal{D}_u\mathbf{f}_n - M_u & q_{nn}\mathcal{D}_v\mathbf{f}_n \\ q_{0n}\mathcal{D}_u\mathbf{g}_0 & q_{0n}\mathcal{D}_v\mathbf{g}_0 + M_v & q_{1n}\mathcal{D}_u\mathbf{g}_1 & q_{1n}\mathcal{D}_v\mathbf{g}_1 & \cdots & q_{nn}\mathcal{D}_u\mathbf{g}_n & q_{nn}\mathcal{D}_v\mathbf{g}_n - M_v \end{array}\right], \tag{2.16}$$

with $\mathcal{D}_u\mathbf{f}_i \in \mathbb{R}^{N\times N}$ denoting the Jacobian of $\mathbf{f}(t, \mathbf{u}, \mathbf{v})$ with respect to $\mathbf{u}$, evaluated at the linearization point $(t_i, \widetilde{\mathbf{u}}_i, \widetilde{\mathbf{v}}_i)$. The other Jacobians $\mathcal{D}_v\mathbf{f}_i, \mathcal{D}_u\mathbf{g}_i, \mathcal{D}_v\mathbf{g}_i$ are denoted analogously.

At each Newton iteration we need to solve a linear system with the matrix $\mathbf{J}$ defined in (2.16). In order to simplify its structure and make it more amenable to preconditioning, we apply a block-column transformation by right-multiplication with a matrix K, adding trailing block-columns to the first and second block-columns of $\mathbf{J}$ and thereby eliminating all appearances of M_u and M_v in the first two block-columns and creating an invertible pivot in the leading 2-by-2 block. More precisely, the matrix K is given by

$$K = \begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ 1 & & 1 & & \\ \vdots & & & \ddots & \\ 1 & & & & 1 \end{bmatrix} \otimes I_{2N}.$$

We denote the resulting matrix by $\widetilde{\mathbf{J}} = \mathbf{J}\cdot K$,

$$\widetilde{\mathbf{J}} = \left[\begin{array}{cc|ccccc} I_N & O & \cdots & \cdots & \cdots & O & O \\ O & I_N & \cdots & \cdots & \cdots & O & I_N \\ \hline \mathcal{D}_u\widetilde{\mathbf{f}}_1 & \mathcal{D}_v\widetilde{\mathbf{f}}_1 & q_{11}\mathcal{D}_u\mathbf{f}_1 - M_u & q_{11}\mathcal{D}_v\mathbf{f}_1 & \cdots & q_{n1}\mathcal{D}_u\mathbf{f}_n & q_{n1}\mathcal{D}_v\mathbf{f}_n \\ \mathcal{D}_u\widetilde{\mathbf{g}}_1 & \mathcal{D}_v\widetilde{\mathbf{g}}_1 & q_{11}\mathcal{D}_u\mathbf{g}_1 & q_{11}\mathcal{D}_v\mathbf{g}_1 - M_v & \cdots & q_{n1}\mathcal{D}_u\mathbf{g}_n & q_{n1}\mathcal{D}_v\mathbf{g}_n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathcal{D}_u\widetilde{\mathbf{f}}_n & \mathcal{D}_v\widetilde{\mathbf{f}}_n & q_{1n}\mathcal{D}_u\mathbf{f}_1 & q_{1n}\mathcal{D}_v\mathbf{f}_1 & \cdots & q_{nn}\mathcal{D}_u\mathbf{f}_n - M_u & q_{nn}\mathcal{D}_v\mathbf{f}_n \\ \mathcal{D}_u\widetilde{\mathbf{g}}_n & \mathcal{D}_v\widetilde{\mathbf{g}}_n & q_{1n}\mathcal{D}_u\mathbf{g}_1 & q_{1n}\mathcal{D}_v\mathbf{g}_1 & \cdots & q_{nn}\mathcal{D}_u\mathbf{g}_n & q_{nn}\mathcal{D}_v\mathbf{g}_n - M_v \end{array}\right],$$

```
1. Initialize $\widetilde{\mathbf{u}}_j := \mathbf{u}_0$ and $\widetilde{\mathbf{v}}_j := \mathbf{v}_T$ for $j = 0, 1, \ldots, n$
2. Set $k := 0$ and $\mathbf{w}^{(0)} := [\widetilde{\mathbf{u}}_0; \widetilde{\mathbf{v}}_0; \widetilde{\mathbf{u}}_1; \widetilde{\mathbf{v}}_1; \ldots; \widetilde{\mathbf{u}}_n; \widetilde{\mathbf{v}}_n]$    (';' stands for row-wise concatenation)
3. Evaluate $\mathbf{R}(\mathbf{w}^{(k)}) =: \begin{bmatrix} a \\ b \end{bmatrix}$, partitioned such that $a \in \mathbb{R}^{2N}$, $b \in \mathbb{R}^{2Nn}$    (note that $a = 0$)
4. If error criterion is satisfied, stop
5. Solve $S\widetilde{y} = b$ where $S := D - CB$    ($S$ varies with each iteration $k$)
6. Set $\widetilde{x} := -B\widetilde{y}$
7. Compute $\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} := K \begin{bmatrix} \widetilde{x} \\ \widetilde{y} \end{bmatrix}$
8. Set $\mathbf{w}^{(k+1)} := \mathbf{w}^{(k)} - \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}$ and go to step 3
```

Fig. 2.1: Pseudocode for the residual-based Newton solver. The notation follows that of Section 2.2.

with $\mathcal{D}_u\widetilde{\mathbf{f}}_i := \sum_{j=0}^{n} q_{ji}\mathcal{D}_u\mathbf{f}_j$, $\mathcal{D}_u\widetilde{\mathbf{g}}_i := \sum_{j=0}^{n} q_{ji}\mathcal{D}_u\mathbf{g}_j$, $\mathcal{D}_v\widetilde{\mathbf{f}}_i := \sum_{j=0}^{n} q_{ji}\mathcal{D}_v\mathbf{f}_j$, and $\mathcal{D}_v\widetilde{\mathbf{g}}_i := \sum_{j=0}^{n} q_{ji}\mathcal{D}_v\mathbf{g}_j$, for $i = 1, \ldots, n$. Let us partition this matrix as

$$\widetilde{\mathbf{J}} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \quad A = I_{2N}.$$

Within each Newton step, we then solve

$$\mathbf{J} \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{x} \\ \widetilde{y} \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix},$$

or equivalently, using the Schur complement $S = D - CA^{-1}B = D - CB$,

$$\begin{bmatrix} A & B \\ O & S \end{bmatrix} \begin{bmatrix} \widetilde{x} \\ \widetilde{y} \end{bmatrix} = \begin{bmatrix} a \\ b - Ca \end{bmatrix}. \tag{2.17}$$

A simplification is obtained by noting that, within the Newton iteration, we have as right-hand side vector the global residual evaluated for the previous iterate, $\begin{bmatrix} a \\ b \end{bmatrix} = \mathbf{R}(\mathbf{w}^{(k)})$. If $\mathbf{w}^{(k)}$ is such that $\widetilde{\mathbf{u}}_0 = \mathbf{u}_0$ and $\widetilde{\mathbf{v}}_n = \mathbf{v}_T$, then $a = 0$. As a consequence, the right-hand side of (2.17) does not require the evaluation of Ca, which involves the submatrix C of the Jacobian $\widetilde{\mathbf{J}}$ that may not be explicitly available and may need to be approximated by finite differencing.

A pseudocode for the resulting residual-based Newton method is given in Figure 2.1. Note that the matrix computations in steps 6–7 can be performed cheaply: B is a very sparse matrix of size $2N$-by-$2Nn$ whose only nonzeros correspond to an $N \times N$ identify matrix in its bottom right, while K can be applied efficiently using its Kronecker representation. The main computational cost of this Newton method is concentrated in step 5, the solution of the Schur complement system. In what follows we will introduce a preconditioner for that problem, allowing the development of an efficient Newton–Krylov solver.

2.3. Preconditioner. As a preconditioner for the Schur complement S arising in (2.17) (and step 5 of Figure 2.1) we consider the following matrix:

$$P = \widehat{D} - \widehat{C}B.$$

Here, $\widehat{D}$ and $\widehat{C}$ are approximations to D and C, respectively, defined as

$$\widehat{D} = \begin{bmatrix} q_{11} & q_{21} & \cdots & q_{n1} \\ q_{12} & q_{22} & \cdots & q_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{nn} \end{bmatrix} \otimes G \quad \text{and} \quad \widehat{C} = \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \vdots \\ \hat{q}_n \end{bmatrix} \otimes G,$$

with $\hat{q}_i = \sum_{j=0}^{n} q_{ji}$ and the matrix G corresponding to

$$G := \begin{bmatrix} \mathcal{D}_u \mathbf{f} - M_u & \mathcal{D}_v \mathbf{f} \\ \mathcal{D}_u \mathbf{g} & \mathcal{D}_v \mathbf{g} - M_v \end{bmatrix} \in \mathbb{R}^{2N \times 2N},$$

where the Jacobians of $\mathbf{f}$ and $\mathbf{g}$ are evaluated for the current Newton iterate at some time point in $[0, T]$. (In our experiments, we use the mid-point $t = T/2$ as the time evaluation point.) In order to use this preconditioner efficiently, we need to be able to apply P^{-1} cheaply. By the Sherman–Morrison formula we have

$$P^{-1} = \widehat{D}^{-1} + \widehat{D}^{-1}\widehat{C}(I_{2N} - B\widehat{D}^{-1}\widehat{C})^{-1}B\widehat{D}^{-1} = \underbrace{[I_{2Nn} + \widehat{D}^{-1}\widehat{C}(I_{2N} - B\widehat{D}^{-1}\widehat{C})^{-1}B]}_{=:F}\,\widehat{D}^{-1}.$$

Now using the facts that

$$\widehat{D}^{-1} = \begin{bmatrix} q_{11} & q_{21} & \cdots & q_{n1} \\ q_{12} & q_{22} & \cdots & q_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{nn} \end{bmatrix}^{-1} \otimes G^{-1}$$

and hence

$$\widehat{D}^{-1}\widehat{C} = \left(\begin{bmatrix} q_{11} & q_{21} & \cdots & q_{n1} \\ q_{12} & q_{22} & \cdots & q_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{nn} \end{bmatrix}^{-1} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \vdots \\ \hat{q}_n \end{bmatrix} \right) \otimes \begin{bmatrix} I_N & O \\ O & I_N \end{bmatrix} =: \begin{bmatrix} \widetilde{q}_1 \\ \widetilde{q}_2 \\ \vdots \\ \widetilde{q}_n \end{bmatrix} \otimes \begin{bmatrix} I_N & O \\ O & I_N \end{bmatrix},$$

a simple calculation shows that

$$F = I_{2Nn} + \widehat{D}^{-1}\widehat{C}(I_{2N} - B\widehat{D}^{-1}\widehat{C})^{-1}B = \left[\begin{array}{cc|cc|cc|cc} I_N & & & & & & O & O \\ & I_N & & & & & O & \widetilde{q}_1 I_N/(1-\widetilde{q}_n) \\ \hline & & I_N & & & & O & O \\ & & & I_N & & & O & \widetilde{q}_2 I_N/(1-\widetilde{q}_n) \\ \hline & & & & \ddots & & \vdots & \vdots \\ & & & & & \ddots & \vdots & \vdots \\ \hline & & & & & & I_N & O \\ & & & & & & O & (1+\widetilde{q}_n/(1-\widetilde{q}_n))I_N \end{array} \right].$$

This allows us to apply $P^{-1} = F\widehat{D}^{-1}$ cheaply. In particular, we have

$$P^{-1}\mathbf{v} = F\widehat{D}^{-1}\mathbf{v} = F\,\mathtt{vec}\left(G^{-1}\mathtt{mat}(\mathbf{v}) \begin{bmatrix} q_{11} & q_{21} & \cdots & q_{n1} \\ q_{12} & q_{22} & \cdots & q_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ q_{1n} & q_{2n} & \cdots & q_{nn} \end{bmatrix}^{-T} \right), \tag{2.18}$$

with the `mat` operation transforming a vector $\mathbf{v} \in \mathbb{R}^{2Nn}$ into an $2N$-by-n matrix, and `vec` being the inverse operation. All computational costs involved in applying the preconditioner are linear with respect to N, with the exception of creating a factorization of the $2N$-by-$2N$ Jacobian matrix G, which needs to be done once per Newton iteration.

2.4. Newton–Krylov implementation. We do not wish to form the Schur complement $S = D - CB$ explicitly, but we will instead approximate its action within a Krylov method using finite differencing. The resulting Newton–Krylov method is standard and described in detail, e.g., in [21, Chapter 3]. We will therefore only outline the key parts required for its implementation.

Recall from (2.16) and the subsequent derivations that the matrix D, which is the lower-right $2Nn$-by-$2Nn$ submatrix of $\widetilde{\mathbf{J}} = \mathbf{J} \cdot K$, is the same as the lower-right submatrix of $\mathbf{J} = \mathcal{D}\mathbf{R}(\mathbf{w})$. We can therefore obtain an approximation to the matrix-vector product Dx as a finite difference

$$\left[\frac{\mathbf{R}(\mathbf{w} + h[0_{2N\times 1}; x]) - \mathbf{R}(\mathbf{w})}{h}\right]_{2N+1:2N(n+1)} \approx Dx,$$

where again ‘;’ corresponds to row-wise concatenation and the MATLAB operator ‘:’ was used to extract vector entries by their indices. For the appropriate choice of the parameter h in floating point arithmetic, we refer to [21, Section 3.2.1]. Similarly, the action of the lower-left $2Nn$-by-$2N$ submatrix C of $\widetilde{\mathbf{J}} = \mathbf{J} \cdot K$ on a vector Bx can be approximated as

$$\left[\frac{\mathbf{R}(\mathbf{w} + hK[Bx; 0_{2Nn\times 1}]) - \mathbf{R}(\mathbf{w})}{h}\right]_{2N+1:2N(n+1)} \approx C(Bx).$$

Overall, three evaluations of $\mathbf{R}(\cdot)$ are required to obtain an approximation of $Sx = Dx - C(Bx)$. If multiple such approximations with different vectors x are required consecutively, e.g., within a Krylov iteration, then the evaluation of $\mathbf{R}(\mathbf{w})$ can be reused for all of them.

The left-preconditioned system $P^{-1}Sx = P^{-1}b$ is solved using GMRES [28]. For these inner solves, the standard Eisenstat–Walker residual stopping criterion based on the norm of the residual, $\|\mathbf{R}(\mathbf{w}^{(k)})\|$, is used [9]; see also [21, Section 3.2.3].

2.5. Stopping criterion. There are several options to stop our method, with the most obvious one being a norm criterion for the residual evaluated in step 3 of the algorithm in Fig. 2.1: stop at Newton iteration k if $\|\mathbf{R}(\mathbf{w}^{(k)})\| \leq \texttt{tol}\, \|\mathbf{R}(\mathbf{w}^{(0)})\|$ for some user-specified tolerance `tol`.

An alternative option is to estimate the error in the iterate

$$\mathbf{w}^{(k)} = \begin{bmatrix} \widetilde{\mathbf{u}}^{(k)} \\ \widetilde{\mathbf{v}}^{(k)} \end{bmatrix}$$

from the difference $\mathbf{w}^{(k+s)} - \mathbf{w}^{(k)}$ for some integer lag parameter $s \geq 1$. This strategy gives good results if the method converges rapidly, as in this case $\mathbf{w}^{(k+s)}$ will be an “accurate” (relative to $\mathbf{w}^{(k)}$) approximation for the vector to which the algorithm converges. As the state ($\widetilde{\mathbf{u}}^{(k)}$) and adjoint components ($\widetilde{\mathbf{v}}^{(k)}$) in $\mathbf{w}^{(k)}$ often vary significantly in magnitude, it is advisable to treat them separately when estimating the error. We therefore suggest to use the following estimate for relative error of $\widetilde{\mathbf{u}}^{(k)}$:

$$\texttt{errest}_{\mathbf{u}}^{(k)} = \frac{\max_{j=0,1,\ldots,n} \|\widetilde{\mathbf{u}}_j^{(k+s)} - \widetilde{\mathbf{u}}_j^{(k)}\|_\infty}{\max_{j=0,1,\ldots,n} \|\widetilde{\mathbf{u}}_j^{(k+s)}\|_\infty},$$

and an analogous estimate for the relative error of $\widetilde{\mathbf{v}}^{(k)}$. Together with an estimate for the expected stagnation error level (dependent on the space and time discretization), these estimators provide a practical stopping criterion. We will illustrate this in Section 3.

3. Numerical experiments. Our numerical experiments are guided by two nonlinear model problems, the 2D Schlögl problem and a 2D reaction–diffusion problem, both posed on the spatial domain $\Omega = (-1, 1)^2$ and with analytic solutions given in Appendix A (with $\alpha = 0.1$ for the Schlögl problem). The experiments have been performed in MATLAB 2019a on a Windows 10 laptop with 8 GB RAM and an Intel(R) Core(TM) i7-8650U CPU running at 2 GHz. A MATLAB implementation of our method, including the preconditioner proposed in Section 2.3, as well as scripts reproducing these experiments, can be downloaded from `https://github.com/nla-group/pdeoptim`. We use Chebfun [7] to generate the spectral collocation matrices in space and time.

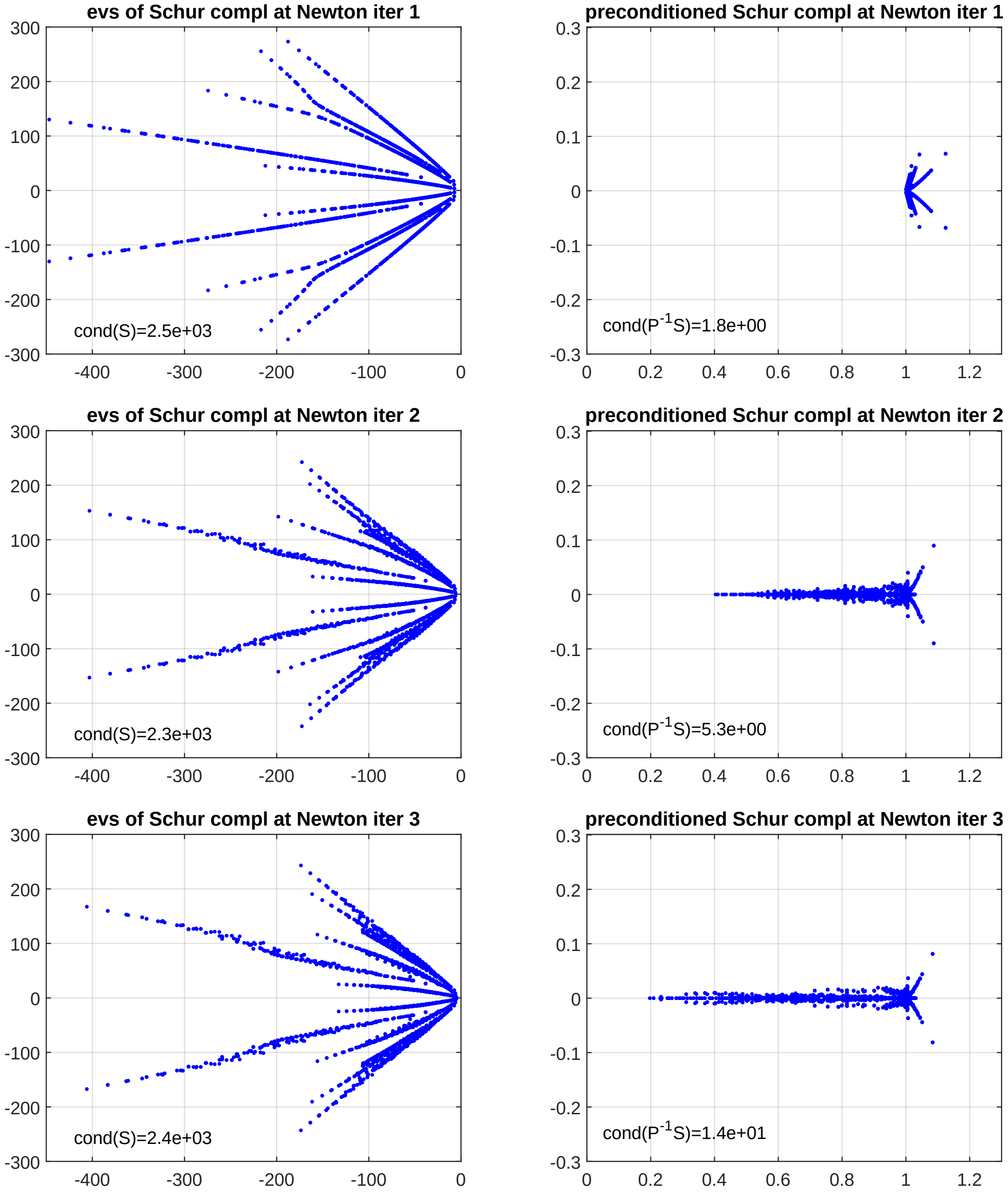

Fig. 3.1: Eigenvalues of the Schur complement S and its preconditioned counterpart $P^{-1}S$ for the 2D Schlögl problem with finite difference discretization. Only the first three Newton iterations are shown as the subsequent ones look qualitatively similar. The condition numbers of the Schur complement S in the first seven iterations (until stagnation of the Newton method as determined in Section 3.2) are $[2.5, 2.3, 2.4, 2.7, 3.0, 3.0, 3.0] \times 10^3$*, and those of the preconditioned Schur complement* $P^{-1}S$ *are* $[1.8, 5.3, 14.2, 24.3, 27.0, 27.4, 27.4]$.

Table 3.1: Outer Newton and inner GMRES iterations required to solve the Schlögl problem using a spectral space discretization, to an accuracy below 10^{-10}.

β	outer iterations	inner iterations	total inner	final accuracy	total time (s)
$1e-1$	4	4, 14, 18, 30	66	$6.92e-12$	0.405
$1e-2$	7	1, 4, 6, 8, 12, 16, 25	72	$6.27e-12$	0.398
$1e-3$	8	1, 2, 4, 6, 9, 12, 20, 35	89	$3.35e-13$	0.480
$1e-4$	8	1, 2, 4, 6, 9, 7, 12, 30	71	$8.49e-13$	0.441
$1e-5$	8	1, 2, 4, 6, 9, 9, 10, 25	66	$1.51e-12$	0.409

3.1. Visual inspection of the preconditioner.

We first explore visually the properties of our preconditioner from Section 2.3 with the help of the Schlögl problem, discretized with finite differences in space. The following parameters have been chosen: final time $T = 2$, regularization parameter $\beta = 0.01$, $n_x = 16$ equidistant interior grid points in each spatial dimension, and degree $n = 5$ Chebyshev collocation on $[0, T]$, that is we use the time collocation points:

$$t_j = T\,\frac{1-\cos(j\pi/n)}{2}, \quad j = 0, 1, \ldots, n.$$

In Figure 3.1 we show the eigenvalues of both the unpreconditioned (left) and preconditioned (right) Schur complements, S and $P^{-1}S$, at the first three Newton iterations. We observe that $P^{-1}S$ has tightly clustered eigenvalues away from the origin, which serves as an *indication* (though not a guarantee) that fast GMRES convergence can be expected for this preconditioned system. We also show, at the bottom left of each eigenvalue plot, the condition numbers of S and $P^{-1}S$, respectively. Again, the reduced condition number of the preconditioned system indicates that the preconditioner is effective. We will now turn to some concrete numerical tests to confirm this.

3.2. Schlögl problem, spectral in space.

To test the achievable accuracy of our solver to the extreme, as well as the robustness with respect to the regularization parameter β, we consider the Schlögl 2D example with a spectral space discretization. We use $n_x = 12$ interior grid points in each spatial dimension and degree $n = 10$ Chebyshev collocation on $[0, T = 2]$ (i.e., $n + 1 = 11$ time collocation points). The error of the computed state and adjoint solutions are obtained by taking the absolute deviation on the spectral collocation grid over all time steps, i.e.,

$$\texttt{err}_{\mathbf{u}} = \frac{\max_{j=0,1,\ldots,n} \|\mathbf{u}_j - \widetilde{\mathbf{u}}_j\|_\infty}{\max_{j=0,1,\ldots,n} \|\mathbf{u}_j\|_\infty},$$

where $\widetilde{\mathbf{u}}_j$ is the computed approximation to $\mathbf{u}(t_j)$, the latter of which is known analytically. Likewise, the error in the adjoint, $\texttt{err}_{\mathbf{v}}$, is computed. The results are shown in Figure 3.2, with the number of Newton iterations varying from 0 to 10. A number of zero Newton iterations corresponds to using the constant initial guess $\widetilde{\mathbf{u}}_j := \mathbf{u}_0$ and $\widetilde{\mathbf{v}}_j := \mathbf{v}_T$ for $j = 0, 1, \ldots, n$.

Figure 3.2 also indicates an error level of 10^{-10}, which we have used in Table 3.1 to evaluate the number of outer Newton iterations and inner GMRES iterations, as well as timings, required to achieve that accuracy. More precisely, we consider the algorithm to have converged to the target accuracy if

$$\texttt{err} := \max\{\texttt{err}_{\mathbf{u}}, \texttt{err}_{\mathbf{v}}\} \le 10^{-10}.$$

As expected, the total computation time is roughly proportional to the number of inner GMRES iterations performed, and this number increases steadily as the Newton iterates approach the sought minimum of the residual norm. Overall, the number of inner iterations and total time does not seem to depend much on the value of the regularization parameter β, and we can solve complex optimal control problems in a fraction of a second using GMRES, up to spectral accuracy, for every β tested.

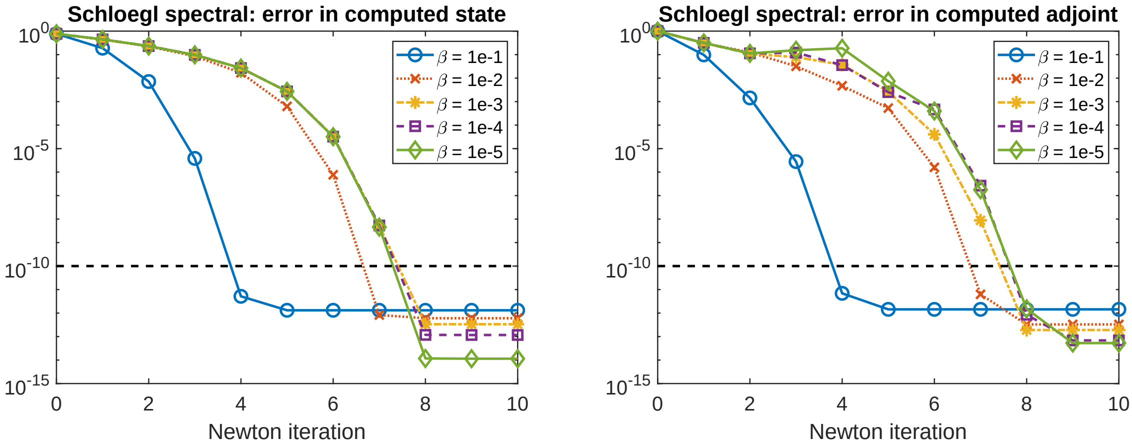

Fig. 3.2: Error in the computed state (left) and adjoint (right) variables for the Schlögl problem using a spectral space discretization, for varying regularization parameter β.

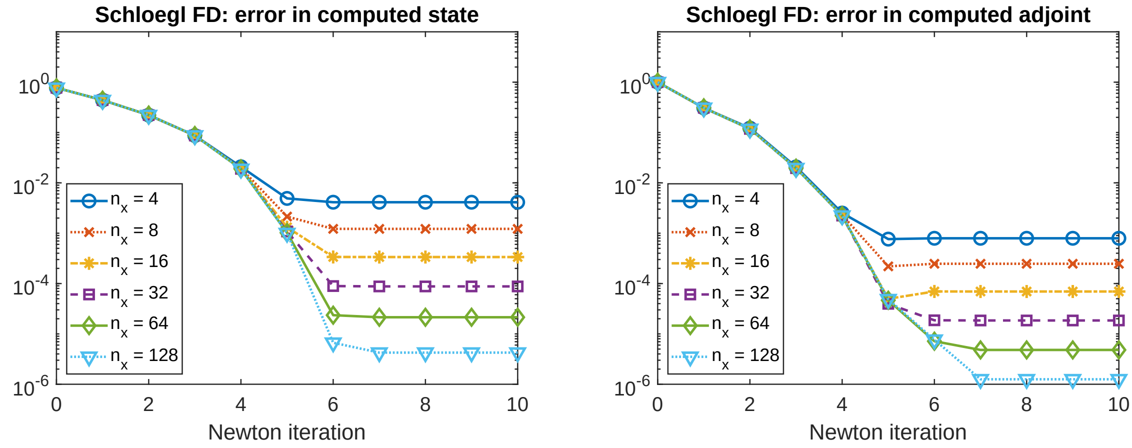

Fig. 3.3: Error in the computed state (left) and adjoint (right) variables for the Schlögl problem using a finite difference space discretization, with a varying number of grid points n_x.

3.3. Schlögl problem, finite differences in space. We now consider the Schlögl example with a finite difference discretization in space, while still using a spectral method in time. This example intends to demonstrate that mixing these two discretizations is a viable option, as the spectral time discretization helps to keep the number of solution vectors $\widetilde{\mathbf{u}}_j, \widetilde{\mathbf{v}}_j \in \mathbb{R}^N$ $(j = 0, 1, \ldots, n)$ small, thereby reducing the overall memory consumption and arithmetic cost. This is in contrast to previous approaches that have used a backward Euler approximation in time and hence required a much larger number of time steps, at the cost of increased linear algebra complexity.

In this test we use degree $n = 5$ Chebyshev collocation in time and vary the number of equidistant spatial interior grid points $n_x = 4, 8, 16, 32, 64, 128$ in both coordinate directions. The time interval is again $[0, T = 2]$, and the errors of the computed state and adjoint solutions are obtained as before.

The results are presented in Figure 3.3 and Table 3.2. We first observe that, as the number of spatial grid points is doubled, the finally attained accuracy improves by a factor of about 4. This is consistent with the second-order centred finite difference scheme, and also indicates that the time discretization is sufficiently fine. We further observe that, independently of n_x, the error of the space discretization is reached after about 6 Newton iterations. By inspecting the number of inner GMRES iterations in

Table 3.2: Outer Newton and inner GMRES iterations required to solve the Schlögl problem using a finite difference space discretization.

n_x	outer iterations	inner iterations	total inner	final accuracy	total time (s)
4	6	1, 3, 4, 6, 9, 10	33	$4.13e-03$	0.0298
8	6	1, 3, 4, 6, 9, 9	32	$1.22e-03$	0.0466
16	6	1, 3, 4, 6, 9, 9	32	$3.37e-04$	0.0923
32	6	1, 3, 4, 6, 9, 9	32	$8.92e-05$	0.388
64	6	1, 3, 4, 6, 9, 9	32	$2.35e-05$	1.43
128	6	1, 3, 4, 6, 9, 9	32	$7.67e-06$	6.71

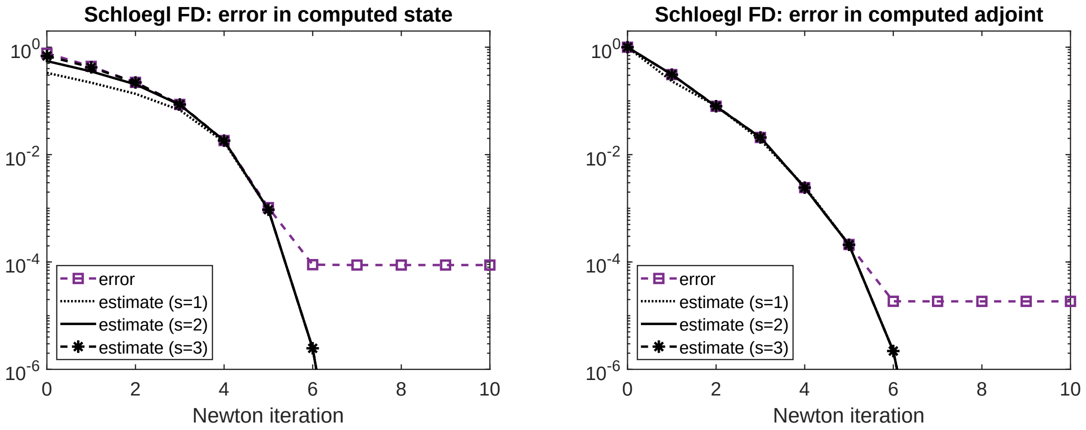

Fig. 3.4: Error and error estimates for the computed state (left) and adjoint (right) for the Schlögl problem using a finite difference space discretization, with $n_x = 32$ interior grid points. The lag parameter for the error estimation is varied between $s = 1, 2, 3$.

Table 3.3, we find that the method is robust with respect to refinements in space, testament to an effective preconditioner. As a consequence, the total computation time scales close to linearly in the number of overall spatial grid points, n_x^2. We highlight that if n_x becomes very large, the factorization of the matrix G in (2.18) at each Newton iteration could become a computational bottleneck, although our results demonstrate that high accuracy can be achieved for moderate n_x, thus mitigating this issue.

We also show in Figure 3.4 the error at each Newton iteration when $n_x = 32$ interior grid points are used in space, as well as the error estimate discussed in Section 2.5, using the lag parameter $s = 1, 2, 3$. Note how the error estimates follow the actual error very closely until the method stagnates on the level of the spatial/temporal discretization error. Together with estimates for the expected discretization error, this error estimate is a useful stopping criterion for our algorithm, and the user can specify a desired tolerance based on a reasonable expected discretization error given the n and n_x chosen.

3.4. Reaction–diffusion problem, finite differences in space. We now turn our attention to a nonlinear reaction–diffusion problem discretized using finite differences in space. We use degree $n = 10$ Chebyshev collocation on the time interval $[0, T = 1]$, and parameters $\beta_y = 1$, $\beta_z = 1$, $\beta_c = 0.01$, $D_1 = 0.5$, $D_2 = 1$, $k_1 = 1$, $k_2 = 1$, $\gamma_1 = 0.4$, $\gamma_2 = 0.6$, noting that the method demonstrates robustness with respect to reasonable modifications of these parameters.

The results of the Newton–Krylov method are shown in Figure 3.5 and Table 3.3. We can see that the error of the space discretization is reached after about 5 Newton iterations, and that a slightly larger n_x is required to obtain a satisfactory numerical solution than for the Schögl problem. As shown in Figure 3.5, the very low value of $n_x = 4$ (with n_x including boundary points for this problem, due

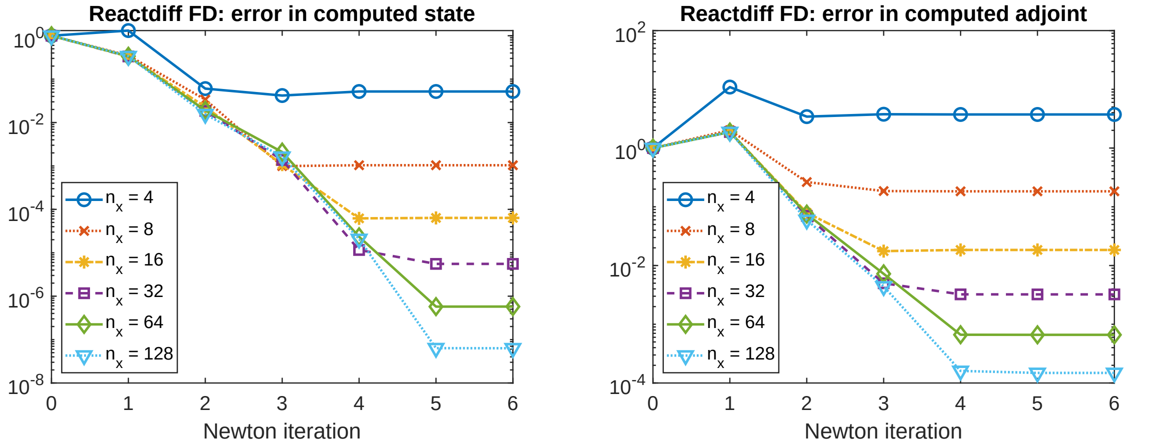

Fig. 3.5: Error in the computed state (left) and adjoint (right) variables for the reaction–diffusion problem using a finite difference space discretization, with a varying number of grid points n_x.

Table 3.3: Outer Newton and inner GMRES iterations required to solve the reaction–diffusion problem using a finite difference space discretization.

n_x	outer iterations	inner iterations	total inner	final accuracy	total time (s)
4	4	12, 23, 33, 60	128	$3.73e+00$	0.556
8	4	18, 28, 39, 68	153	$1.83e-01$	0.915
16	4	18, 30, 38, 60	146	$1.84e-02$	2.62
32	4	18, 31, 37, 55	141	$3.23e-03$	7.90
64	4	18, 31, 36, 50	135	$6.63e-04$	38.4
128	4	18, 32, 36, 51	137	$1.61e-04$	221

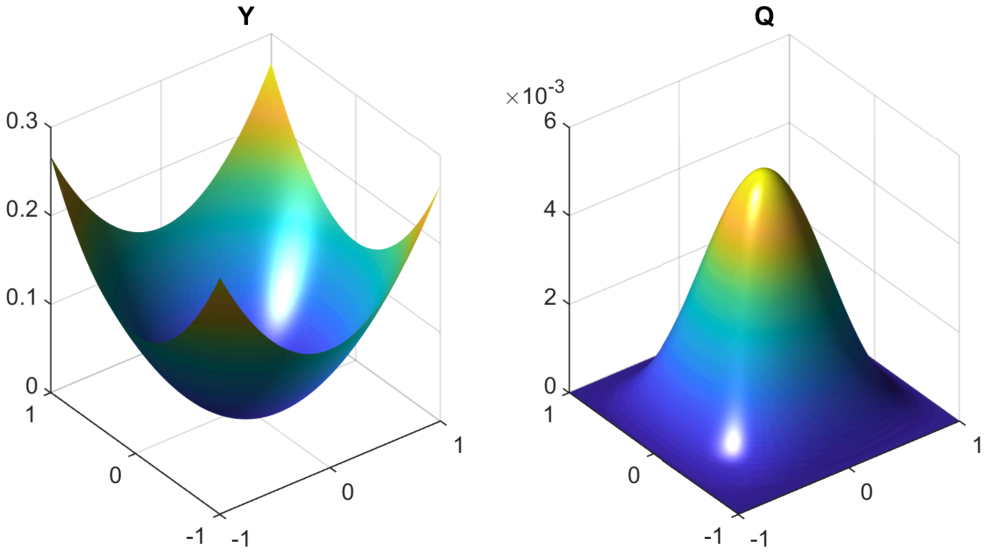

Fig. 3.6: Computed solution of the reaction–diffusion problem at time $t = 0.5$, using a finite difference space discretization with $n_x = 128$ spatial grid points.

to the imposition of Neumann boundary conditions) leads to worse error properties in the computed adjoint than an arbitrary initial guess; however, increasing the number of spatial grid points gives very high accuracy once again. We may see from Table 3.3 that, for sufficiently large n_x, doubling this value leads to an improved accuracy by a factor of roughly 4 once more, as is expected. We also observe that the computation time scales close to linearly with respect to number of spatial grid points, with the only nonlinearity arising from the factorization of the matrix G: this nonlinear scaling becomes visible for a smaller n_x than for the Schlögl problem, as G is a larger matrix for the reaction–diffusion problem, however all computation times are low considering the high complexity of the problem being solved.

A visualization of the computed solution components y and q at time $t = 0.5$ using a finite difference space discretization with $n_x = 128$ spatial grid points is shown in Figure 3.6. We conclude this section by highlighting that we can also solve this reaction–diffusion control example using a spectral method in space, and indeed our approach can in principle be applied to any space discretization, with all that is required being the matrices arising from the discretization of the linearized PDE operators at a given time-step. For instance, one may apply a finite element discretization in space, and take advantage of the finite element method's greater flexibility when it comes to grid and mesh structure.

4. Concluding remarks.

In this short paper, we have derived a Newton–Krylov method to allow the spectral-in-time solution of nonlinear time-dependent PDE-constrained optimization problems, suggested a preconditioner to be applied within this method, and validated our approach numerically on test problems relating to the optimal control of the Schlögl equation as well as a reaction–diffusion system. Thanks to the spectral time discretization, as opposed to more commonly used low-order time discretizations (such as backward Euler in [26, 27, 29]), our method allows for a fast and accurate solution of such problems with relatively low memory requirements (as the solution is only collocated at a small number of time nodes). The method is also amenable to a range of strategies for the space discretization, and to apply the method the user only needs to provide matrices arising from their preferred discretization of the linearized PDE operators at a given time-step. Possible extensions of this approach include the solution of more sophisticated systems of PDEs or integro-PDEs, problems with additional algebraic constraints on the state and control variables, and connecting with time-parallel methods for parabolic control problems, for instance the work presented in [10]. The derivation of more sophisticated preconditioning strategies and stopping criteria for the Newton–Krylov solver, taking account of the specific structures of the PDE operators involved, would also be of interest.

Acknowledgements. We are grateful to two anonymous referees for many insightful comments and suggestions that have improved this work. JWP gratefully acknowledges support from the Engineering and Physical Sciences Research Council (EPSRC) grant EP/S027785/1. SG and JWP acknowledge support from The Alan Turing Institute, EPSRC grant EP/N510129/1.

Appendix A. Test problems with analytic solutions.

Our first test problem involves the optimal control of the Schlögl equation, as given in (2.1). The continuous optimality conditions (2.3)–(2.5) are solved by the following functions:

$$
\begin{aligned}
y &= \alpha\left(\frac{4}{(d\pi^2-4)\beta}e^T - \frac{4}{d\pi^2\beta}e^t\right)\ \prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right),\\
p &= \alpha(e^T - e^t)\ \prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right), \qquad \left[c = \frac{1}{\beta}p\right]\\
\widehat{y} &= \alpha\left(\left[\frac{4}{(d\pi^2-4)\beta} + \frac{d\pi^2}{4} - 1\right]e^T + \left[2 - \frac{4}{d\pi^2\beta} - \frac{d\pi^2}{4}\right]e^t\right)\ \prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right)\\
&\quad + 3\alpha^3\left(\frac{4}{(d\pi^2-4)\beta}e^T - \frac{4}{d\pi^2\beta}e^t\right)^2 (e^T - e^t)\ \left[\prod_{k=1}^{d}\cos\left(\frac{\pi x_k}{2}\right)\right]^3,
\end{aligned}
$$

$$
\begin{aligned}
f &= \alpha^3 \left(\frac{4}{(d\pi^2-4)\beta} e^T - \frac{4}{d\pi^2\beta} e^t \right)^3 \left[\prod_{k=1}^{d} \cos\left(\frac{\pi x_k}{2}\right) \right]^3, \\
y_0 &= \alpha \left(\frac{4}{(d\pi^2-4)\beta} e^T - \frac{4}{d\pi^2\beta} \right) \prod_{k=1}^{d} \cos\left(\frac{\pi x_k}{2}\right), \\
h &= 0,
\end{aligned}
$$

where $d \in \{2,3\}$ is the dimension of the problem, solved on the space–time domain $(-1,1)^d \times (0,T)$, and $\alpha \neq 0$ denotes an arbitrary constant.

Our second test problem involves the reaction–diffusion control problem (2.2). The continuous optimality conditions (2.6)–(2.8) are solved by the following functions:

$$
\begin{aligned}
y &= \frac{1}{8}(e^T - e^t) \sum_{k=1}^{d} x_k^2, \\
z &= \frac{1}{8}(e^T - e^t), \\
p &= \frac{\beta_c D_1}{4}(e^T - e^t), \qquad \left[c = \frac{1}{\beta_c} p \quad \text{on } \partial\Omega \times (0,T) \right] \\
q &= \frac{\beta_c D_1}{4}(e^T - e^t) \prod_{k=1}^{d} \left[1 + \cos(\pi x_k)\right], \\
\widehat{y} &= \frac{1}{\beta_y} \Bigg\{ \frac{\beta_c D_1}{4} e^t + (e^T - e^t) \left[\frac{\beta_y}{8} \sum_{k=1}^{d} x_k^2 + \frac{k_1 \beta_c D_1}{4} \right] \\
&\quad + \frac{\beta_c D_1}{32}(e^T - e^t)^2 \left[\gamma_1 + \gamma_2 \prod_{k=1}^{d} \left[1 + \cos(\pi x_k)\right] \right] \Bigg\}, \\
\widehat{z} &= \frac{1}{\beta_z} \Bigg\{ \frac{\beta_c D_1}{4} e^t \prod_{k=1}^{d} \left[1 + \cos(\pi x_k)\right] + (e^T - e^t) \Bigg[\frac{\beta_z}{8} + \\
&\quad \frac{\pi^2 \beta_c D_1 D_2}{4} \sum_{k=1}^{d} \left[\cos(\pi x_k) \prod_{j=1, j\neq k}^{d} \left[1 + \cos(\pi x_j)\right] \right] + \frac{k_2 \beta_c D_1}{4} \prod_{k=1}^{d} \left[1 + \cos(\pi x_k)\right] \Bigg] \\
&\quad + \frac{\beta_c D_1}{32}(e^T - e^t)^2 \sum_{k=1}^{d} x_k^2 \left[\gamma_1 + \gamma_2 \prod_{j=1}^{d} \left[1 + \cos(\pi x_j)\right] \right] \Bigg\}, \\
f &= -\frac{1}{8} e^t \sum_{k=1}^{d} x_k^2 + (e^T - e^t) \left[-\frac{d D_1}{4} + \frac{k_1}{8} \sum_{k=1}^{d} x_k^2 \right] + \frac{\gamma_1}{64}(e^T - e^t)^2 \sum_{k=1}^{d} x_k^2, \\
g &= -\frac{1}{8} e^t + \frac{k_2}{8}(e^T - e^t) + \frac{\gamma_2}{64}(e^T - e^t)^2 \sum_{k=1}^{d} x_k^2, \\
y_0 &= \frac{1}{8}(e^T - 1) \sum_{k=1}^{d} x_k^2, \\
z_0 &= \frac{1}{8}(e^T - 1),
\end{aligned}
$$

with $d \in \{2,3\}$ again the dimension of the problem, solved on the space–time domain $(-1,1)^d \times (0,T)$.

REFERENCES

[1] S. R. Arridge. Optical tomography in medical imaging. *Inverse Probl.*, 15(2):R41–R93, 1999.

[2] W. Barthel, C. John, and F. Tröltzsch. Optimal boundary control of a system of reaction diffusion equations. *J. Appl. Math. Mech.*, 90(12):966–982, 2010.

[3] A. Borzì and V. Schulz. Multigrid methods for PDE optimization. *SIAM Rev.*, 51(2):361–395, 2009.

[4] R. Buchholz, H. Engel, E. Kammann, and F. Tröltzsch. On the optimal control of the Schlögl-model. *Comput. Optim. Appl.*, 56(1):153–185, 2013.

[5] M. Cheney, D. Isaacson, and J. C. Newell. Electrical impedance tomography. *SIAM Rev.*, 41(1):85–101, 1999.

[6] S. Dolgov and M. Stoll. Low-rank solution to an optimization problem constrained by the Navier–Stokes equations. *SIAM J. Sci. Comput.*, 39(1):A255–A280, 2017.

[7] T. A. Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide.* Pafnuty Publications, Oxford, 2014.

[8] A. Dutt, L. Greengard, and V. Rokhlin. Spectral deferred correction methods for ordinary differential equations. *BIT Numer. Math.*, 40(2):241–266, 2000.

[9] S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. *IMA J. Numer. Anal.*, 17(1):16–32, 1996.

[10] S. Götschel and M. L. Minion. An efficient parallel-in-time method for optimization with parabolic PDEs. *SIAM J. Sci. Comput.*, 41(3):C603–C626, 2019.

[11] R. Griesse and S. Volkwein. A primal-dual active set strategy for optimal boundary control of a nonlinear reaction-diffusion system. *SIAM J. Cont. Opt.*, 44(2):467–494, 2005.

[12] M. Gunzberger. *Perspectives in Flow Control and Optimization.* SIAM, 2010.

[13] M. D. Gunzburger, L. Hou, and T. P. Svobudny. Finite element approximations of an optimal control problem associated with the scalar Ginzburg-Landau equation. *Comput. Math. Appl.*, 21(2–3):123–132, 1991.

[14] S. Güttel and J. W. Pearson. A rational deferred correction approach to parabolic optimal control problems. *IMA J. Numer. Anal.*, 38(4):1861–1892, 2018.

[15] M. Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *J. Comput. Appl. Math.*, 173(1):169–198, 2005.

[16] M. Hinze, M. Köster, and S. Turek. A space-time multigrid method for optimal flow control. In *Constrained Optimization and Optimal Control for Partial Differential Equations*, volume 160 of *Internat. Ser. Numer. Math.*, pages 147–170. Springer Basel AG, 2012.

[17] J. Huang, J. Jia, and M. Minion. Accelerating the convergence of spectral deferred correction methods. *J. Comput. Phys.*, 214(2):633–656, 2006.

[18] J. Huang, J. Jia, and M. Minion. Arbitrary order Krylov deferred correction methods for differential algebraic equations. *J. Comput. Phys.*, 221(2):739–760, 2007.

[19] K. Ito and K. Kunisch. *Lagrange Multiplier Approach to Variational Problems and Applications.* Vol. 15 of Advances in Design and Control, SIAM, 2008.

[20] J. Jia and J. Huang. Krylov deferred correction accelerated method of lines transpose for parabolic problems. *J. Comput. Phys.*, 227(3):1739–1753, 2008.

[21] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method.* SIAM, 2003.

[22] D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, 193(2):357–397, 2004.

[23] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations.* Grundlehren der Mathematischen Wissenschaften, 1971.

[24] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *C. R. Math.*, 335(4):387–392, 2002.

[25] T. P. Mathew, M. Sarkis, and C. E. Schaerer. Analysis of block parareal preconditioners for parabolic optimal control problems. *SIAM J. Sci. Comput.*, 32(3):1180–1200, 2010.

[26] J. W. Pearson and M. Stoll. Fast iterative solution of reaction–diffusion control problems arising from chemical processes. *SIAM J. Sci. Comput.*, 35(5):B987–B1009, 2013.

[27] J. W. Pearson, M. Stoll, and A. J. Wathen. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *SIAM J. Matrix Anal. Appl.*, 33(4):1126–1152, 2012.

[28] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 7:856–869, 1986.

[29] M. Stoll and T. Breiten. A low-rank in time approach to PDE-constrained optimization. *SIAM J. Sci. Comput.*, 37(1):B1–B29, 2015.

[30] F. Tröltzsch. *Optimal Control of Partial Differential Equations – Theory, Methods and Applications.* Graduate Studies in Mathematics, Vol. 112. American Mathematical Society, 2010.

[31] M. Weiser. Faster SDC convergence on non-equidistant grids by DIRK sweeps. *BIT Numer. Math.*, 55(4):1219–1241, 2015.