Journal of Computational Physics 304 (2016) 27–45

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Fast solvers for optimal control problems from pattern formation

Martin Stoll [a], John W. Pearson [b,*], Philip K. Maini [c]

[a] Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany
[b] School of Mathematics, Statistics and Actuarial Science, University of Kent, Cornwallis Building (East), Canterbury, CT2 7NF, UK
[c] Wolfson Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Radcliffe Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, UK

ARTICLE INFO

Article history:
Received 24 March 2014
Received in revised form 5 July 2015
Accepted 1 October 2015
Available online 13 October 2015

Keywords:
PDE-constrained optimization
Reaction–diffusion
Pattern formation
Newton iteration
Preconditioning
Schur complement

ABSTRACT

The modeling of pattern formation in biological systems using various models of reaction–diffusion type has been an active research topic for many years. We here look at a parameter identification (or PDE-constrained optimization) problem where the Schnakenberg and Gierer–Meinhardt equations, two well-known pattern formation models, form the constraints to an objective function. Our main focus is on the efficient solution of the associated nonlinear programming problems via a Lagrange–Newton scheme. In particular we focus on the fast and robust solution of the resulting large linear systems, which are of saddle point form. We illustrate this by considering several two- and three-dimensional setups for both models. Additionally, we discuss an image-driven formulation that allows us to identify parameters of the model to match an observed quantity obtained from an image.

1. Introduction

One of the fundamental problems in developmental biology is to understand how spatial patterns, such as pigmentation patterns, skeletal structures, and so on, arise. In 1952, Alan Turing [42] proposed his theory of pattern formation in which he hypothesized that a system of chemicals, reacting and diffusing, could be driven unstable by diffusion, leading to spatial patterns (solutions which are steady in time but vary in space). He proposed that these chemical patterns, which he termed morphogen patterns, set up pre-patterns which would then be interpreted by cells in a concentration-dependent manner, leading to the patterns that we see.

These models have been applied to a very wide range of areas (see, for example, Murray [27]) and have been shown to exist in chemistry [6,30]. While their applicability to biology remains controversial, there are many examples which suggest that Turing systems may be underlying key patterning processes (see [2,8,40] for the most recent examples). Two important models which embody the essence of the original Turing model are the Gierer–Meinhardt [14] and Schnakenberg models [39] and it is upon these models which we focus.[1] In light of the fact that, to date, no Turing morphogens have

* Corresponding author.
E-mail addresses: stollm@mpi-magdeburg.mpg.de (M. Stoll), j.w.pearson@kent.ac.uk (J.W. Pearson), maini@maths.ox.ac.uk (P.K. Maini).

[1] Although the second model is commonly referred to as the Schnakenberg model, it was actually first proposed by Gierer and Meinhardt in [14] along with the model usually referenced as the Gierer–Meinhardt model – we therefore refer to the first and second models as '*GM*1' and '*GM*2' within our working.

http://dx.doi.org/10.1016/j.jcp.2015.10.006
0021-9991/

been unequivocally demonstrated, we do not have model parameter values so a key problem in mathematical biology is to determine parameters that give rise to certain observed patterns. It is this problem that the present study investigates.

More recently, an area in applied and numerical mathematics that has generated much research interest is that of PDE-constrained optimization problems (see [41] for an excellent introduction to this field). It has been found that one key application of such optimal control formulations is to find solutions to pattern formation problems [11,12], and so it is natural to explore this particular application here.

In this paper, we consider the numerical solution of optimal control (in this case parameter identification) formulations of these Turing models – in particular we wish to devise preconditioned iterative solvers for the matrix systems arising from the application of Newton and Gauss–Newton methods to the problems. The crucial aspect of the preconditioners is the utilization of saddle point theory to obtain effective approximations to the $(1,1)$-block and Schur complement of these matrix systems. The solvers incorporate aspects of iterative solution strategies developed by the first and second authors to tackle simpler optimal control problems in literature such as [32–35].

This paper is structured as follows. In Section 2 we introduce the Gierer–Meinhardt ($GM1$) and Schnakenberg ($GM2$) models that we consider, and outline the corresponding optimal control problems. In Section 3 we discuss the outer (Newton-type) iteration that we employ for these problems, and state the resulting matrix systems at each iteration. We then motivate and derive our preconditioning strategies in Section 4. In Section 5 we present numerical results to demonstrate the effectiveness of our approaches, and finally in Section 6 we make some concluding remarks.

2. A parameter identification problem

Parameter identification problems are crucial in determining the setup of a mathematical model, often given by a system of differential equations, that is best suited to describe measured data or an observed phenomenon. These problems are often posed as PDE-constrained optimization problems [20,41]. We here want to minimize an objective function of misfit type, i.e., the function is designed to penalize deviations of the function values from the observed or measured data. The particular form is given by [11,12]:

$$\begin{aligned}\mathcal{J}(u,v,a,b) = {} & \frac{\beta_1}{2}\,\|u(\mathbf{x},t)-\widehat{u}(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])} + \frac{\beta_2}{2}\,\|v(\mathbf{x},t)-\widehat{v}(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])} \\ & + \frac{\beta_{T,1}}{2}\,\|u(\mathbf{x},T)-\widehat{u}_T(\mathbf{x})\|^2_{L_2(\Omega)} + \frac{\beta_{T,2}}{2}\,\|v(\mathbf{x},T)-\widehat{v}_T(\mathbf{x})\|^2_{L_2(\Omega)} \\ & + \frac{\nu_1}{2}\,\|a(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])} + \frac{\nu_2}{2}\,\|b(\mathbf{x},t)\|^2_{L_2(\Omega\times[0,T])}\,, \end{aligned} \tag{2.1}$$

where u, v are the *state variables*, and a, b the *control variables*, in our formulation. This is to say we wish to ensure that the state variables are as close as possible in the L_2-norm to some observed or desired states $\widehat{u}$, $\widehat{v}$, $\widehat{u}_T$, $\widehat{v}_T$, but at the same time penalize the enforcement of controls that have large magnitudes in this norm. The space–time domain on which this problem is considered is given by $\Omega\times[0,T]$, where $\Omega\subset\mathbb{R}^d$, $d\in\{2,3\}$.

Our goal is to identify the parameters of classical pattern formation equations such that the resulting optimal parameters allow the use of these models for real-world data. We here use models of reaction–diffusion type typically exploited to generate patterns seen in biological systems. The two formulations we consider are the $GM1$ model [14,27]:

$$\begin{aligned} u_t - D_u\Delta u - \frac{ru^2}{v} + au &= r, \quad \text{on } \Omega\times[0,T], \\ v_t - D_v\Delta v - ru^2 + bv &= 0, \quad \text{on } \Omega\times[0,T], \\ u(\mathbf{x},0)=u_0(\mathbf{x}), \quad v(\mathbf{x},0)&=v_0(\mathbf{x}), \quad \text{on } \Omega, \\ \frac{\partial u}{\partial \nu} = \frac{\partial v}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned} \tag{2.2}$$

and the $GM2$ model [14,27,39]:

$$\begin{aligned} u_t - D_u\Delta u + \gamma(u-u^2v) - \gamma a &= 0, \quad \text{on } \Omega\times[0,T], \\ v_t - D_v\Delta v + \gamma u^2v - \gamma b &= 0, \quad \text{on } \Omega\times[0,T], \\ u(\mathbf{x},0)=u_0(\mathbf{x}), \quad v(\mathbf{x},0)&=v_0(\mathbf{x}), \quad \text{on } \Omega, \\ \frac{\partial u}{\partial \nu} = \frac{\partial v}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned} \tag{2.3}$$

where r and γ are non-negative parameters involved in the respective models.

Both the $GM1$ and $GM2$ formulations are models of reaction–diffusion processes occurring in many types of pattern formation and morphogenesis processes [14,27,39]. The $GM1$ model relates to an "activator–inhibitor" system, whereas the $GM2$ model represents substrate-depletion. Within both models the variables u and v, the state variables in our formulation, represent the concentrations of chemical products. The parameters D_u and D_v denote the diffusion coefficients – typically

it is assumed that v diffuses faster than u, so $D_u < D_v$ [14]. The (given) parameters r and γ are positive: the value r in the *GM*1 model denotes the (small) production rate of the activator [14], and the parameter γ in the *GM*2 model is the *Hill coefficient*, which describes the cooperativity within a binding process. The variables a and b, the control variables in our problem, represent the rates of decay for u and v, respectively. The initial conditions u_0 and v_0 are known.

Throughout the remainder of this article we consider the minimization of the cost functional (2.1), with PDE constraints taking the form of the *GM*1 model or the *GM*2 model. PDE-constrained optimization problems of similar form have been considered in the literature, such as in [11,12]. When solving these problems we consider a range of values of the parameters involved in the PDE models, as well as variations in the coefficients β_i, $\beta_{T,i}$, ν_i $(i = 1, 2)$ within $\mathcal{J}(u, v, a, b)$. One typically chooses β_i (and frequently $\beta_{T,i}$) to be larger than ν_i in order for the states to closely resemble the desired states, due to the input of control not being severely penalized – the case of larger ν_i relates to the control variables being very small, with the state variables therefore failing to match the desired states as closely. However it is possible to formulate these problems using a wide range of parameters, and so within the numerical results of Section 5 we vary the parameter setup to demonstrate the robustness of our methods.

Note that the PDE constraints, for either model (2.2) or (2.3), are nonlinear, and as a result the optimization problem $\min_{(u,v,a,b)} \mathcal{J}(u, v, a, b)$ is itself nonlinear (although the cost functional is quadratic). To solve this problem we are therefore required to apply nonlinear programming [29] algorithms. Many of these are generalizations of Newton's method [29]. We here focus on a Lagrange–Newton (or basic SQP) scheme and a Gauss–Newton method. At the heart of both approaches lies the solution of large linear systems, which are often in saddle point form [4,9], that represent the Hessian or an approximation to it. In order to be able to solve these large linear systems we need to employ iterative solvers [9,37], which can be accelerated using effective preconditioners.

3. Nonlinear programming

A standard way of how to proceed with the above nonlinear program is to consider a classical Lagrangian approach [41]. In our case, with a nonlinear constraint, we apply a nonlinear solver to the first order conditions. We hence start by deriving the first order conditions, or Karush–Kuhn–Tucker conditions, of the Lagrangian

$$\mathcal{L}(u, v, a, b, p, q) = \mathcal{J}(u, v, a, b) + (p, \mathcal{R}_1(u, v, a, b)) + (q, \mathcal{R}_2(u, v, a, b)),$$

where $\mathcal{R}_1(u, v, a, b)$, $\mathcal{R}_2(u, v, a, b)$ represent the first two equations of both *GM*1 and *GM*2 models, and p, q denote the *adjoint variables* (or *Lagrange multipliers*). Note that for convenience our Lagrangian ignores the boundary and initial conditions. In general form the first order conditions are given by

$$\begin{aligned} &\mathcal{L}_u = 0, \quad \mathcal{L}_v = 0, \\ &\mathcal{L}_a = 0, \quad \mathcal{L}_b = 0, \\ &\mathcal{L}_p = 0, \quad \mathcal{L}_q = 0. \end{aligned}$$

The equations are in general nonlinear and a standard Newton method can be applied to them to give the following Lagrange–Newton or SQP scheme:

$$\begin{bmatrix} \mathcal{L}_{uu} & \mathcal{L}_{uv} & \mathcal{L}_{ua} & \mathcal{L}_{ub} & \mathcal{L}_{up} & \mathcal{L}_{uq} \\ \mathcal{L}_{vu} & \mathcal{L}_{vv} & \mathcal{L}_{va} & \mathcal{L}_{vb} & \mathcal{L}_{vp} & \mathcal{L}_{vq} \\ \mathcal{L}_{au} & \mathcal{L}_{av} & \mathcal{L}_{aa} & \mathcal{L}_{ab} & \mathcal{L}_{ap} & \mathcal{L}_{aq} \\ \mathcal{L}_{bu} & \mathcal{L}_{bv} & \mathcal{L}_{ba} & \mathcal{L}_{bb} & \mathcal{L}_{bp} & \mathcal{L}_{bq} \\ \mathcal{L}_{pu} & \mathcal{L}_{pv} & \mathcal{L}_{pa} & \mathcal{L}_{pb} & \mathcal{L}_{pp} & \mathcal{L}_{pq} \\ \mathcal{L}_{qu} & \mathcal{L}_{qv} & \mathcal{L}_{qa} & \mathcal{L}_{qb} & \mathcal{L}_{qp} & \mathcal{L}_{qq} \end{bmatrix} \begin{bmatrix} \delta u \\ \delta v \\ \delta a \\ \delta b \\ \delta p \\ \delta q \end{bmatrix} = - \begin{bmatrix} \mathcal{L}_u \\ \mathcal{L}_v \\ \mathcal{L}_a \\ \mathcal{L}_b \\ \mathcal{L}_p \\ \mathcal{L}_q \end{bmatrix}, \tag{3.1}$$

where δu, δv, δa, δb, δp, δq denote the Newton updates for u, v, a, b, p, q.

Note that our formulation does not include any globalization techniques such as trust region or line search approaches [28]. In order for the optimization algorithm to converge these should in general be incorporated. As our focus here is on large-scale linear systems we do not focus on these approaches now. At this stage we simply state the systems obtained for both *GM*1 and *GM*2 models and refer the interested reader to Appendix A, where all quantities are derived in detail. The system given in (3.1) represents the most general Newton system but it is often possible to only use approximations to this system. The Gauss–Newton method [16] is often used as the corresponding system matrix in (3.1) – this ignores the mixed derivatives with respect to the primal variables, i.e. with the system matrix given by

$$\begin{bmatrix} \widehat{\mathcal{L}}_{uu} & 0 & 0 & 0 & \mathcal{L}_{up} & \mathcal{L}_{uq} \\ 0 & \widehat{\mathcal{L}}_{vv} & 0 & 0 & \mathcal{L}_{vp} & \mathcal{L}_{vq} \\ 0 & 0 & \widehat{\mathcal{L}}_{aa} & 0 & \mathcal{L}_{ap} & \mathcal{L}_{aq} \\ 0 & 0 & 0 & \widehat{\mathcal{L}}_{bb} & \mathcal{L}_{bp} & \mathcal{L}_{bq} \\ \mathcal{L}_{pu} & \mathcal{L}_{pv} & \mathcal{L}_{pa} & \mathcal{L}_{pb} & \mathcal{L}_{pp} & \mathcal{L}_{pq} \\ \mathcal{L}_{qu} & \mathcal{L}_{qv} & \mathcal{L}_{qa} & \mathcal{L}_{qb} & \mathcal{L}_{qp} & \mathcal{L}_{qq} \end{bmatrix}, \tag{3.2}$$

where the matrices denoted by $\widehat{\mathcal{L}}_{:,:}$ do not contain second derivative information (see [16,29] for more details). Additionally, to derive the infinite-dimensional Newton system we discretize the resulting equations using finite elements in space and a backward Euler scheme in time. The resulting system for the $GM1$ model is given by

$$\underbrace{\begin{bmatrix} \mathbf{A}_{u,GM1} & -2\tau r\mathbf{M}_{up/v^2} & -\tau\mathbf{M}_p & 0 & -\mathbf{L}_{u,GM1}^T & 2\tau r\mathbf{M}_u \\ -2\tau r\mathbf{M}_{up/v^2} & \mathbf{A}_{v,GM1} & 0 & -\tau\mathbf{M}_q & -\tau r\mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T \\ -\tau\mathbf{M}_p & 0 & \tau\nu_1\mathbf{M} & 0 & -\tau\mathbf{M}_u & 0 \\ 0 & -\tau\mathbf{M}_q & 0 & \tau\nu_2\mathbf{M} & 0 & -\tau\mathbf{M}_v \\ -\mathbf{L}_{u,GM1} & -\tau r\mathbf{M}_{u^2/v^2} & -\tau\mathbf{M}_u & 0 & 0 & 0 \\ 2\tau r\mathbf{M}_u & -\mathbf{L}_{v,GM1} & 0 & -\tau\mathbf{M}_v & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \boldsymbol{\delta u} \\ \boldsymbol{\delta v} \\ \boldsymbol{\delta a} \\ \boldsymbol{\delta b} \\ \boldsymbol{\delta p} \\ \boldsymbol{\delta q} \end{bmatrix} = \mathbf{f},$$

where

$$\mathbf{A}_{u,GM1} = \tau\beta_1\mathbf{M} + \beta_{T,1}\mathbf{M}_T + 2\tau r\mathbf{M}_{p/v} + 2\tau r\mathbf{M}_q,$$

$$\mathbf{A}_{v,GM1} = \tau\beta_2\mathbf{M} + \beta_{T,2}\mathbf{M}_T + 2\tau r\mathbf{M}_{u^2p/v^3},$$

$$\mathbf{L}_{u,GM1} = \mathbf{M}_E + \tau D_u\mathbf{K} - 2\tau r\mathbf{M}_{u/v} + \tau\mathbf{M}_a,$$

$$\mathbf{L}_{v,GM1} = \mathbf{M}_E + \tau D_v\mathbf{K} + \tau\mathbf{M}_b.$$

Note that M and K denote standard finite element mass and stiffness matrices, respectively. Here the matrices

$$\mathbf{M}_E := \begin{bmatrix} M & & & & \\ -M & M & & & \\ & -M & M & & \\ & & \ddots & \ddots & \\ & & & -M & M \end{bmatrix}, \quad \mathbf{M}_T := \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & \ddots & & \\ & & & 0 & \\ & & & & M \end{bmatrix},$$

correspond to, respectively, the time-stepping scheme used, and the values at the final time $t = T$. All other mass matrices $\mathbf{M}_\psi = \text{blkdiag}(M_\psi, \ldots, M_\psi)$ are obtained from evaluating integrals of the form $[M_\psi]_{ij} = \int \psi\phi_i\phi_j$ for each matrix entry and for every time-step, where ϕ_i denote the finite element basis functions used (see the group finite element method in [23]). Furthermore, the matrix $\mathbf{K} = \text{blkdiag}(K, \ldots, K)$. The parameter τ denotes the (constant) time-step used. The vector $\mathbf{f}$ is the discrete representation at each Newton step of the following vector function:

$$\begin{bmatrix} \beta_1\int(\widehat{u} - \bar{u}) + \int(-\bar{p}_t - D_u\Delta\bar{p} - 2r\frac{\bar{u}}{\bar{v}}\bar{p} + \bar{a}\bar{p} - 2r\bar{u}\bar{q}) \\ \beta_2\int(\widehat{v} - \bar{v}) + \int(-\bar{q}_t - D_v\Delta\bar{q} + r\frac{\bar{u}^2}{\bar{v}^2}\bar{p} + \bar{b}\bar{q}) \\ \int(\bar{u}\bar{p} - \nu_1\bar{a}) \\ \int(\bar{v}\bar{q} - \nu_2\bar{b}) \\ \int(\bar{u}_t - D_u\Delta\bar{u} - \frac{r\bar{u}^2}{\bar{v}} + \bar{a}\bar{u} - r) \\ \int(\bar{v}_t - D_v\Delta\bar{v} - r\bar{u}^2 + \bar{b}\bar{v}) \end{bmatrix},$$

where $\bar{u}$, $\bar{v}$, $\bar{a}$, $\bar{b}$, $\bar{p}$, $\bar{q}$ denote the previous Newton iterates for u, v, a, b, p, q.

The Gauss–Newton type matrix for this problem now becomes

$$\underbrace{\begin{bmatrix} \beta_1\tau\mathbf{M} & 0 & 0 & 0 & -\mathbf{L}_{u,GM1}^T & 2\tau r\mathbf{M}_u \\ 0 & \beta_2\tau\mathbf{M} & 0 & 0 & -\tau r\mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T \\ 0 & 0 & \nu_1\tau\mathbf{M} & 0 & -\tau\mathbf{M}_u & 0 \\ 0 & 0 & 0 & \nu_2\tau\mathbf{M} & 0 & -\tau\mathbf{M}_v \\ -\mathbf{L}_{u,GM1} & -\tau r\mathbf{M}_{u^2/v^2} & -\tau\mathbf{M}_u & 0 & 0 & 0 \\ 2\tau r\mathbf{M}_u & -\mathbf{L}_{v,GM1} & 0 & -\tau\mathbf{M}_v & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \boldsymbol{\delta u} \\ \boldsymbol{\delta v} \\ \boldsymbol{\delta a} \\ \boldsymbol{\delta b} \\ \boldsymbol{\delta p} \\ \boldsymbol{\delta q} \end{bmatrix} = \mathbf{f}_{GN},$$

with all matrices as previously defined (see [5,16] for details on the Gauss–Newton matrix structure). We consider this matrix system as well as the "pure Newton" formulation of the $GM1$ model, as we find that the Gauss–Newton method often results in favorable properties from an iterative solver point-of-view.

Moving on to the *GM*2 model, Appendix A reveals the following structure of the Newton system:

$$\underbrace{\begin{bmatrix} \mathbf{A}_{u,GM2} & -2\tau\gamma\mathbf{M}_{u(q-p)} & 0 & 0 & -\mathbf{L}_{u,GM2}^T & -2\tau\gamma\mathbf{M}_{uv} \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} & 0 & 0 & \tau\gamma\mathbf{M}_{u^2} & -\mathbf{L}_{v,GM2}^T \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 & \tau\gamma\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} & 0 & \tau\gamma\mathbf{M} \\ -\mathbf{L}_{u,GM2} & \tau\gamma\mathbf{M}_{u^2} & \tau\gamma\mathbf{M} & 0 & 0 & 0 \\ -2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} & 0 & \tau\gamma\mathbf{M} & 0 & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \boldsymbol{\delta u} \\ \boldsymbol{\delta v} \\ \boldsymbol{\delta a} \\ \boldsymbol{\delta b} \\ \boldsymbol{\delta p} \\ \boldsymbol{\delta q} \end{bmatrix} = \mathbf{g},$$

with

$$\begin{aligned} \mathbf{A}_{u,GM2} &= \tau\beta_1\mathbf{M} + \beta_{T,1}\mathbf{M}_T + 2\tau\gamma\mathbf{M}_{v(q-p)}, \\ \mathbf{A}_{v,GM2} &= \tau\beta_2\mathbf{M} + \beta_{T,2}\mathbf{M}_T, \\ \mathbf{L}_{u,GM2} &= \mathbf{M}_E + \tau D_u\mathbf{K} + \tau\gamma\mathbf{M} - 2\tau\gamma\mathbf{M}_{uv}, \\ \mathbf{L}_{v,GM2} &= \mathbf{M}_E + \tau D_v\mathbf{K} + \tau\gamma\mathbf{M}_{u^2}, \end{aligned}$$

and $\mathbf{g}$ the discrete representation of the vector function:

$$\begin{bmatrix} \beta_1 \int (\widehat{u} - \bar{u}) + \int (-\bar{p}_t - D_u\Delta\bar{p} + 2\gamma\bar{u}\bar{v}(\bar{q} - \bar{p}) + \gamma\bar{p}) \\ \beta_2 \int (\widehat{v} - \bar{v}) + \int (-\bar{q}_t - D_v\Delta\bar{q} + \gamma\bar{u}^2(\bar{q} - \bar{p})) \\ -\int (\nu_1\bar{a} + \gamma\bar{p}) \\ -\int (\nu_2\bar{b} + \gamma\bar{q}) \\ \int (\bar{u}_t - D_u\Delta\bar{u} + \gamma(\bar{u} - \bar{u}^2\bar{v}) - \gamma\bar{a}) \\ \int (\bar{v}_t - D_v\Delta\bar{v} + \gamma\bar{u}^2\bar{v} - \gamma\bar{b}) \end{bmatrix}.$$

The main challenge is now the numerical evaluation of the discretized problems. As we here opt for an all-at-once approach where we discretize in space and time and then solve the resulting linear system for all time steps simultaneously, we need to be able to perform this operation efficiently. Similar approaches have recently been considered in [33]. The goal of the next section is to introduce the appropriate methodology.

4. Preconditioning and Krylov subspace solver

The solution of large-scale linear systems of the *saddle point* form:

$$\mathcal{A}\mathbf{x} = \mathbf{b}, \qquad \text{with} \quad \mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}, \tag{4.1}$$

where $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times m}$ (with $m \geq n$), $C \in \mathbb{R}^{n\times n}$, is a topic of major interest within the numerical analysis community [4,9]. Due to the vast dimensionality of the systems derived earlier we cannot use factorization-based approaches [7]. We hence employ a Krylov subspace method [37] where we construct a Krylov subspace of the form

$$\mathcal{K}_k(\mathcal{A}, \mathbf{r}_0) = \text{span}\left\{\mathbf{r}_0, \mathcal{A}\mathbf{r}_0, \mathcal{A}^2\mathbf{r}_0, \ldots, \mathcal{A}^k\mathbf{r}_0\right\},$$

within which we seek an approximation to the solution of a given linear system. These methods are cheap as they only require multiplication with the system matrix, which is often possible to perform in a matrix-free way, that is to say the matrix $\mathcal{A}$ can be a black-box that only computes $\mathcal{A}\mathbf{w}$ for some vector $\mathbf{w}$. As a rule-of-thumb (rigorously in the case of symmetric $\mathcal{A}$) the eigenvalues of $\mathcal{A}$ determine how fast the approximate solution converges towards the true solution.

It is very well recognized that the eigenvalues of a saddle point matrix $\mathcal{A}$ depend strongly on the eigenvalues of the individual blocks A, B, C.[2] Clearly the eigenvalues of the individual matrices within these blocks depend on the mesh-size and time-step used, as well as all the other parameters describing the PDE and the objective function. [To give one example, the eigenvalues of K are contained within $[c_1h^d, c_2h^{d-2}]$ for constants c_1, c_2, where a constant mesh-size h is taken.] As a result, for our problem, the eigenvalues of $\mathcal{A}$ depend on these problem parameters. The convergence of an iterative method

[2] For illustrative purposes, a fundamental result [36] is as follows: if A is symmetric positive definite, B is full rank, and $C = 0$, the eigenvalues of $\mathcal{A}$ are contained within the intervals

$$\lambda(\mathcal{A}) \in \left[\frac{1}{2}\left(\mu_m - \sqrt{\mu_m^2 + 4\sigma_1^2}\right), \frac{1}{2}\left(\mu_1 - \sqrt{\mu_1^2 + 4\sigma_n^2}\right)\right] \cup \left[\mu_m, \frac{1}{2}\left(\mu_1 + \sqrt{\mu_1^2 + 4\sigma_1^2}\right)\right],$$

where μ_1, μ_m are the largest and smallest eigenvalues of A, and σ_1, σ_n denote the largest and smallest singular values of B.

applied directly to $\mathcal{A}$ can therefore be prohibitively slow, especially for large problems where h and τ are small. Our goal is therefore to find a preconditioning matrix $\mathcal{P}$ such that we can solve the equivalent preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b},$$

and $\mathcal{P}$ captures the properties of $\mathcal{A}$ well. If this can be achieved, the dependence of eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ on the problem parameters can be mitigated, and in the best case removed.

For a saddle point problem of the form (4.1), this is typically achieved by preconditioners of the form

$$\mathcal{P} = \begin{bmatrix} \widetilde{A} & 0 \\ 0 & \widetilde{S} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \widetilde{A} & 0 \\ B & -\widetilde{S} \end{bmatrix}, \tag{4.2}$$

where $\widetilde{A}$ approximates the $(1,1)$-block A of the saddle point matrix $\mathcal{A}$, and $\widetilde{S}$ approximates the (negative) Schur complement $S := C + BA^{-1}B^T$. This is motivated by results obtained in [22,26] where it is shown that the exact preconditioners $\widetilde{A} = A$ and $\widetilde{S} = S$ lead to a very small number of eigenvalues for the preconditioned system, and hence iteration numbers. The choice of the outer Krylov subspace solver typically depends on the nature of the system matrix and the preconditioner. For symmetric indefinite systems such as the ones presented here we usually choose MINRES [31] based on a three-term recurrence relation. However as MINRES typically requires a symmetric positive definite preconditioner, in the case of an indefinite preconditioner $\mathcal{P}$ we cannot use this method. We then need to apply a nonsymmetric solver of which there exist many, and it is not obvious which of them is best suited to any particular problem. Our rule-of-thumb is that if one carefully designs a preconditioner such that the eigenvalues of the preconditioned system are tightly clustered (or are contained within a small number of clusters), many different solvers perform in a fairly similar way. For simplicity we here choose BICG [10], which is the extension of CG [17] to nonsymmetric problems and is based on the nonsymmetric Lanczos process [15].

4.1. *GM*1 model

We now wish to derive preconditioners for all of the above linear systems. When examining the *GM*1 model using a Newton method the matrix $\mathcal{A}$ is written in the form of the saddle point system (4.1), with

$$A = \begin{bmatrix} \mathbf{A}_{u,GM1} & -2\tau r\mathbf{M}_{up/v^2} & -\tau\mathbf{M}_p & 0 \\ -2\tau r\mathbf{M}_{up/v^2} & \mathbf{A}_{v,GM1} & 0 & -\tau\mathbf{M}_q \\ -\tau\mathbf{M}_p & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & -\tau\mathbf{M}_q & 0 & \tau\nu_2\mathbf{M} \end{bmatrix},$$

$$B = \begin{bmatrix} -\mathbf{L}_{u,GM1} & -\tau r\mathbf{M}_{u^2/v^2} & -\tau\mathbf{M}_u & 0 \\ 2\tau r\mathbf{M}_u & -\mathbf{L}_{v,GM1} & 0 & -\tau\mathbf{M}_v \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Consider first approximating the matrix A. We observe that its block structure means we can also write it in saddle point type form. The matrix blkdiag$(\tau\nu_1\mathbf{M}, \tau\nu_2\mathbf{M})$ is comparatively straightforward to work with, as it is a block diagonal matrix consisting solely of mass matrices, so we may devise saddle point approximations with this as the leading block, i.e.,

$$\widetilde{A} = \begin{bmatrix} \widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & \widetilde{\mathbf{A}}_2 & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} -\widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & -\widetilde{\mathbf{A}}_2 & 0 & 0 \\ -\tau\mathbf{M}_p & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & -\tau\mathbf{M}_q & 0 & \tau\nu_2\mathbf{M} \end{bmatrix}, \tag{4.3}$$

for suitable choices of $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$. To make these selections, we seek to approximate the Schur complement of A, that is:

$$\begin{bmatrix} \widetilde{\mathbf{A}}_1 & 0 \\ 0 & \widetilde{\mathbf{A}}_2 \end{bmatrix} \approx \underbrace{\begin{bmatrix} \mathbf{A}_{u,GM1} & -2\tau r\mathbf{M}_{up/v^2} \\ -2\tau r\mathbf{M}_{up/v^2} & \mathbf{A}_{v,GM1} \end{bmatrix}}_{=:\widetilde{\mathbf{A}}_{(1,2)}} - \begin{bmatrix} \tau\nu_1^{-1}\mathbf{M}_p\mathbf{M}^{-1}\mathbf{M}_p & 0 \\ 0 & \tau\nu_2^{-1}\mathbf{M}_q\mathbf{M}^{-1}\mathbf{M}_q \end{bmatrix}.$$

We then utilize the heuristic of approximating $\widetilde{\mathbf{A}}_{(1,2)}$ by its own (block diagonal) saddle point approximation, leading to the following candidates for $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$:

$$\widetilde{\mathbf{A}}_1 = \mathbf{A}_{u,GM1} - (2\tau r)^2\mathbf{M}_{up/v^2}\mathbf{A}_{v,GM1}^{-1}\mathbf{M}_{up/v^2} - \tau\nu_1^{-1}\mathbf{M}_p\mathbf{M}^{-1}\mathbf{M}_p,$$
$$\widetilde{\mathbf{A}}_2 = \mathbf{A}_{v,GM1} - \tau\nu_2^{-1}\mathbf{M}_q\mathbf{M}^{-1}\mathbf{M}_q,$$

which we apply within (4.3). Within our implementation we replace the matrix $\mathbf{M}^{-1}$ with diagonal approximations, in order for the application of our solver to be computationally feasible.

Turning our attention now to the Schur complement of the entire matrix $\mathcal{A}$:

$$S = \begin{bmatrix} -\mathbf{L}_{u,GM1} & -\tau r \mathbf{M}_{u^2/v^2} \\ 2\tau r \mathbf{M}_u & -\mathbf{L}_{v,GM1} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} -\mathbf{L}_{u,GM1}^T & 2\tau r \mathbf{M}_u \\ -\tau r \mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T \end{bmatrix} + \tau \begin{bmatrix} \nu_1^{-1} \mathbf{M}_u \mathbf{M}^{-1} \mathbf{M}_u & 0 \\ 0 & \nu_2^{-1} \mathbf{M}_v \mathbf{M}^{-1} \mathbf{M}_v \end{bmatrix},$$

we construct an approximation

$$\widetilde{S} = \begin{bmatrix} -\mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)} & \tau r \mathbf{M}_{u^2/v^2} \\ -2\tau r \mathbf{M}_u & -\mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} -\mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)} & -2\tau r \mathbf{M}_u \\ \tau r \mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix},$$

for suitably chosen matrices $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$ and $\widehat{\mathbf{M}}_2^{(2)}$. To do this we apply a 'matching strategy', where we seek an additional (outer) term of the Schur complement approximation to match the second term of the exact Schur complement,[3] as follows:

$$\begin{bmatrix} \widehat{\mathbf{M}}_1^{(1)} & 0 \\ 0 & \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} \widehat{\mathbf{M}}_1^{(2)} & 0 \\ 0 & \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix} \approx \tau \begin{bmatrix} \nu_1^{-1} \mathbf{M}_u \mathbf{M}^{-1} \mathbf{M}_u & 0 \\ 0 & \nu_2^{-1} \mathbf{M}_v \mathbf{M}^{-1} \mathbf{M}_v \end{bmatrix}.$$

By examining the diagonal blocks of $\widetilde{\mathbf{A}}_{(1,2)}^{-1}$, we see that this strategy motivates the approximations:

$$\widehat{\mathbf{M}}_1^{(1)} \widetilde{\mathbf{A}}_1^{-1} \widehat{\mathbf{M}}_1^{(2)} \approx \frac{\tau}{\nu_1} \mathbf{M}_u \mathbf{M}^{-1} \mathbf{M}_u, \qquad \widetilde{\mathbf{A}}_1 := \mathbf{A}_{u,GM1} - (2\tau r)^2 \mathbf{M}_{up/v^2} \mathbf{A}_{v,GM1}^{-1} \mathbf{M}_{up/v^2},$$
$$\widehat{\mathbf{M}}_2^{(1)} \widetilde{\mathbf{A}}_2^{-1} \widehat{\mathbf{M}}_2^{(2)} \approx \frac{\tau}{\nu_2} \mathbf{M}_v \mathbf{M}^{-1} \mathbf{M}_v, \qquad \widetilde{\mathbf{A}}_2 := \mathbf{A}_{v,GM1} - (2\tau r)^2 \mathbf{M}_{up/v^2} \mathbf{A}_{u,GM1}^{-1} \mathbf{M}_{up/v^2}.$$

To achieve such an approximation, we again recommend selecting diagonal matrices $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$ and $\widehat{\mathbf{M}}_2^{(2)}$, with diagonal entries given by

$$\left[\widehat{\mathbf{M}}_1^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}} \, [\mathbf{M}_u]_{jj} \cdot [\mathbf{M}]_{jj}^{-1/2} \cdot \left|\left[\widetilde{\mathbf{A}}_1\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_1^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}} \, [\mathbf{M}]_{jj}^{-1/2} \cdot [\mathbf{M}_u]_{jj},$$
$$\left[\widehat{\mathbf{M}}_2^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}} \, [\mathbf{M}_v]_{jj} \cdot [\mathbf{M}]_{jj}^{-1/2} \cdot \left|\left[\widetilde{\mathbf{A}}_2\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_2^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}} \, [\mathbf{M}]_{jj}^{-1/2} \cdot [\mathbf{M}_v]_{jj}.$$

Now, for any practical method, we are only interested in the inverse of the Schur complement approximation $\widetilde{S}$. We therefore evaluate the inverse of the first and last block using a fixed number of steps of an Uzawa method [37] with preconditioners

$$\text{blkdiag}\left(\left(-\mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)}\right)_{\text{AMG}}, \left(-\mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)}\right)_{\text{AMG}}\right) \quad \text{or}$$
$$\text{blkdiag}\left(\left(-\mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)}\right)_{\text{AMG}}, \left(-\mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)}\right)_{\text{AMG}}\right),$$

where $(\cdot)_{\text{AMG}}$ denotes the application of an algebraic multigrid method to the relevant matrix.

For the Gauss–Newton case the derivation of the preconditioners is more straightforward. The approximation of the Hessian is typically not as good as in the Newton setting but the Gauss–Newton matrices are easier to handle from a preconditioning viewpoint. To approximate A we write

$$\begin{bmatrix} \beta_1 \tau \mathbf{M} & 0 & 0 & 0 \\ 0 & \beta_2 \tau \mathbf{M} & 0 & 0 \\ 0 & 0 & \nu_1 \tau \mathbf{M} & 0 \\ 0 & 0 & 0 & \nu_2 \tau \mathbf{M} \end{bmatrix} \approx \begin{bmatrix} \beta_1 \tau \widetilde{\mathbf{M}} & 0 & 0 & 0 \\ 0 & \beta_2 \tau \widetilde{\mathbf{M}} & 0 & 0 \\ 0 & 0 & \nu_1 \tau \widetilde{\mathbf{M}} & 0 \\ 0 & 0 & 0 & \nu_2 \tau \widetilde{\mathbf{M}} \end{bmatrix} =: \widetilde{A},$$

where $\widetilde{\mathbf{M}}$ is equal to $\mathbf{M}$ for (diagonal) lumped mass matrices. If consistent mass matrices are used instead, some approximation such as the application of Chebyshev semi-iteration [43] is chosen. The inverse of the Schur complement approximation

$$\widetilde{S} := \begin{bmatrix} -\mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)} & \tau r \mathbf{M}_{u^2/v^2} \\ -2\tau r \mathbf{M}_u & -\mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \widetilde{\mathbf{A}}_{(1,2)}^{-1} \begin{bmatrix} -\mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)} & -2\tau r \mathbf{M}_u \\ \tau r \mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix},$$

[3] This matching strategy was originally developed by the authors [33–35] to generate approximations for more fundamental PDE-constrained optimization problems. Considering the Poisson control problem, for instance, the Schur complement and its approximation take the form [34]:

$$S = K M^{-1} K + \frac{1}{\beta} M, \qquad \widetilde{S} = \left(K + \frac{1}{\sqrt{\beta}} M\right) M^{-1} \left(K + \frac{1}{\sqrt{\beta}} M\right),$$

whereupon it can be shown that the eigenvalues of $\widetilde{S}^{-1} S$ are contained within $[\frac{1}{2}, 1]$, independently of problem parameters and matrix dimension. This bound can be proved due to the comparatively simple structures of the matrices involved, and thus cannot be replicated for the complex systems studied here, however we have found this strategy to be very effective for a range of PDE-constrained optimization problems.

where here $\widetilde{\mathbf{A}}_{(1,2)} = \text{blkdiag}(\beta_1\tau\mathbf{M}, \beta_2\tau\mathbf{M})$, $\widehat{\mathbf{M}}_1^{(1)} = \widehat{\mathbf{M}}_1^{(2)} = \tau\sqrt{\frac{\beta_1}{\nu_1}}\,\mathbf{M}_u$, and $\widehat{\mathbf{M}}_2^{(1)} = \widehat{\mathbf{M}}_2^{(2)} = \tau\sqrt{\frac{\beta_2}{\nu_2}}\,\mathbf{M}_v$, is applied at each step of our iterative method.

4.2. GM2 model

In a completely analogous way we can derive saddle point preconditioners for the *GM2* model, for which

$$A = \begin{bmatrix} \mathbf{A}_{u,GM2} & -2\tau\gamma\mathbf{M}_{u(q-p)} & 0 & 0 \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix},$$

$$B = \begin{bmatrix} -\mathbf{L}_{u,GM2} & \tau\gamma\mathbf{M}_{u^2} & \tau\gamma\mathbf{M} & 0 \\ -2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} & 0 & \tau\gamma\mathbf{M} \end{bmatrix}, \qquad C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

in the notation of (4.1).

We may approximate the matrix A, in this case using the saddle point type structure of its upper sub-matrix, by

$$\widetilde{A} = \begin{bmatrix} \widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & \mathbf{A}_{v,GM2} & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} -\widetilde{\mathbf{A}}_1 & 0 & 0 & 0 \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} & 0 & 0 \\ 0 & 0 & \tau\nu_1\mathbf{M} & 0 \\ 0 & 0 & 0 & \tau\nu_2\mathbf{M} \end{bmatrix},$$

where

$$\widetilde{\mathbf{A}}_1 = \mathbf{A}_{u,GM2} - (2\tau\gamma)^2\mathbf{M}_{u(q-p)}\mathbf{A}_{v,GM2}^{-1}\mathbf{M}_{u(q-p)}.$$

We follow a similar strategy as before to approximate the Schur complement

$$S = \begin{bmatrix} -\mathbf{L}_{u,GM2} & \tau\gamma\mathbf{M}_{u^2} \\ -2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} \end{bmatrix}\widetilde{\mathbf{A}}_{(1,2)}^{-1}\begin{bmatrix} -\mathbf{L}_{u,GM2}^T & -2\tau\gamma\mathbf{M}_{uv} \\ \tau\gamma\mathbf{M}_{u^2} & -\mathbf{L}_{v,GM2}^T \end{bmatrix} + \tau\gamma^2\begin{bmatrix} \nu_1^{-1}\mathbf{M} & 0 \\ 0 & \nu_2^{-1}\mathbf{M} \end{bmatrix},$$

where for this problem

$$\widetilde{\mathbf{A}}_{(1,2)} := \begin{bmatrix} \mathbf{A}_{u,GM2} & -2\tau\gamma\mathbf{M}_{u(q-p)} \\ -2\tau\gamma\mathbf{M}_{u(q-p)} & \mathbf{A}_{v,GM2} \end{bmatrix}.$$

Again applying our matching strategy, we obtain the following approximation:

$$\widetilde{S} = \begin{bmatrix} -\mathbf{L}_{u,GM2} + \widehat{\mathbf{M}}_1^{(1)} & -\tau\gamma\mathbf{M}_{u^2} \\ 2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix}\widetilde{\mathbf{A}}_{(1,2)}^{-1}\begin{bmatrix} -\mathbf{L}_{u,GM2}^T + \widehat{\mathbf{M}}_1^{(2)} & 2\tau\gamma\mathbf{M}_{uv} \\ -\tau\gamma\mathbf{M}_{u^2} & -\mathbf{L}_{v,GM2}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix}.$$

Examining the diagonal blocks of $\widetilde{\mathbf{A}}_{(1,2)}^{-1}$ (as for the *GM1* model), and applying a matching strategy to approximate the second term of S, we motivate the following approximations:

$$\widehat{\mathbf{M}}_1^{(1)}\widetilde{\mathbf{A}}_1^{-1}\widehat{\mathbf{M}}_1^{(2)} \approx \frac{\tau\gamma^2}{\nu_1}\mathbf{M}, \qquad \widetilde{\mathbf{A}}_1 := \mathbf{A}_{u,GM2} - (2\tau\gamma)^2\mathbf{M}_{u(q-p)}\mathbf{A}_{v,GM2}^{-1}\mathbf{M}_{u(q-p)},$$

$$\widehat{\mathbf{M}}_2^{(1)}\widetilde{\mathbf{A}}_2^{-1}\widehat{\mathbf{M}}_2^{(2)} \approx \frac{\tau\gamma^2}{\nu_2}\mathbf{M}, \qquad \widetilde{\mathbf{A}}_2 := \mathbf{A}_{v,GM2} - (2\tau\gamma)^2\mathbf{M}_{u(q-p)}\mathbf{A}_{u,GM2}^{-1}\mathbf{M}_{u(q-p)}.$$

These approximations may be achieved by constructing diagonal matrices $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$, $\widehat{\mathbf{M}}_2^{(2)}$ with diagonal entries given by

$$\left[\widehat{\mathbf{M}}_1^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2}\cdot\left|\left[\widetilde{\mathbf{A}}_1\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_1^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_1}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2},$$

$$\left[\widehat{\mathbf{M}}_2^{(1)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2}\cdot\left|\left[\widetilde{\mathbf{A}}_2\right]_{jj}\right|, \qquad \left[\widehat{\mathbf{M}}_2^{(2)}\right]_{jj} = \sqrt{\frac{\tau}{\nu_2}}\,\gamma\,[\mathbf{M}]_{jj}^{1/2}.$$

We can again build these choices of $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$, $\widehat{\mathbf{M}}_2^{(2)}$ into the approximation $\widetilde{S}$ within our preconditioner.

For each of our suggested iterative methods, we insert our approximations of A and $S = BA^{-1}B^T$ into the general preconditioners for saddle point systems stated in (4.2).

4.3. Computational cost

When implementing our preconditioned iterative methods, the vast majority of the computational expense occurs when applying the inverse of our preconditioners. We therefore now wish to detail the solves that we are required to carry out when applying our preconditioner to each problem.

To enact our preconditioner for the *GM*1 model, we are required to perform the following:

- Solves for $\widetilde{\mathbf{A}}_1$ and $\widetilde{\mathbf{A}}_2$ ('mass-like' matrices),
- Chebyshev semi-iteration/diagonal solves for $\tau\nu_1\mathbf{M}$, $\tau\nu_2\mathbf{M}$ (mass matrices),
- 1 multigrid per Uzawa iteration for each of $-\mathbf{L}_{u,GM1}+\widehat{\mathbf{M}}_1^{(1)}$, $-\mathbf{L}_{v,GM1}+\widehat{\mathbf{M}}_2^{(1)}$, $-\mathbf{L}_{u,GM1}^T+\widehat{\mathbf{M}}_1^{(2)}$, $-\mathbf{L}_{v,GM1}^T+\widehat{\mathbf{M}}_2^{(2)}$ (to apply $\widetilde{S}^{-1}$).

From a computational point-of-view, the most straightforward operations involve inverting mass matrices, with the multigrid operations the most expensive.

For the Gauss–Newton approach for the same problem, the preconditioner is cheaper to apply, with the following operations dominating the computational cost:

- Chebyshev semi-iteration/diagonal solves for $\beta_1\tau\mathbf{M}$, $\beta_2\tau\mathbf{M}$, $\tau\nu_1\mathbf{M}$, $\tau\nu_2\mathbf{M}$ (mass matrices),
- 1 multigrid per Uzawa iteration for each of $-\mathbf{L}_{u,GM1}+\widehat{\mathbf{M}}_1^{(1)}$, $-\mathbf{L}_{v,GM1}+\widehat{\mathbf{M}}_2^{(1)}$, $-\mathbf{L}_{u,GM1}^T+\widehat{\mathbf{M}}_1^{(2)}$, $-\mathbf{L}_{v,GM1}^T+\widehat{\mathbf{M}}_2^{(2)}$ (for the new choices of $\widehat{\mathbf{M}}_1^{(1)}$, $\widehat{\mathbf{M}}_2^{(1)}$, $\widehat{\mathbf{M}}_1^{(2)}$, $\widehat{\mathbf{M}}_2^{(2)}$).

Further, the dominant computational operations for solving the *GM*2 model are:

- Solves for $\widetilde{\mathbf{A}}_1$, $\mathbf{A}_{u,GM2}$ ('mass-like' matrices),
- Chebyshev semi-iteration/diagonal solves for $\tau\nu_1\mathbf{M}$, $\tau\nu_2\mathbf{M}$ (mass matrices),
- 1 multigrid per Uzawa iteration for each of $-\mathbf{L}_{u,GM1}+\widehat{\mathbf{M}}_1^{(1)}$, $-\mathbf{L}_{v,GM1}+\widehat{\mathbf{M}}_2^{(1)}$, $-\mathbf{L}_{u,GM1}^T+\widehat{\mathbf{M}}_1^{(2)}$, $-\mathbf{L}_{v,GM1}^T+\widehat{\mathbf{M}}_2^{(2)}$ (to apply $\widetilde{S}^{-1}$).

We believe the amount of computational work required to apply our preconditioner is satisfactory when taking into account the complex structure and large dimension of the matrix systems.

4.4. Alternative methods

Before presenting numerical results we wish to briefly discuss alternative approaches for the solution of the optimization problem, or the linear systems at the heart of the nonlinear solver. Due to the highly complex structure of the problem statements and associated matrix systems, we are not aware of any robust methods that have previously been developed for solving these systems. We believe that this underlines the value of investigating preconditioned iterative methods for this important class of problems. However we wish to outline other classes of methods which could potentially be applied to these problems, in many cases building on the work described in this article.

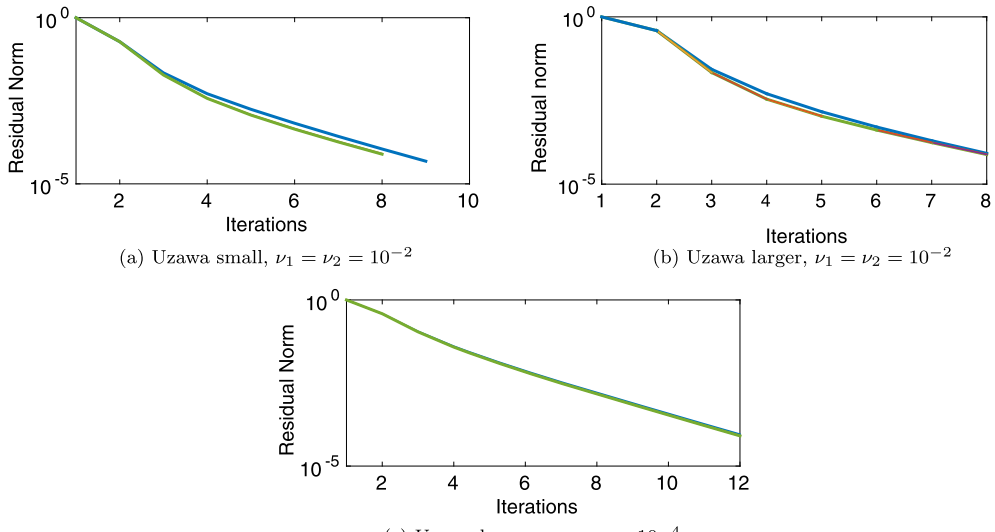
The method we have presented is applied when using either a Newton or a Gauss–Newton approach for the nonlinear program. Alternatively, we could apply a simple gradient descent coupled with a line search procedure [29], which typically converges very slowly. Another alternative would be to employ an interior point scheme [44], which also requires the solution of saddle point problems, and we believe that many of our proposed techniques could be carried over to this case. In [21] the authors follow a so-called one-shot method that can be viewed as a stationary iteration of the form $\mathbf{x}_{k+1}=\mathbf{x}_k+\mathcal{P}^{-1}\mathbf{r}_k$. The preconditioner in this case is given by a block matrix that requires the (approximate) inversion of the adjoint and forward PDE operator, as well as the solution of a complicated Schur complement system (here written for the Gauss–Newton system (3.2) with the block containing second derivatives with respect to Lagrange multipliers equal to zero):

$$S_A \approx \begin{bmatrix} \widehat{\mathcal{L}}_{aa} & \\ & \widehat{\mathcal{L}}_{bb} \end{bmatrix} + \begin{bmatrix} \mathcal{L}_{ap} & \mathcal{L}_{aq} \\ \mathcal{L}_{bp} & \mathcal{L}_{bq} \end{bmatrix} \begin{bmatrix} \mathcal{L}_{up} & \mathcal{L}_{uq} \\ \mathcal{L}_{vp} & \mathcal{L}_{vq} \end{bmatrix}^{-1} \begin{bmatrix} \widehat{\mathcal{L}}_{uu} & \\ & \widehat{\mathcal{L}}_{vv} \end{bmatrix} \begin{bmatrix} \mathcal{L}_{pu} & \mathcal{L}_{pv} \\ \mathcal{L}_{qu} & \mathcal{L}_{qv} \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{L}_{pa} & \mathcal{L}_{pb} \\ \mathcal{L}_{qa} & \mathcal{L}_{qb} \end{bmatrix}.$$

Note that this is in general harder to approximate than the Schur complements obtained using our approach, as we here have the sum of mass matrices and inverse PDE operators within the Schur complement approximation. As it is the action of S_A^{-1} that is important for preconditioning purposes, this approximation is extremely difficult to apply in practice.

Recently, operator preconditioning approaches have proven successful for many PDE preconditioning problems (see [25, 45]). Their use for nonlinear problems has recently attracted attention within the field [1,38], and we are currently investigating how to extend these approaches to the reaction–diffusion type setting.

Of the approaches currently within reach, the stationary iteration approaches were found not to converge, and the above approximation S_A is of a very complex nature and is infeasible to apply. The only alternative approach which we found



(a) Uzawa small, $\nu_1 = \nu_2 = 10^{-2}$

(b) Uzawa larger, $\nu_1 = \nu_2 = 10^{-2}$

(c) Uzawa larger, $\nu_1 = \nu_2 = 10^{-4}$

Fig. 4.1. Iteration numbers for Uzawa scheme with a block triangular version of our preconditioner. Plots are shown for every SQP step, however they can often not be distinguished due to very similar convergence behavior. The smaller example uses 729 degrees of freedom in space, and the larger example 4096.

to generate sensible results for a range of parameter regimes is that of an Uzawa method, using a preconditioner of the form derived in this paper. In Fig. 4.1 we show the iteration numbers that are required using such an approach for two different meshes and two regularization parameters. We note that, while the results show that this method performs well, we are required to move from the block diagonal preconditioner derived in this section to a more expensive block triangular preconditioner, as the Uzawa method diverges when a block diagonal approach is taken. Apart from this change, the major computational operations required per Uzawa iteration are largely similar to those required to apply our preconditioner with a single Uzawa step. We also highlight that this method is itself only feasible due to the preconditioners derived in this paper.

5. Numerical results

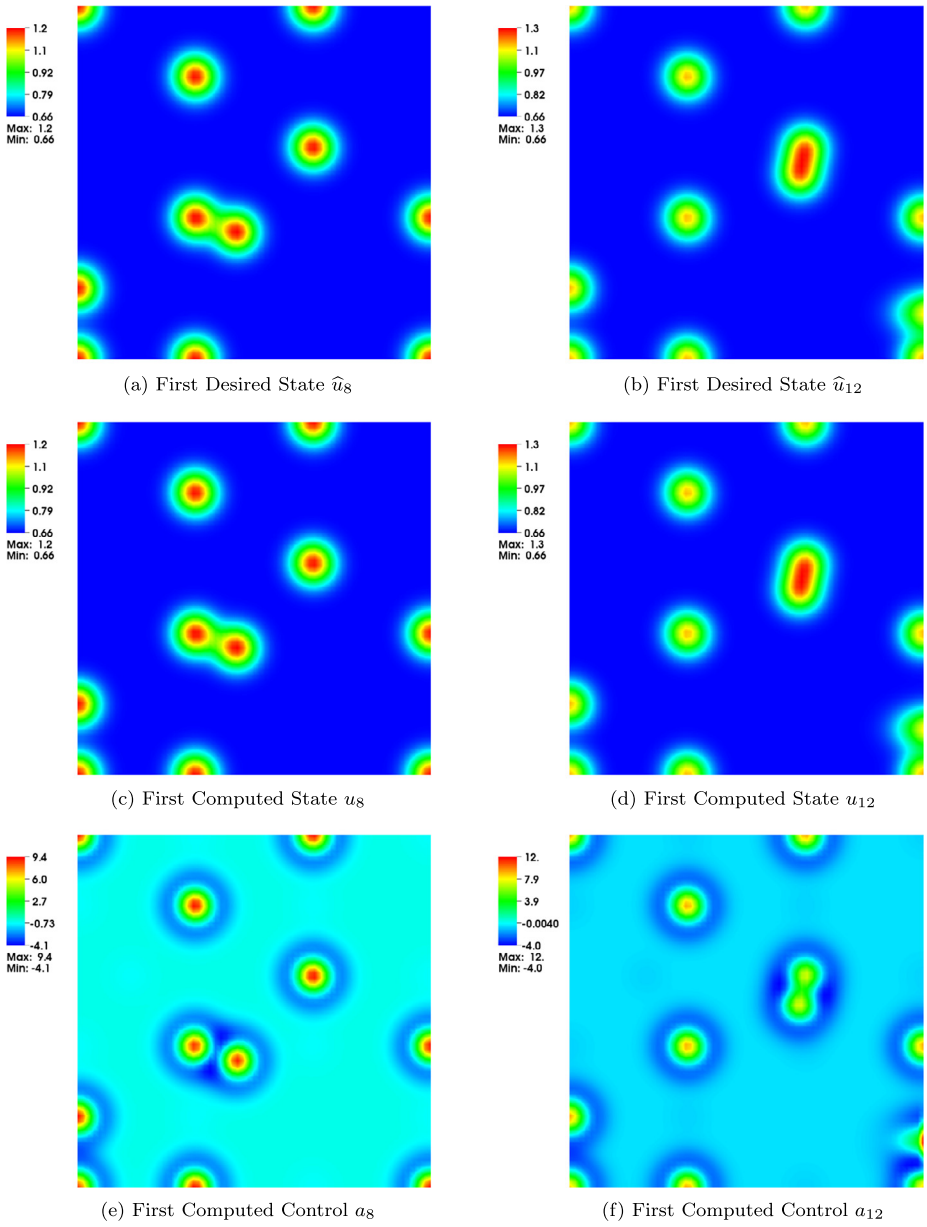
We now wish to apply our methodology to a number of test problems. All results presented in this section are based on an implementation of the presented algorithms and (block diagonal) preconditioners within the deal.II [3] framework using $Q1$ finite elements. The AMG preconditioner we use is part of the Trilinos ML package [13] that implements a smoothed aggregation AMG. Within the algebraic multigrid routine we typically apply 10 steps of a Chebyshev smoother in combination with the application of two V-cycles. Typically we apply 4 iterations of the Uzawa scheme within our Schur complement approximation, to guarantee high accuracy. For our implementation of BiCG we use a stopping tolerance of 10^{-4}. Our experiments are performed for $T = 1$ and $\tau = 0.05$, i.e. 20 time-steps. Typically, the spatial domain Ω is considered to be the unit square or cube. All results are performed on a Centos Linux machine with Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz CPUs and 48 GB of RAM.

5.1. *GM*2 model

For both *GM*2 and *GM*1 models we start creating desired states using Gaussians placed at different positions in the unit square/cube that might depend on the time t. In Fig. 5.1 we illustrate two instances of the desired state and computed results for the *GM*2 formulation, with the parameters set to $D_u = 1$, $D_v = 10$, $\beta_1 = \beta_2 = 1$, $\gamma = 50$, and $\nu_1 = \nu_2 = 10^{-6}$. As the regularization parameters become smaller we see that the desired and computed states are very close. This is reflected in the third set of images within Fig. 5.1, where the control is shown with sometimes rather high values. In Table 5.1 we present BiCG iteration numbers for solving this test problem for a range of degrees of freedom and regularization parameters – the results indicate that our solver is robust in a large number of problem setups.

5.2. *GM*1 model with Newton and Gauss–Newton methods

For the next problem we examine, the desired state for the *GM*1 model is created using Gaussian functions placed in the unit cube. This is illustrated in Fig. 5.2, where we present the desired state for the first component, the computed first state variable, and the corresponding control variable. The parameters for this case are chosen to be $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-2}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. For many interesting parameter setups (including for a range of values



(a) First Desired State $\widehat{u}_8$ (b) First Desired State $\widehat{u}_{12}$

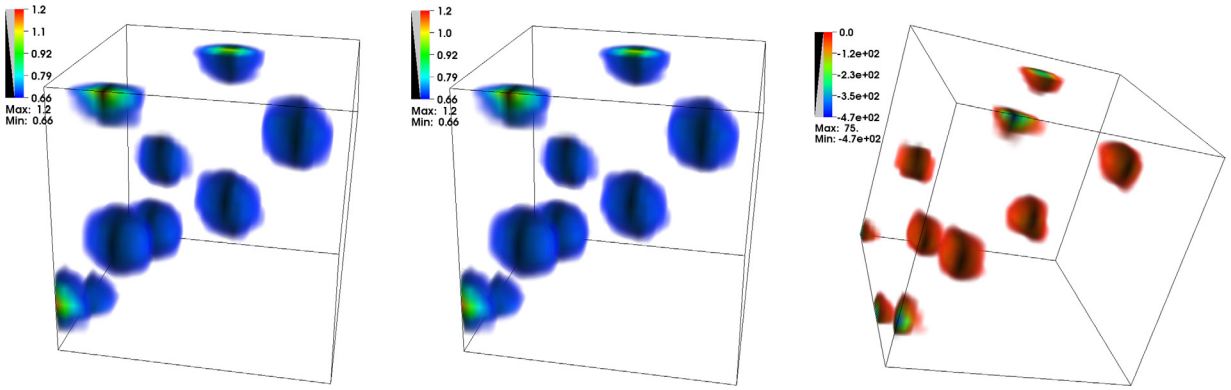
(c) First Computed State u_8 (d) First Computed State u_{12}

(e) First Computed Control a_8 (f) First Computed Control a_{12}

Fig. 5.1. Desired state for 8th and 12th grid points in time (upper two), computed state using the *GM2* model (middle two), and the computed control (lower two) for two reactants using the *GM2* model. The parameters are set to be $D_u = 1$, $D_v = 10$, $\beta_1 = \beta_2 = 1$, $\gamma = 50$, and $\nu_1 = \nu_2 = 10^{-6}$.

Table 5.1
Results on unit square with $D_u = 1$, $D_v = 10$, $\beta_1 = \beta_2 = 1$, and $\gamma = 50$. Stated are BiCG iteration numbers for each Newton step.

DoF	$\nu_1 = \nu_2 = 10^{-2}$		$\nu_1 = \nu_2 = 10^{-4}$		$\nu_1 = \nu_2 = 10^{-6}$	
	Newton	BiCG	Newton	BiCG	Newton	BiCG
507,000	step 1	18	step 1	16	step 1	16
	step 2	20	step 2	15	step 2	15
	step 3	20	step 3	15	step 3	15
	step 4	20	step 4	15	step 4	15
	step 5	20	step 5	15		
1,996,920	step 1	23	step 1	17	step 1	17
	step 2	23	step 2	18	step 2	16
	step 3	24	step 3	18	step 3	16
	step 4	23	step 4	18	step 4	16
	step 5	23	step 5	18		



(a) First Desired State (b) Computed First State (c) Computed Control

Fig. 5.2. Desired state, computed state and computed control for the first reactant in the *GM*1 model with parameters at $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-2}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$.

Table 5.2
Results on unit cube with $\beta_1 = \beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. Stated are BiCG iteration numbers for each Gauss–Newton step.

DoF	$\nu_1 = \nu_2 = 10^{-2}$		$\nu_1 = \nu_2 = 10^{-4}$		$\nu_1 = \nu_2 = 10^{-6}$	
	GN	BiCG	GN	BiCG	GN	BiCG
87,480	step 1	11	step 1	11	step 1	9
	step 2	11	step 2	11	step 2	9
	step 3	11	step 3	11	step 3	9
	step 4	11	step 4	11		
	step 5	11	step 5	11		
589,560	step 1	11	step 1	13	step 1	11
	step 2	11	step 2	12	step 2	11
	step 3	11	step 3	12	step 3	11
	step 4	11	step 4	12	step 4	11
	step 5	11	step 5	12		
4,312,440	step 1	11	step 1	13	step 1	11
	step 2	11	step 2	12	step 2	11
	step 3	11	step 3	12	step 3	11
	step 4	11	step 4	12	step 4	11
	step 5	11	step 5	12		

of r) it is not trivial to find a configuration of the Newton scheme that demonstrates satisfying convergence properties. We instead focus on the Gauss–Newton method here, and we illustrate the BiCG iteration numbers achieved for a range of problems in Table 5.2 – these results demonstrate much greater robustness, with rapid convergence of the inner solver.

As we have already highlighted, the complex structure of the linear systems makes the design of efficient preconditioners harder when the Newton scheme is applied compared to the Gauss–Newton scheme. We use the results presented in Table 5.3 to illustrate the performance of both Newton and Gauss–Newton schemes. We observe that the Newton method and our associated preconditioner perform well when the regularization parameters are chosen to be rather large. In this case, whereas the Gauss–Newton scheme generates low iteration numbers, it seems to generate less meaningful numerical results. Whereas, in the case of smaller regularization parameters, the Gauss–Newton scheme requires a greater number of outer iterations, the number of inner BiCG iterations remains low. For this setup the preconditioner for the Newton scheme does not allow the method to converge, and therefore the scheme failed. This makes clear that careful choices concerning the outer iteration and preconditioner need to be made, in order to achieve good performance of the method for a particular parameter case.

Additionally, we illustrate in Table 5.4 the performance of the Gauss–Newton scheme when the tolerance of the inner solver is relaxed. We can see that for this mesh the choice of a tolerance decrease from 10^{-4} to 10^{-2} does not have a significant influence on the output of the optimization routine and the number of outer iterations.

We also wish to highlight that it is possible to include additional control constraints $\underline{a} \leq a \leq \bar{a}$ and $\underline{b} \leq b \leq \bar{b}$, to be enforced along with the systems of PDEs (2.2) or (2.3). Our approach to deal with these additional bounds is to include a Moreau–Yosida penalization [18] that can be used with a non-smooth Newton scheme. The structure of the Newton system is very similar to the one without control constraints, and we refer to [32] for more details on the derivation of the non-smooth Newton system and the choice of preconditioner. In Table 5.5 we present results for the setup $0 \leq a$ and $0 \leq b$, where the Gauss–Newton scheme is used in conjunction with BiCG.

Table 5.3
Results on unit cube with $\beta_1 = \beta_2 = 1$, $D_u = 1$, $D_v = 10$, and $r = 10^{-4}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. We show the iteration numbers for BICG, the value of the data misfit term $\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$ in the objective function, and the relative change in the optimization variable between two consecutive Gauss–Newton iterations (GN_Δ). The outer iteration is stopped if GN_Δ is smaller than 10^{-4}.

DoF		Newton	GN	GN
87,480		$\nu_1 = \nu_2 = 10^2$ BICG/$\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$/$GN_\Delta$	$\nu_1 = \nu_2 = 10^2$ BICG/$\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$/$GN_\Delta$	$\nu_1 = \nu_2 = 10^{-3}$ BICG/$\|\mathbf{y} - \widehat{\mathbf{y}}\|^2$/$GN_\Delta$
	step 1	2/4.67/–	2/38.53/–	2/38.53/–
	step 2	5/4.27/7.4×10^{-2}	7/38.45/3.6×10^{-3}	16/5.1×10^{-1}/9.8×10^{-1}
	step 3	5/4.27/1.2×10^{-2}	7/38.45/9.7×10^{-4}	19/3.8×10^{-2}/2.1×10^{0}
	step 4	5/4.27/1.9×10^{-3}	7/38.45/3.3×10^{-6}	16/3.3×10^{-2}/4.3×10^{-1}
	step 5	5/4.27/2.2×10^{-4}		15/3.3×10^{-2}/2.7×10^{-3}
	step 6	5/4.27/1.8×10^{-5}		15/3.2×10^{-2}/7.9×10^{-5}

Table 5.4
Results on unit cube with $\beta_1 = \beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. We show the iteration numbers for BICG and the value of the objective function $\mathcal{J}(\cdot)$. The outer iteration was stopped if the relative difference between two consecutive iterates was smaller than 10^{-4}.

DoF		Newton	GN	GN
87,480		$\text{tol} = 10^{-4}$ BICG/$\mathcal{J}(\cdot)$	$\text{tol} = 10^{-2}$ BICG/$\mathcal{J}(\cdot)$	$\text{tol} = 10^{-1}$ BICG/$\mathcal{J}(\cdot)$
	step 1	2/1926.98	2/1926.98	2/1926.98
	step 2	11/13.57	7/13.58	3/12.63
	step 3	11/1.78	7/1.79	3/1.19
	step 4	11/2.20	5/2.20	3/2.30
	step 5	11/2.20	5/2.22	3/2.17
	step 6	11/2.20	5/2.22	3/2.18
	step 7		5/2.22	3/2.18
	step 8			3/2.18
	step 9			3/2.18

Table 5.5
Results on unit cube with $\beta_1 = \beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2. Stated are BICG iteration numbers for each Gauss–Newton step. The tolerance for the Gauss–Newton method is 10^{-2}.

DoF	$\nu_1 = \nu_2 = 10^{-2}$		$\nu_1 = \nu_2 = 10^{-4}$	
	GN	BICG	GN	BICG
130,680	step 1	2	step 1	2
	step 2	9	step 2	13
	step 3	13	step 3	14
507,000	step 1	4	step 1	4
	step 2	9	step 2	15
	step 3	10	step 3	15
1,996,920	step 1	10	step 1	10
	step 2	14	step 2	17
	step 3	14	step 3	17

We now wish to compare the performance of our preconditioned BICG approach with an unpreconditioned version of the same solver. We do so in order to demonstrate the clearly superior convergence properties of the preconditioned method, and show the very high accuracy to which our solver is able to solve the matrix systems. Fig. 5.3 illustrates the residual error of the unpreconditioned and preconditioned approaches for two test problems: it is clear that running the preconditioned method for only a few iterations easily outperforms the unpreconditioned version. Indeed the unpreconditioned method appears to diverge in many cases, demonstrating the need to construct effective preconditioners for the complex matrix systems involved.

5.3. Image-driven desired state and GM1 model

An attractive feature of this methodology is that it is also possible to obtain desired states by reading in pattern information from an image. This may be done for the *GM*1 and *GM*2 models, whether or not control constraints are included. Image-driven parameter estimation techniques can also be found in [19]. For this problem, we choose to take an image similar to those used in [24] – this involves reading in a pattern found on a mature jaguar. As this problem is not necessarily time-dependent we wish to illustrate the performance of our method by scaling the desired pattern by τ_i, where i denotes

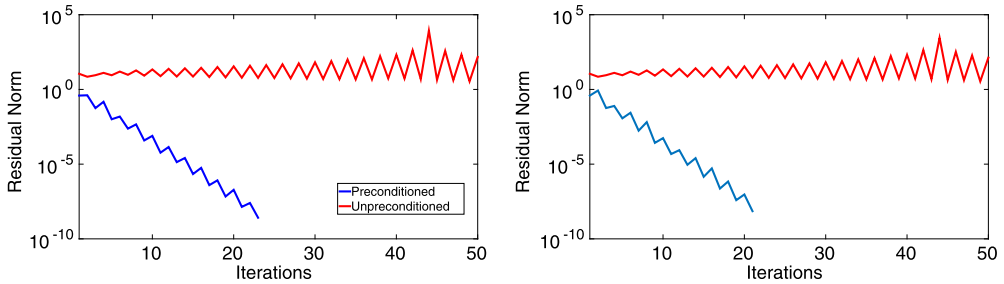


Fig. 5.3. Residual errors of unpreconditioned (red) and preconditioned (blue) BiCG method for two test problems (DoFs 4096). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

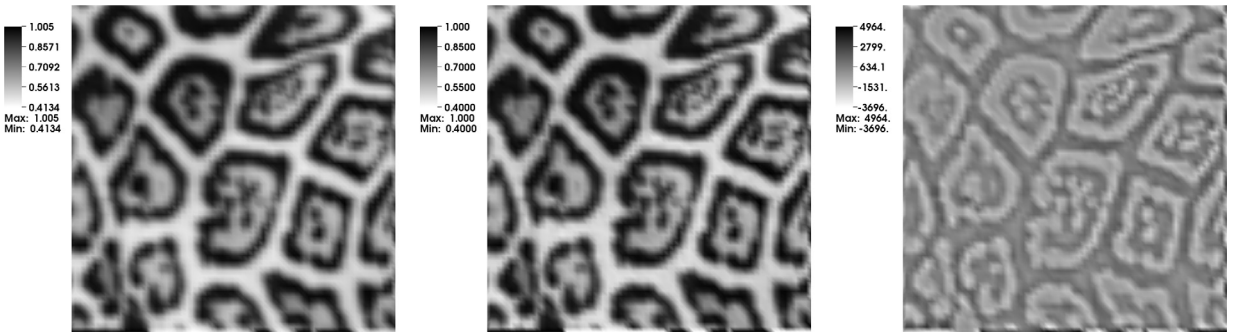


Fig. 5.4. Results for image-driven model: Shown are computed state, desired state, and computed control for the parameter setups using $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-7}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-5}$.

the relevant index in time. The results for applying the Gauss–Newton scheme to this image-driven problem are shown in Fig. 5.4.

In Fig. 5.4(b) we show the desired state used, with the computed state presented in Fig. 5.4(a) and the associated control in Fig. 5.4(c).

The parameters for this setup are $\beta_1 = \beta_2 = 10^2$, $\nu_1 = \nu_2 = 10^{-7}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-5}$. For the computations from which Fig. 5.4 is generated, a tolerance of 10^{-2} is taken for the Gauss–Newton scheme. Within these computations 8 steps of the Gauss–Newton iteration are required, with an average of 20.5 BiCG iterations per Gauss–Newton step.

Overall the numerical results presented for the above experiments indicate that we are able to solve a wide range of parameter identification problems from pattern formation, with our observed BiCG iteration numbers (as well as computation times) being low for a large number of parameter regimes. Furthermore, the iteration numbers behave in a fairly robust way as the parameters involved in the problem are varied.

6. Concluding remarks and future work

In this article, we have considered the development of preconditioned iterative methods for the numerical solution of parameter identification problems arising from pattern formation. We have constructed our methods using effective strategies for approximating the $(1,1)$-block and Schur complement of the saddle point systems that result from these problems.

The numerical results we have obtained when applying our techniques to a number of test examples (using both *GM*1 and *GM*2 models) indicate that our proposed solvers are effective ones for a wide range of parameter setups. Another key aspect of our methodology is that we are able to feed desired states (or "target patterns") into our implementation using experimental or computational data, and use this to obtain appropriate solutions to the Turing model in question. Furthermore, our solvers are found to be effective at handling additional inequality constraints on the control variables.

There are a number of related areas of research which we hope to consider, including the incorporation of additional constraints on the state or control variables (for instance integral constraints, or bounds on the state variables), different time-stepping schemes, and possibly different techniques for the outer iteration. We also wish to investigate a version of the problem where the L_2-distance between the states and desired states is only measured at the final time $t = T$ (i.e. where $\beta_1 = \beta_2 = 0$), as we find that such problems have considerable physical applicability. Furthermore, we now hope to tackle other problems of significant interest to the mathematical biology community using the methodology presented in this paper.

Acknowledgements

We would like to thank two anonymous referees for their helpful and insightful comments. The second author was supported for this work in part by the Engineering and Physical Sciences Research Council (EPSRC) Grant EP/P505216/1, and by an EPSRC Doctoral Prize (EP/K503113/1). The authors are grateful for the award of a European Science Foundation (ESF) Exchange Grant OPTPDE-3997 under the OPTPDE programme, and express their thanks to the Max Planck Institute in Magdeburg.

Appendix A. Derivation of the Newton systems

For the Gierer–Meinhardt ($GM1$) formulation, we examine the forward equations

$$\begin{aligned} u_t - D_u\Delta u - \frac{ru^2}{v} + au &= r, \quad \text{on } \Omega\times[0,T],\\ v_t - D_v\Delta v - ru^2 + bv &= 0, \quad \text{on } \Omega\times[0,T],\\ u(\mathbf{x},0) = u_0(\mathbf{x}), \quad v(\mathbf{x},0) &= v_0(\mathbf{x}), \quad \text{on } \Omega,\\ \frac{\partial u}{\partial \nu} = \frac{\partial v}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned}$$

and the adjoint equations (see [11])

$$\begin{aligned} -p_t - D_u\Delta p - 2r\frac{u}{v}p + ap - 2ruq &= \beta_1(u-\widehat{u}), \quad \text{on } \Omega\times[0,T],\\ -q_t - D_v\Delta q + r\frac{u^2}{v^2}p + bq &= \beta_2(v-\widehat{v}), \quad \text{on } \Omega\times[0,T],\\ p(\mathbf{x},T) = \beta_{T,1}(u(\mathbf{x},T)-\widehat{u}(\mathbf{x},T)), \quad q(\mathbf{x},T) &= \beta_{T,2}(v(\mathbf{x},T)-\widehat{v}(\mathbf{x},T)), \quad \text{on } \Omega,\\ \frac{\partial p}{\partial \nu} = \frac{\partial q}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned}$$

where p and q denote the *adjoint variables*.

We now employ a Newton iteration, by writing at each Newton step

$$u = \bar{u}+\delta u, \quad v = \bar{v}+\delta v, \quad a = \bar{a}+\delta a, \quad b = \bar{b}+\delta b, \quad p = \bar{p}+\delta p, \quad q = \bar{q}+\delta q,$$

where $\bar{u}$, $\bar{v}$, $\bar{a}$, $\bar{b}$, $\bar{p}$, $\bar{q}$ denote the most recent iterates of u, v, a, b, p, q, with δu, δv, δa, δb, δp, δq denoting the changes in the solutions at each Newton step.

Applying this to the forward equations yields

$$\begin{aligned} (\bar{u}+\delta u)_t - D_u\Delta(\bar{u}+\delta u) - \frac{r(\bar{u}+\delta u)^2}{\bar{v}+\delta v} + (\bar{a}+\delta a)(\bar{u}+\delta u) &= r, \quad \text{on } \Omega\times[0,T],\\ (\bar{v}+\delta v)_t - D_v\Delta(\bar{v}+\delta v) - r(\bar{u}+\delta u)^2 + (\bar{b}+\delta b)(\bar{v}+\delta v) &= 0, \quad \text{on } \Omega\times[0,T],\\ (\bar{u}+\delta u)(\mathbf{x},0) = u_0(\mathbf{x}), \quad (\bar{v}+\delta v)(\mathbf{x},0) &= v_0(\mathbf{x}), \quad \text{on } \Omega,\\ \frac{\partial(\bar{u}+\delta u)}{\partial \nu} = \frac{\partial(\bar{v}+\delta v)}{\partial \nu} &= 0, \quad \text{on } \partial\Omega\times[0,T], \end{aligned}$$

whereupon we can use the assumption $(\bar{u}+\delta u)^2 \approx \bar{u}^2 + 2\bar{u}\cdot\delta u$ and the resulting derivation

$$\frac{(\bar{u}+\delta u)^2}{\bar{v}+\delta v} \approx \frac{\bar{v}-\delta v}{\bar{v}^2}(\bar{u}^2+2\bar{u}\cdot\delta u) \approx \frac{\bar{u}^2\bar{v}-\bar{u}^2\cdot\delta v+2\bar{u}\bar{v}\cdot\delta u}{\bar{v}^2}$$

to write

$$(\delta u)_t - D_u\Delta(\delta u) + r\frac{\bar{u}^2\cdot\delta v - 2\bar{u}\bar{v}\cdot\delta u}{\bar{v}^2} + \bar{u}\cdot\delta a + \bar{a}\cdot\delta u = r - \left(\bar{u}_t - D_u\Delta\bar{u} - \frac{r\bar{u}^2}{\bar{v}} + \bar{a}\bar{u}\right), \quad \text{on } \Omega\times[0,T], \tag{A.1}$$

$$(\delta v)_t - D_v\Delta(\delta v) - 2r\bar{u}\cdot\delta u + \bar{v}\cdot\delta b + \bar{b}\cdot\delta v = -(\bar{v}_t - D_v\Delta\bar{v} - r\bar{u}^2 + \bar{b}\bar{v}), \quad \text{on } \Omega\times[0,T], \tag{A.2}$$

$$(\delta u)(\mathbf{x},0) = (\delta v)(\mathbf{x},0) = 0, \quad \text{on } \Omega, \tag{A.3}$$

$$\frac{\partial(\delta u)}{\partial \nu} = \frac{\partial(\delta v)}{\partial \nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \tag{A.4}$$

Considering now a Newton iteration applied to the adjoint equations, we have

$$
\begin{aligned}
&-(\bar{p}+\delta p)_t - D_u\Delta(\bar{p}+\delta p) - 2r\frac{\bar{u}+\delta u}{\bar{v}+\delta v}(\bar{p}+\delta p) + (\bar{a}+\delta a)(\bar{p}+\delta p) - 2r(\bar{u}+\delta u)(\bar{q}+\delta q)\\
&\quad = \beta_1((\bar{u}+\delta u)-\widehat{u}), \quad \text{on } \Omega\times[0,T],\\
&-(\bar{q}+\delta q)_t - D_v\Delta(\bar{q}+\delta q) + r\frac{(\bar{u}+\delta u)^2}{(\bar{v}+\delta v)^2}(\bar{p}+\delta p) + (\bar{b}+\delta b)(\bar{q}+\delta q)\\
&\quad = \beta_2((\bar{v}+\delta v)-\widehat{v}), \quad \text{on } \Omega\times[0,T],\\
&(\bar{p}+\delta p)(\mathbf{x},T) = \beta_{T,1}((\bar{u}+\delta u)(\mathbf{x},T)-\widehat{u}(\mathbf{x},T)), \quad \text{on } \Omega,\\
&(\bar{q}+\delta q)(\mathbf{x},T) = \beta_{T,2}((\bar{v}+\delta v)(\mathbf{x},T)-\widehat{v}(\mathbf{x},T)), \quad \text{on } \Omega,\\
&\quad \frac{\partial(\bar{p}+\delta p)}{\partial\nu} = \frac{\partial(\bar{q}+\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T].
\end{aligned}
$$

Now, using the approximations

$$
\begin{aligned}
\frac{\bar{u}+\delta u}{\bar{v}+\delta v}(\bar{p}+\delta p) &\approx \frac{(\bar{u}+\delta u)(\bar{v}-\delta v)(\bar{p}+\delta p)}{\bar{v}^2}\\
&\approx \frac{\bar{u}\bar{v}\bar{p}+\bar{v}\bar{p}\cdot\delta u-\bar{u}\bar{p}\cdot\delta v+\bar{u}\bar{v}\cdot\delta p}{\bar{v}^2},\\
\frac{(\bar{u}+\delta u)^2}{(\bar{v}+\delta v)^2}(\bar{p}+\delta p) &\approx \frac{(\bar{u}+2\bar{u}\cdot\delta u)(\bar{v}^2-2\bar{v}\cdot\delta v)(\bar{p}+\delta p)}{\bar{v}^4}\\
&\approx \frac{\bar{u}}{\bar{v}^3}(\bar{u}\bar{v}\bar{p}+2\bar{v}\bar{p}\cdot\delta u-2\bar{u}\bar{p}\cdot\delta v+\bar{u}\bar{v}\cdot\delta p),
\end{aligned}
$$

we may write

$$
\begin{aligned}
&-(\delta p)_t - D_u\Delta(\delta p) - 2r\frac{\bar{u}\bar{p}\cdot\delta v-\bar{v}\bar{p}\cdot\delta u-\bar{u}\bar{v}\cdot\delta p}{\bar{v}^2} + \bar{p}\cdot\delta a+\bar{a}\cdot\delta p-2r(\bar{u}\cdot\delta q+\bar{q}\cdot\delta u)-\beta_1\delta u\\
&\quad = \beta_1(\bar{u}-\widehat{u}) - \left(-\bar{p}_t - D_u\Delta\bar{p} - 2r\frac{\bar{u}}{\bar{v}}\bar{p}+\bar{a}\bar{p}-2r\bar{u}\bar{q}\right), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.5)}\\
&-(\delta q)_t - D_v\Delta(\delta q) + r\bar{u}\frac{2\bar{v}\bar{p}\cdot\delta u+\bar{u}\bar{v}\cdot\delta p-2\bar{u}\bar{p}\cdot\delta v}{\bar{v}^2} + \bar{q}\cdot\delta b+\bar{b}\cdot\delta q-\beta_2\delta v\\
&\quad = \beta_2(\bar{v}-\widehat{v}) - \left(-\bar{q}_t - D_v\Delta\bar{q} + r\frac{\bar{u}^2}{\bar{v}^2}\bar{p}+\bar{b}\bar{q}\right), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.6)}\\
&(\delta p)(\mathbf{x},T) = \beta_{T,1}(\delta u)(\mathbf{x},T), \quad (\delta q)(\mathbf{x},T) = \beta_{T,2}(\delta v)(\mathbf{x},T), \quad \text{on } \Omega, \qquad \text{(A.7)}\\
&\quad \frac{\partial(\delta p)}{\partial\nu} = \frac{\partial(\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \qquad \text{(A.8)}
\end{aligned}
$$

Now, the forward and adjoint equations can clearly be derived by differentiating the Lagrangian

$$
\begin{aligned}
\mathcal{J}_{GM1}(u,v,a,b,p,q) = &\frac{\beta_1}{2}\|u-\widehat{u}\|^2_{L_2(\Omega\times[0,T])} + \frac{\beta_2}{2}\|v-\widehat{v}\|^2_{L_2(\Omega\times[0,T])}\\
&+\frac{\beta_{T,1}}{2}\|u-\widehat{u}_T\|^2_{L_2(\Omega)} + \frac{\beta_{T,2}}{2}\|v-\widehat{v}_T\|^2_{L_2(\Omega)}\\
&+\frac{\nu_1}{2}\|a\|^2_{L_2(\Omega\times[0,T])} + \frac{\nu_2}{2}\|b\|^2_{L_2(\Omega\times[0,T])}\\
&-\int_{\Omega\times[0,T]} p\left(u_t - D_u\Delta u - \frac{ru^2}{v}+au-r\right)\\
&-\int_{\Omega\times[0,T]} q\left(v_t - D_v\Delta v - ru^2+bv\right),
\end{aligned}
$$

with respect to the adjoint variables p, q and the state variables u, v, respectively. Within this cost functional, we have excluded the constraints on the boundary conditions for readability reasons. To obtain the gradient equations we require for a closed system of equations, we also need to differentiate the above cost functional with respect to the control variables a and b. Differentiating with respect to a gives the requirement

$$\int_{\Omega\times[0,T]} (up - \nu_1 a) = 0,$$

and differentiating with respect to b yields similarly that

$$\int_{\Omega\times[0,T]} (vq - \nu_2 b) = 0.$$

Applying a Newton iteration to these equations gives constraints of the form

$$\int_{\Omega\times[0,T]} (\bar{p}\cdot\delta u + \bar{u}\cdot\delta p - \nu_1\delta a) = -\int_{\Omega\times[0,T]} (\bar{u}\bar{p} - \nu_1\bar{a}), \tag{A.9}$$

$$\int_{\Omega\times[0,T]} (\bar{q}\cdot\delta v + \bar{v}\cdot\delta q - \nu_2\delta b) = -\int_{\Omega\times[0,T]} (\bar{v}\bar{q} - \nu_2\bar{b}), \tag{A.10}$$

at each Newton step.

Therefore the complete system which we need to solve at each Newton step corresponds to the adjoint equations (A.5)–(A.8), the gradient equations (A.9) and (A.10), and the forward equations (A.1)–(A.4).

We now turn our attention to the Schnakenberg ($GM2$) model, where we wish to deal with the forward equations

$$\begin{aligned}
u_t - D_u\Delta u + \gamma(u - u^2 v) - \gamma a = 0, \quad & \text{on } \Omega\times[0,T],\\
v_t - D_v\Delta v + \gamma u^2 v - \gamma b = 0, \quad & \text{on } \Omega\times[0,T],\\
u(\mathbf{x},0) = u_0(\mathbf{x}), \quad v(\mathbf{x},0) = v_0(\mathbf{x}), \quad & \text{on } \Omega,\\
\frac{\partial u}{\partial \nu} = \frac{\partial v}{\partial \nu} = 0, \quad & \text{on } \partial\Omega\times[0,T],
\end{aligned}$$

and the adjoint equations (see [11])

$$\begin{aligned}
-p_t - D_u\Delta p + 2\gamma u v(q-p) + \gamma p = \beta_1(u - \widehat{u}), \quad & \text{on } \Omega\times[0,T],\\
-q_t - D_v\Delta q + \gamma u^2(q-p) = \beta_2(v - \widehat{v}), \quad & \text{on } \Omega\times[0,T],\\
p(\mathbf{x},T) = \beta_{T,1}(u(\mathbf{x},T) - \widehat{u}(\mathbf{x},T)), \quad q(\mathbf{x},T) = \beta_{T,2}(v(\mathbf{x},T) - \widehat{v}(\mathbf{x},T)), \quad & \text{on } \Omega,\\
\frac{\partial p}{\partial \nu} = \frac{\partial q}{\partial \nu} = 0, \quad & \text{on } \partial\Omega\times[0,T].
\end{aligned}$$

Now, substituting

$$u = \bar{u} + \delta u, \quad v = \bar{v} + \delta v, \quad a = \bar{a} + \delta a, \quad b = \bar{b} + \delta b, \quad p = \bar{p} + \delta p, \quad q = \bar{q} + \delta q,$$

into the forward equations at each Newton step gives

$$\begin{aligned}
(\bar{u}+\delta u)_t - D_u\Delta(\bar{u}+\delta u) + \gamma((\bar{u}+\delta u) - (\bar{u}+\delta u)^2(\bar{v}+\delta v)) - \gamma(\bar{a}+\delta a) = 0, \quad & \text{on } \Omega\times[0,T],\\
(\bar{v}+\delta v)_t - D_v\Delta(\bar{v}+\delta v) + \gamma(\bar{u}+\delta u)^2(\bar{v}+\delta v) - \gamma(\bar{b}+\delta b) = 0, \quad & \text{on } \Omega\times[0,T],\\
(\bar{u}+\delta u)(\mathbf{x},0) = u_0(\mathbf{x}), \quad (\bar{v}+\delta v)(\mathbf{x},0) = v_0(\mathbf{x}), \quad & \text{on } \Omega,\\
\frac{\partial(\bar{u}+\delta u)}{\partial \nu} = \frac{\partial(\bar{v}+\delta v)}{\partial \nu} = 0, \quad & \text{on } \partial\Omega\times[0,T],
\end{aligned}$$

which we may expand and simplify to give

$$\begin{aligned}
&(\delta u)_t - D_u\Delta(\delta u) + \gamma(\delta u - \bar{u}^2\cdot\delta v - 2\bar{u}\bar{v}\cdot\delta u) - \gamma\delta a\\
&\quad = -(\bar{u}_t - D_u\Delta\bar{u} + \gamma(\bar{u} - \bar{u}^2\bar{v}) - \gamma\cdot\bar{a}), \quad \text{on } \Omega\times[0,T],
\end{aligned} \tag{A.11}$$

$$(\delta v)_t - D_v\Delta(\delta v) + \gamma(\bar{u}^2\cdot\delta v + 2\bar{u}\bar{v}\cdot\delta u) - \gamma\delta b = -(\bar{v}_t - D_v\Delta\bar{v} + \gamma\bar{u}^2\bar{v} - \gamma\cdot\bar{b}), \quad \text{on } \Omega\times[0,T], \tag{A.12}$$

$$(\delta u)(\mathbf{x},0) = (\delta v)(\mathbf{x},0) = 0, \quad \text{on } \Omega, \tag{A.13}$$

$$\frac{\partial(\delta u)}{\partial \nu} = \frac{\partial(\delta v)}{\partial \nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \tag{A.14}$$

Applying the same substitutions to the adjoint equations gives

$$
\begin{aligned}
-(\bar{p}+\delta p)_t - D_u\Delta(\bar{p}+\delta p) + 2\gamma\bar{u}\bar{v}((\bar{q}+\delta q)-(\bar{p}+\delta p)) + \gamma(\bar{p}+\delta p) &= \beta_1((\bar{u}+\delta u)-\widehat{u}), \quad \text{on } \Omega\times[0,T],\\
-(\bar{q}+\delta q)_t - D_v\Delta(\bar{q}+\delta q) + \gamma\bar{u}^2((\bar{q}+\delta q)-(\bar{p}+\delta p)) &= \beta_2((\bar{v}+\delta v)-\widehat{v}), \quad \text{on } \Omega\times[0,T],\\
(\bar{p}+\delta p)(\mathbf{x},T) &= \beta_{T,1}((\bar{u}+\delta u)(\mathbf{x},T)-\widehat{u}(\mathbf{x},T)), \quad \text{on } \Omega,\\
(\bar{q}+\delta q)(\mathbf{x},T) &= \beta_{T,2}((\bar{v}+\delta v)(\mathbf{x},T)-\widehat{v}(\mathbf{x},T)), \quad \text{on } \Omega,\\
\frac{\partial(\bar{p}+\delta p)}{\partial\nu} = \frac{\partial(\bar{q}+\delta q)}{\partial\nu} &= 0, \quad \text{on } \partial\Omega\times[0,T],
\end{aligned}
$$

which may then be expanded and simplified to give

$$
\begin{aligned}
&-(\delta p)_t - D_u\Delta(\delta p) + 2\gamma(\bar{v}\bar{q}\cdot\delta u + \bar{u}\bar{q}\cdot\delta v + \bar{u}\bar{v}\cdot\delta q - \bar{v}\bar{p}\cdot\delta u - \bar{u}\bar{p}\cdot\delta v - \bar{u}\bar{v}\cdot\delta p) + \gamma\delta p - \beta_1\delta u\\
&\quad = \beta_1(\bar{u}-\widehat{u}) - (-\bar{p}_t - D_u\Delta\bar{p} + 2\gamma\bar{u}\bar{v}(\bar{q}-\bar{p}) + \gamma\bar{p}), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.15)}\\
&-(\delta q)_t - D_v\Delta(\delta q) + \gamma(\bar{u}^2\cdot\delta q + 2\bar{u}\bar{q}\cdot\delta u - \bar{u}^2\delta p - 2\bar{u}\bar{p}\cdot\delta u) - \beta_2\delta v\\
&\quad = \beta_2(\bar{v}-\widehat{v}) - (-\bar{q}_t - D_v\Delta\bar{q} + \gamma\bar{u}^2(\bar{q}-\bar{p})), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.16)}\\
&(\delta p)(\mathbf{x},T) = \beta_{T,1}(\delta u)(\mathbf{x},T), \quad (\delta q)(\mathbf{x},T) = \beta_{T,2}(\delta v)(\mathbf{x},T), \quad \text{on } \Omega, \qquad \text{(A.17)}\\
&\frac{\partial(\delta p)}{\partial\nu} = \frac{\partial(\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \qquad \text{(A.18)}
\end{aligned}
$$

The forward and adjoint equations can be derived by differentiating the Lagrangian

$$
\begin{aligned}
\mathcal{J}_{GM2}(u,v,a,b,p,q) = &\frac{\beta_1}{2}\|u-\widehat{u}\|^2_{L_2(\Omega\times[0,T])} + \frac{\beta_2}{2}\|v-\widehat{v}\|^2_{L_2(\Omega\times[0,T])}\\
&+ \frac{\beta_{T,1}}{2}\|u-\widehat{u}_T\|^2_{L_2(\Omega)} + \frac{\beta_{T,2}}{2}\|v-\widehat{v}_T\|^2_{L_2(\Omega)}\\
&+ \frac{\nu_1}{2}\|a\|^2_{L_2(\Omega\times[0,T])} + \frac{\nu_2}{2}\|b\|^2_{L_2(\Omega\times[0,T])}\\
&- \int_{\Omega\times[0,T]} p\left(u_t - D_u\Delta u + \gamma(u-u^2v) - \gamma a\right)\\
&- \int_{\Omega\times[0,T]} q\left(v_t - D_v\Delta v + \gamma u^2 v - \gamma b\right),
\end{aligned}
$$

with respect to u, v, p and q, similarly as for the $GM1$ model. The gradient equations for this problem may be derived by differentiating this Lagrangian with respect to the control variables a and b, which gives the conditions

$$
\int_{\Omega\times[0,T]} (\nu_1 a + \gamma p) = 0, \qquad \int_{\Omega\times[0,T]} (\nu_2 b + \gamma q) = 0.
$$

Applying Newton iteration to these equations gives

$$
\int_{\Omega\times[0,T]} (\nu_1\delta a + \gamma\delta p) = -\int_{\Omega\times[0,T]} (\nu_1\bar{a} + \gamma\bar{p}), \qquad \text{(A.19)}
$$

$$
\int_{\Omega\times[0,T]} (\nu_2\delta b + \gamma\delta q) = -\int_{\Omega\times[0,T]} (\nu_2\bar{b} + \gamma\bar{q}), \qquad \text{(A.20)}
$$

at each Newton step.

Hence the system of equations which need to be solved at each Newton step are the adjoint equations (A.15)–(A.18), the gradient equations (A.19) and (A.20), and the forward equations (A.11)–(A.14).

References

[1] O. Axelsson, J. Karátson, Equivalent operator preconditioning for elliptic problems, Numer. Algorithms 50 (2009) 297–380.
[2] A. Badugu, C. Kraemer, P. Germann, D. Menshykau, D. Iber, Digit patterning during limb development as a result of the BMP-receptor interaction, Sci. Rep. 991 (2012) 1–13.
[3] W. Bangerth, R. Hartmann, G. Kanschat, deal.II—a general-purpose object-oriented finite element library, ACM Trans. Math. Softw. 33 (2007), Art. 24, 27 pp.
[4] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[5] M. Benzi, E. Haber, L. Taralli, A preconditioning technique for a class of PDE-constrained optimization problems, Adv. Comput. Math. 35 (2011) 149–173.

[6] V. Castets, E. Dulos, J. Boissonade, P.D. Kepper, Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern, Adv. Comput. Math. 35 (2011) 2953–2956.
[7] I.S. Duff, A.M. Erisman, J.K. Reid, Direct Methods for Sparse Matrices, Monogr. Numer. Anal., The Clarendon Press, Oxford University Press, New York, 1989.
[8] A.D. Economou, A. Ohazama, T. Porntaveetus, P.T. Sharpe, S. Kondo, M.A. Basson, A. Gritli-Linde, M.T. Coburne, J.B.A. Green, Periodic stripe formation by a Turing mechanism operating at growth zones in the mammalian palate, Nat. Genet. 44 (2012) 348–351.
[9] H.C. Elman, D.J. Silvester, A.J. Wathen, Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics, Numer. Math. Sci. Comput., Oxford University Press, New York, 2005.
[10] R. Fletcher, Conjugate gradient methods for indefinite systems, in: Numerical Analysis, Proc. 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975, in: Lect. Notes Math., vol. 506, Springer, Berlin, 1976, pp. 73–89.
[11] M.R. Garvie, P.K. Maini, C. Trenchea, An efficient and robust numerical algorithm for estimating parameters in Turing systems, J. Comput. Phys. 229 (2010) 7058–7071.
[12] M.R. Garvie, C. Trenchea, Identification of space–time distributed parameters in the Gierer–Meinhardt reaction–diffusion system, SIAM J. Appl. Math. 74 (2014) 147–166.
[13] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, M.G. Sala, ML 5.0 smoothed aggregation user's guide, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.
[14] A. Gierer, H. Meinhardt, A theory of biological pattern formation, Biol. Cybern. 12 (1972) 30–39.
[15] G.H. Golub, C.F.V. Loan, Matrix Computations, third ed., Johns Hopkins Stud. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 1996.
[16] E. Haber, U.M. Ascher, D. Oldenburg, On optimization techniques for solving nonlinear inverse problems, Inverse Probl. 16 (2000) 1263–1280.
[17] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (1952) 409–436, 1953.
[18] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints, Math. Model. Theor. Appl., Springer-Verlag, New York, 2009.
[19] C. Hogea, C. Davatzikos, G. Biros, An image-driven parameter estimation problem for a reaction–diffusion glioma growth model with mass effects, J. Math. Biol. 56 (2008) 793–825.
[20] K. Ito, K. Kunisch, Lagrange Multiplier Approach to Variational Problems and Applications, Adv. Des. Control, vol. 15, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008.
[21] K. Ito, K. Kunisch, V. Schulz, I. Gherman, Approximate nullspace iterations for KKT systems, SIAM J. Matrix Anal. Appl. 31 (2010) 1835–1847.
[22] Y.A. Kuznetsov, Efficient iterative solvers for elliptic finite element problems on nonmatching grids, Russ. J. Numer. Anal. Math. Model. 10 (1995) 187–211.
[23] H.P. Langtangen, Computational Partial Differential Equations. Numerical Methods and Diffpack Programming, second ed., Springer, Berlin, 2003.
[24] R.T. Liu, S.S. Liaw, P.K. Maini, Two-stage Turing model for generating pigment patterns on the leopard and the jaguar, Phys. Rev. E 74 (2006) 011914.
[25] K.A. Mardal, R. Winther, Preconditioning discretizations of systems of partial differential equations, Numer. Linear Algebra Appl. 18 (2011) 1–40.
[26] M.F. Murphy, G.H. Golub, A.J. Wathen, A note on preconditioning for indefinite linear systems, SIAM J. Sci. Comput. 21 (2000) 1969–1972.
[27] J.D. Murray, Mathematical Biology, vol. 2: Spatial Models and Biomedical Applications, third ed., Springer, New York, NY, 2003.
[28] J. Nocedal, S.J. Wright, Numerical Optimization, Springer Ser. Oper. Res., Springer-Verlag, New York, 1999.
[29] J. Nocedal, S.J. Wright, Numerical Optimization, second ed., Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.
[30] Q. Ouyang, H.L. Swinney, Transition from a uniform state to hexagonal and striped Turing patterns, Nature 352 (1991) 610–612.
[31] C.C. Paige, M.A. Saunders, Solutions of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (1975) 617–629.
[32] J.W. Pearson, M. Stoll, Fast iterative solution of reaction–diffusion control problems arising from chemical processes, SIAM J. Sci. Comput. 35 (2013) B987–B1009.
[33] J.W. Pearson, M. Stoll, A.J. Wathen, Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems, SIAM J. Matrix Anal. Appl. 33 (2012) 1126–1152.
[34] J.W. Pearson, A.J. Wathen, A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, Numer. Linear Algebra Appl. 19 (2012) 816–829.
[35] J.W. Pearson, A.J. Wathen, Fast iterative solvers for convection–diffusion control problems, Electron. Trans. Numer. Anal. 40 (2013) 294–310.
[36] T. Rusten, R. Winther, A preconditioned iterative method for saddle point problems, SIAM J. Matrix Anal. Appl. 13 (1992) 887–904.
[37] Y. Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
[38] A. Schiela, S. Ulbrich, Operator preconditioning for a class of inequality constrained optimal control problems, SIAM J. Optim. 24 (2014) 435–466.
[39] J. Schnakenberg, Simple chemical reaction systems with limit cycle behaviour, J. Theor. Biol. 81 (1979) 389–400.
[40] R. Sheth, L. Marcon, M.F. Bastida, M. Junco, L. Quintana, R. Dahn, M. Kmita, J. Sharpe, M.A. Ros, Hox genes regulate digit patterning by controlling the wavelength of a Turing-type mechanism, Science 338 (2012) 1476–1480.
[41] F. Tröltzsch, Optimal Control of Partial Differential Equations: Theory, Methods and Applications, American Mathematical Society, 2010.
[42] A. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. B 237 (1952) 37–72.
[43] A.J. Wathen, T. Rees, Chebyshev semi-iteration in preconditioning for problems including the mass matrix, Electron. Trans. Numer. Anal. 34 (2008) 125–135.
[44] S.J. Wright, Primal–Dual Interior-Point Methods, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.
[45] W. Zulehner, Non-standard norms and robust estimates for saddle point problems, SIAM J. Matrix Anal. Appl. 32 (2011) 536–560.