FAST SOLVERS FOR OPTIMAL CONTROL PROBLEMS FROM PATTERN FORMATION

MARTIN STOLL[*], JOHN W. PEARSON[†], AND PHILIP K. MAINI[‡]

Abstract. The modelling 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.

Key words. PDE-constrained optimization, reaction-diffusion, pattern formation, Newton iteration, preconditioning, Schur complement.

AMS subject classifications. 65F08, 65F10, 65F50, 92-08, 93C20

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 [36] 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 [23]) and have been shown to exist in chemistry [5, 26]. While their applicability to biology remains controversial, there are many examples which suggest that Turing systems may be underlying key patterning processes (see [1, 7, 34] for the most recent examples). Two important models which embody the essence of the original Turing model are the Gierer-Meinhardt [13] and Schnakenberg models [33] and it is upon these models which we focus.[1] In light of the fact that to date, no Turing morphogens have 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 optimal control problems (see [35] 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 [10, 11], and so it is natural to explore this particular application here.

[*] Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de),

[†] School of Mathematics, The University of Edinburgh, James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ, UK (j.pearson@ed.ac.uk),

[‡] Wolfson Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Radcliffe Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, UK (maini@maths.ox.ac.uk)

[1] Although the second model is commonly referred to as the Schnakenberg model, it was actually first proposed by Gierer and Meinhardt in [13] along with the model usually referenced as the Gierer-Meinhardt model – we therefore refer to the first and second models as '$GM1$' and '$GM2$' within our working.

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 of the $(1,1)$-block and Schur complement of these matrix systems. The solvers will incorporate aspects of iterative solution strategies developed by the first and second authors to tackle simpler optimal control problems in literature such as [28, 29, 30, 31].

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 [19, 35]. 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 [13]

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

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.

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 [13, 23]

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

and the $GM2$ model [23, 33]

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

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 [13, 23, 33]. 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 $D_u < D_v$ [13]. The parameters r and γ are positive parameters: the value r in the $GM1$ model denotes the (small) production rate of the activator [13], and the parameter γ in the $GM2$ 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.

Throughout the remainder of this article we will consider the minimization of the cost functional (2.1), with PDE constraints taking the form of the $GM1$ model or the $GM2$ model. PDE-constrained optimization problems of similar form have been considered in the literature, such as in [10, 11].

As the optimization problem $\min_{(u,v,a,b)} J(u,v,a,b)$ subject to (2.2) or (2.3) is nonlinear due to the nature of the constraints, we have to apply nonlinear programming [25] algorithms. Many of these are generalizations of Newton's method [25]. 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 [3, 8], 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 [8, 32], 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 [35]. In our case with a nonlinear constraint we apply a nonlinear solver to the first order conditions. Hence, we start by deriving the first order conditions or Karush-Kuhn-Tucker conditions of the Lagrangian

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

where $\mathcal{R}_1(u,v,a,b)$ and $\mathcal{R}_2(u,v,a,b)$ represent the first two equations of both $GM1$ and $GM2$ models. Note that for convenience our Lagrangian ignores the boundary and initial conditions. In general form the first order conditions are given by

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

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 [24]. 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 at this point. At this stage we simply state the systems obtained for both $GM1$ and $GM2$ 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 [15] is often used as the corresponding system matrix in (3.1) – this ignores the mixed derivatives with respect to the primal variables, i.e.,

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

where the matrices denoted by $\widehat{\mathcal{L}}_{:,:}$ do not contain second derivative information (see [15, 25] 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

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

where

$$\begin{aligned} \mathbf{A}_{u,GM1} &= \tau\beta_1\mathbf{M} + \beta_{T,1}\mathbf{M}_T + 2\tau r\mathbf{M}_{p/v} + 2\tau r\mathbf{M}_q, \\ \mathbf{A}_{v,GM1} &= \tau\beta_2\mathbf{M} + \beta_{T,2}\mathbf{M}_T + 2\tau r\mathbf{M}_{u^2p/v^3}, \\ \mathbf{L}_{u,GM1} &= \mathbf{M}_E + \tau D_u\mathbf{K} - 2\tau r\mathbf{M}_{u/v} + \tau\mathbf{M}_a, \\ \mathbf{L}_{v,GM1} &= \mathbf{M}_E + \tau D_v\mathbf{K} + \tau\mathbf{M}_b. \end{aligned}$$

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, where ϕ_i denote the finite element basis functions used. Furthermore, the matrix $\mathbf{K} = \text{blkdiag}(K, \ldots, K)$. The vector $\mathbf{f}$ is the discrete representation at each Newton step of

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

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

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

with all matrices as previously defined (see [4, 15] 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 favourable properties from an iterative solver point-of-view.

Moving on to the $GM2$ model, Appendix A reveals the following structure of the Newton system:

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

with

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

and $\mathbf{g}$ the discrete representation of

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

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 [29]. 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 saddle point form is a topic of major interest within the numerical analysis community [3, 8]. Due to the vast dimensionality of the systems derived earlier we cannot use factorization-based approaches [6]. We hence employ a Krylov subspace method [32] where we construct a Krylov subspace of the form

$$\operatorname{span}\left\{\mathbf{r}_0, \mathcal{A}\mathbf{r}_0, \mathcal{A}^2\mathbf{r}_0, \ldots\right\},$$

within which we seek an approximation to the solution of the 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, i.e., 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. As for our problem the eigenvalues of $\mathcal{A}$ depend on the mesh-parameter and all the other parameters describing the PDE and the objective function, the convergence can be very slow. The goal is hence to find a preconditioning matrix $\mathcal{P}$ such that we can solve the equivalent preconditioned system

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

For a saddle point problem in the form of

$$\left[\begin{array}{cc} A & B^T \\ B & 0 \end{array}\right],$$

this is typically achieved by preconditioners of the form

$$\mathcal{P} = \left[\begin{array}{cc} \widetilde{A} & 0 \\ 0 & \widetilde{S} \end{array}\right], \quad \mathcal{P} = \left[\begin{array}{cc} \widetilde{A} & 0 \\ B & -\widetilde{S} \end{array}\right],$$

where $\widetilde{A}$ approximates the $(1,1)$-block of the saddle point matrix $\mathcal{A}$ and $\widetilde{S}$ approximates the (negative) Schur complement $BA^{-1}B^T$. This is motivated by results obtained in [20, 22] where it is shown that the exact preconditioners $\widetilde{A} = A$ and $\widetilde{S} = BA^{-1}B^T$ lead to a very small number of eigenvalues 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 [27] based on a three-term recurrence relation. However as Minres typically needs 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 [9], which is the extension of cg [16] to nonsymmetric problems and is based on the nonsymmetric Lanczos process [14].

In the following working we wish to derive preconditioners for all of the above linear systems. For the $GM1$ model the matrix A is approximated as follows:

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

where $\widetilde{A}_1$ and $\widetilde{A}_2$ are (double) Schur complement approximations defined by

$$\begin{aligned} \widetilde{\mathbf{A}}_1 &= \mathbf{A}_{u,GM1} - (2\tau r)^2\mathbf{M}_{up/v^2}\mathbf{A}_{v,GM1}^{-1}\mathbf{M}_{up/v^2} - \tau\nu_1^{-1}\mathbf{M}_p\mathbf{M}^{-1}\mathbf{M}_p, \\ \widetilde{\mathbf{A}}_2 &= \mathbf{A}_{v,GM1} - \tau\nu_2^{-1}\mathbf{M}_q\mathbf{M}^{-1}\mathbf{M}_q. \end{aligned} \tag{4.1}$$

We then use $\widetilde{A}$ within the approximation of the Schur complement, i.e.,

$$\begin{aligned} S \approx B\widetilde{A}^{-1}B^T = &\begin{bmatrix} -\mathbf{L}_{u,GM1} & -\tau r\mathbf{M}_{u^2/v^2} \\ 2\tau r\mathbf{M}_u & -\mathbf{L}_{v,GM1} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}}_1^{-1} & 0 \\ 0 & \widetilde{\mathbf{A}}_2^{-1} \end{bmatrix} \begin{bmatrix} -\mathbf{L}_{u,GM1}^T & 2\tau r\mathbf{M}_u \\ -\tau r\mathbf{M}_{u^2/v^2} & -\mathbf{L}_{v,GM1}^T \end{bmatrix} \\ &+ \begin{bmatrix} \tau\mathbf{M}_u & 0 \\ 0 & \tau\mathbf{M}_v \end{bmatrix} \begin{bmatrix} (\tau\nu_1\mathbf{M})^{-1} & 0 \\ 0 & (\tau\nu_2\mathbf{M})^{-1} \end{bmatrix} \begin{bmatrix} \tau\mathbf{M}_u & 0 \\ 0 & \tau\mathbf{M}_v \end{bmatrix}, \end{aligned} \tag{4.2}$$

which we approximate using

$$\widetilde{S} = \begin{bmatrix} \mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)} & \tau r\mathbf{M}_{u^2/v^2} \\ -2\tau r\mathbf{M}_u & \mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}}_1^{-1} & 0 \\ 0 & \widetilde{\mathbf{A}}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)} & -2\tau r\mathbf{M}_u \\ \tau r\mathbf{M}_{u^2/v^2} & \mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix},$$

with $\widehat{\mathbf{M}}_1^{(1)} := \sqrt{\frac{\tau}{\nu_1}}\mathbf{M}_u\mathbf{M}^{-1/2}\widetilde{\mathbf{A}}_1$, $\widehat{\mathbf{M}}_1^{(2)} := \sqrt{\frac{\tau}{\nu_1}}\mathbf{M}^{-1/2}\mathbf{M}_u$, $\widehat{\mathbf{M}}_2^{(1)} := \sqrt{\frac{\tau}{\nu_2}}\mathbf{M}_v\mathbf{M}^{-1/2}\widetilde{\mathbf{A}}_2$, and $\widehat{\mathbf{M}}_2^{(2)} := \sqrt{\frac{\tau}{\nu_2}}\mathbf{M}^{-1/2}\mathbf{M}_v$. These approximations are based on a 'matching strategy' to incorporate both terms of Equation (4.2) into the Schur complement approximation [29, 30, 31]. For any practical method we are only interested in the inverse of the Schur complement approximation

$$\widetilde{S}^{-1} = \begin{bmatrix} \mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)} & -2\tau r\mathbf{M}_u \\ \tau r\mathbf{M}_{u^2/v^2} & \mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix}^{-1} \begin{bmatrix} \widetilde{\mathbf{A}}_1 & 0 \\ 0 & \widetilde{\mathbf{A}}_2 \end{bmatrix} \begin{bmatrix} \mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)} & \tau r\mathbf{M}_{u^2/v^2} \\ -2\tau r\mathbf{M}_u & \mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix}^{-1},$$

where we can evaluate the inverse of the first and last block by a fixed number of steps of a Uzawa method [32] with block diagonal (possibly block triangular) preconditioner

$$\mathcal{P}_{BD}^{-1} = \text{blkdiag}\left((\mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)})_{AMG}, (\mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(2)})_{AMG} \right),$$

with $(\cdot)_{AMG}$ denoting the application of an algebraic multigrid (AMG) 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 lumped mass matrices. If consistent mass matrices are used instead, some approximation such as the application of Chebyshev semi-iteration [37] is chosen. The inverse of the Schur complement approximation

$$\begin{bmatrix} \mathbf{L}_{u,GM1}^T + \widehat{\mathbf{M}}_1^{(2)} & -2\tau r\mathbf{M}_u \\ \tau r\mathbf{M}_{u^2/v^2} & \mathbf{L}_{v,GM1}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix} \begin{bmatrix} \beta_1\tau\mathbf{M} & 0 \\ 0 & \beta_2\tau\mathbf{M} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{L}_{u,GM1} + \widehat{\mathbf{M}}_1^{(1)} & \tau r\mathbf{M}_{u^2/v^2} \\ -2\tau r\mathbf{M}_u & \mathbf{L}_{v,GM1} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix},$$

with $\widehat{\mathbf{M}}_1^{(1)} = \widehat{\mathbf{M}}_1^{(2)} := \tau\sqrt{\frac{\beta_1}{\nu_1}}\mathbf{M}_u$, 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.

In a completely analogous way we can derive preconditioners for the $GM2$ model. We approximate the matrix A as follows:

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

with $\widetilde{\mathbf{A}}_1 = \mathbf{A}_{u,GM2} - (2\tau\gamma)^2\mathbf{M}_{u(q-p)}\mathbf{A}_{v,GM2}^{-1}\mathbf{M}_{u(q-p)}$. We follow a similar strategy as before to approximate the Schur complement

$$\begin{aligned} B\widetilde{A}^{-1}B^T = &\begin{bmatrix} -\mathbf{L}_{u,GM2} & \tau\gamma\mathbf{M}_{u^2} \\ -2\tau\gamma\mathbf{M}_{uv} & -\mathbf{L}_{v,GM2} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}}_1^{-1} & 0 \\ 0 & \widetilde{\mathbf{A}}_2^{-1} \end{bmatrix} \begin{bmatrix} -\mathbf{L}_{u,GM2}^T & -2\tau\gamma\mathbf{M}_{uv} \\ \tau\gamma\mathbf{M}_{u^2} & -\mathbf{L}_{v,GM2}^T \end{bmatrix} \\ &+ \begin{bmatrix} \tau\gamma\mathbf{M} & 0 \\ 0 & \tau\gamma\mathbf{M} \end{bmatrix} \begin{bmatrix} (\tau\nu_1\mathbf{M})^{-1} & 0 \\ 0 & (\tau\nu_2\mathbf{M})^{-1} \end{bmatrix} \begin{bmatrix} \tau\gamma\mathbf{M} & 0 \\ 0 & \tau\gamma\mathbf{M} \end{bmatrix}. \end{aligned}$$

Again using the matching strategy from [29, 30, 31] we obtain the following approximation

$$\widetilde{S} = \begin{bmatrix} \mathbf{L}_{u,GM2} + \widehat{\mathbf{M}}_1^{(1)} & -\tau\gamma\mathbf{M}_{u^2} \\ 2\tau\gamma\mathbf{M}_{uv} & \mathbf{L}_{v,GM2} + \widehat{\mathbf{M}}_2^{(1)} \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{A}}_1^{-1} & 0 \\ 0 & \widetilde{\mathbf{A}}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{L}_{u,GM2}^T + \widehat{\mathbf{M}}_1^{(2)} & 2\tau\gamma\mathbf{M}_{uv} \\ -\tau\gamma\mathbf{M}_{u^2} & \mathbf{L}_{v,GM2}^T + \widehat{\mathbf{M}}_2^{(2)} \end{bmatrix},$$

with $\widehat{\mathbf{M}}_1^{(1)} = \sqrt{\frac{\tau}{\nu_1}}\gamma\mathbf{M}^{1/2}\widetilde{\mathbf{A}}_1$, $\widehat{\mathbf{M}}_1^{(2)} = \sqrt{\frac{\tau}{\nu_1}}\gamma\mathbf{M}^{1/2}$, $\widehat{\mathbf{M}}_2^{(1)} = \sqrt{\frac{\tau}{\nu_2}}\gamma\mathbf{M}^{1/2}\widetilde{\mathbf{A}}_2$, and $\widehat{\mathbf{M}}_2^{(2)} = \sqrt{\frac{\tau}{\nu_2}}\gamma\mathbf{M}^{1/2}$.

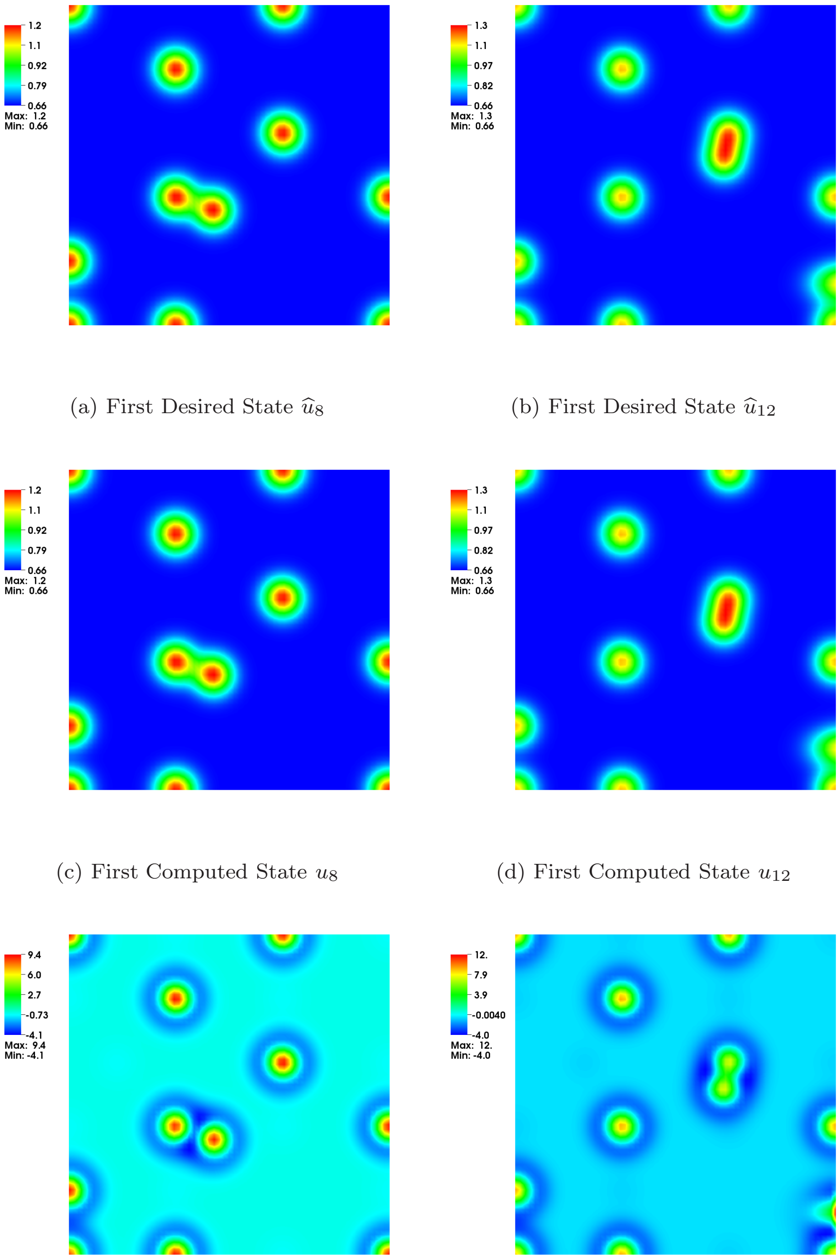
In each of our suggested iterative methods, we insert our approximations of A and $BA^{-1}B^T$ into general preconditioners for saddle point systems stated in (4.1).

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 given algorithms and preconditioners within the deal.II [2] framework using $Q1$ finite elements. The AMG preconditioner we use is part of the Trilinos ML package [12] 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. 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 @ 2.67GHz CPUs and 48GB of RAM.

5.1. *GM2* model. For both $GM2$ and $GM1$ 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 Figure 5.1 we illustrate two instances of the desired state and computed results for the $GM2$ 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 Figure 5.1 where the control is shown with sometimes rather high values. In Table 5.1 we present iteration numbers for solving this test problem for a range of degrees of freedom and regularization parameters.

DoF		Bicg		Bicg		Bicg
		$\nu_1 = \nu_2 = 1e-2$		$\nu_1 = \nu_2 = 1e-4$		$\nu_1 = \nu_2 = 1e-6$
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		

Table 5.1: Results on unit square with $D_u = 1$, $D_v = 10$, $\beta_1 = \beta_2 = 1$, and $\gamma = 50$.



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

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

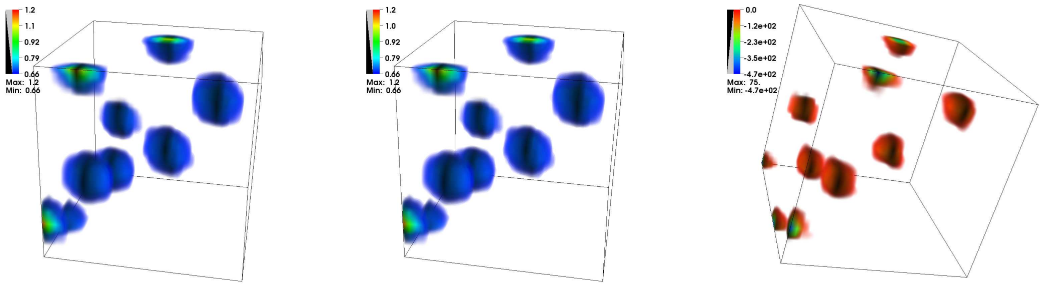
(c) First Computed State u_8

(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 $\nu_1 = \nu_2 = 10^{-6}$.



(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 $GM1$ model with parameters at $\beta_1 = 10^2$, $\beta_2 = 10^2$, $\nu_1 = 10^{-2}$, $\nu_2 = 10^{-2}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$.

5.2. *GM*1 model with Newton and Gauss-Newton methods. For the next problem we examine, the desired state for the $GM1$ model is created using Gaussian functions placed in the three-dimensional unit cube. This is illustrated in Figure 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 $\beta_1 = 10^2$, $\beta_2 = 10^2$, $\nu_1 = 10^{-2}$, $\nu_2 = 10^{-2}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. For many interesting parameter setups (including for a range of values 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 iteration numbers achieved for a range of problems in Table 5.2.

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 [17] 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 [28] for more details on the derivation of the non-smooth Newton system and the choice of preconditioner. In Table 5.3 we present some results for the setup $0 \leq a$ and $0 \leq b$, where the Gauss-Newton scheme is used in conjunction with Bicg.

5.3. Image-driven desired state and *GM*1 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 $GM1$ and $GM2$ models, whether or not control constraints are included. Image-driven parameter estimation techniques can also be found in [18]. For this problem, we choose to take an image similar to those used in [21] – 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 the relevant index in time. The results for applying the Gauss-Newton scheme to this image-driven problem are shown in Figure 5.3.

In Figure 5.3b the desired state is shown. The computed state is shown in Figure 5.3a and the associated control in Figure 5.3c.

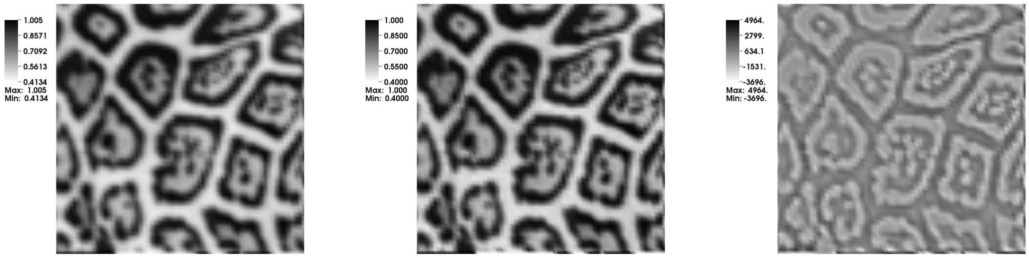
DoF		Bicg		Bicg		Bicg
		$\nu_1 = \nu_2 = 1e-2$		$\nu_1 = \nu_2 = 1e-4$		$\nu_1 = \nu_2 = 1e-6$
43740	step 1	11	step 1	11	step 1	9
	step 2	11	step 2	11	step 2	9
	step 3	11	step 3	11	step 3	9
	step 4	11	step 4	11		
	step 5	11	step 5	11		
294780	step 1	11	step 1	13	step 1	11
	step 2	11	step 2	12	step 2	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		
2156220	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		

Table 5.2: Results on unit cube $\beta_1 = 10^2$, $\beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2.

DoF		Bicg		Bicg
		$\nu_1 = \nu_2 = 1e-2$		$\nu_1 = \nu_2 = 1e-4$
253500	step 1	8	step 1	8
	step 2	11	step 2	15
	step 3	11	step 3	15
998460	step 1	16	step 1	16
	step 2	26	step 2	23
	step 3	20	step 3	55
3962940	step 1	30	step 1	30
	step 2	31	step 2	35
	step 3	31	step 3	34

Table 5.3: Results on unit cube $\beta_1 = 10^2$, $\beta_2 = 10^2$, $D_u = 1$, $D_v = 10$, and $r = 10^{-2}$. We here vary the mesh-size and the regularization parameters ν_1 and ν_2.

The parameters for this setup are $\beta_1 = 10^2$, $\beta_2 = 10^2$, $\nu_1 = 10^{-7}$, $\nu_2 = 10^{-7}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-5}$. For the computations from which Figure 5.3 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.



(a) Computed State

(b) Desired State

(c) Computed Control

Fig. 5.3: Results for image-driven model: Shown are computed state, desired state, and computed control for the parameter setups using $\beta_1 = 10^2$, $\beta_2 = 10^2$, $\nu_1 = 10^{-7}$, $\nu_2 = 10^{-7}$, $D_u = 1$, $D_v = 10$, and $r = 10^{-5}$.

Overall the numerical results presented for the above test problems indicate that we are able to solve a wide range of parameter identification problems from pattern formation, with our observed 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 $GM1$ and $GM2$ models) indicate that out 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 for 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. 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. The authors are grateful for the award of a European Science Foundation (ESF) Exchange Grant 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 [10])

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

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

$$
\begin{aligned}
(\delta u)_t - D_u\Delta(\delta u) + r\frac{\bar{u}^2\cdot\delta v - 2\bar{u}\bar{v}\cdot\delta u}{\bar{v}^2} + \bar{u}\cdot\delta a + \bar{a}\cdot\delta u& \qquad (A.1)\\
= r - \left(\bar{u}_t - D_u\Delta\bar{u} - \frac{r\bar{u}^2}{\bar{v}} + \bar{a}\bar{u}\right),& \quad \text{on } \Omega\times[0,T],\\
(\delta v)_t - D_v\Delta(\delta v) - 2r\bar{u}\cdot\delta u + \bar{v}\cdot\delta b + \bar{b}\cdot\delta v& \qquad (A.2)\\
= -(\bar{v}_t - D_v\Delta\bar{v} - r\bar{u}^2 + \bar{b}\bar{v}),& \quad \text{on } \Omega\times[0,T],\\
(\delta u)(\mathbf{x},0) = 0, \quad (\delta v)(\mathbf{x},0) = 0,& \quad \text{on } \Omega, \qquad (A.3)\\
\frac{\partial(\delta u)}{\partial\nu} = \frac{\partial(\delta v)}{\partial\nu} = 0,& \quad \text{on } \partial\Omega\times[0,T]. \qquad (A.4)
\end{aligned}
$$

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

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

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

$$\begin{aligned} &-(\delta p)_t - D_u\Delta(\delta p) - 2r\frac{\bar{u}\bar{p}\cdot\delta v - \bar{v}\bar{p}\cdot\delta u - \bar{u}\bar{v}\cdot\delta p}{\bar{v}^2} \\ &\qquad + \bar{p}\cdot\delta a + \bar{a}\cdot\delta p - 2r(\bar{u}\cdot\delta q + \bar{q}\cdot\delta u) - \beta_1\delta u \\ &\qquad = \beta_1(\bar{u}-\widehat{u}) - \left(-\bar{p}_t - D_u\Delta\bar{p} - 2r\frac{\bar{u}}{\bar{v}}\bar{p} + \bar{a}\bar{p} - 2r\bar{u}\bar{q}\right), \quad \text{on } \Omega\times[0,T], \end{aligned} \tag{A.5}$$

$$\begin{aligned} &-(\delta q)_t - D_v\Delta(\delta q) + r\bar{u}\frac{2\bar{v}\bar{p}\cdot\delta u + \bar{u}\bar{v}\cdot\delta p - 2\bar{u}\bar{p}\cdot\delta v}{\bar{v}^2} + \bar{q}\cdot\delta b + \bar{b}\cdot\delta q - \beta_2\delta v \\ &\qquad = \beta_2(\bar{v}-\widehat{v}) - \left(-\bar{q}_t - D_v\Delta\bar{q} + r\frac{\bar{u}^2}{\bar{v}^2}\bar{p} + \bar{b}\bar{q}\right), \quad \text{on } \Omega\times[0,T], \end{aligned} \tag{A.6}$$

$$(\delta p)(\mathbf{x},T) = \beta_{T,1}(\delta u)(\mathbf{x},T), \quad (\delta q)(\mathbf{x},T) = \beta_{T,2}(\delta v)(\mathbf{x},T), \quad \text{on } \Omega, \tag{A.7}$$

$$\frac{\partial(\delta p)}{\partial\nu} = \frac{\partial(\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \tag{A.8}$$

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

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

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 will give 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 will 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 [10])

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

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

$$\begin{aligned} (\delta u)_t - D_u \Delta(\delta u) + \gamma(\delta u - \bar{u}^2 \cdot \delta v - 2\bar{u}\bar{v} \cdot \delta u) - \gamma \delta a & \qquad \text{(A.11)} \\ = -(\bar{u}_t - D_u \Delta \bar{u} + \gamma(\bar{u} - \bar{u}^2 \bar{v}) - \gamma \bar{a}), &\quad \text{on } \Omega \times [0,T], \\ (\delta v)_t - D_v \Delta(\delta v) + \gamma(\bar{u}^2 \cdot \delta v + 2\bar{u}\bar{v} \cdot \delta u) - \gamma \delta b & \qquad \text{(A.12)} \\ = -(\bar{v}_t - D_v \Delta \bar{v} + \gamma \bar{u}^2 \bar{v} - \gamma \bar{b}), &\quad \text{on } \Omega \times [0,T], \\ (\delta u)(\mathbf{x},0) = 0, \quad (\delta v)(\mathbf{x},0) = 0, \quad \text{on } \Omega, & \qquad \text{(A.13)} \\ \frac{\partial(\delta u)}{\partial \nu} = \frac{\partial(\delta v)}{\partial \nu} = 0, \quad \text{on } \partial\Omega \times [0,T]. & \qquad \text{(A.14)} \end{aligned}$$

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

$$\begin{aligned}
&-(\delta p)_t - D_u\Delta(\delta p) + 2\gamma(\bar{v}\bar{q}\cdot\delta u + \bar{u}\bar{q}\cdot\delta v + \bar{u}\bar{v}\cdot\delta q \\
&\qquad - \bar{v}\bar{p}\cdot\delta u - \bar{u}\bar{p}\cdot\delta v - \bar{u}\bar{v}\cdot\delta p) + \gamma\delta p - \beta_1\delta u \\
&\qquad = \beta_1(\bar{u}-\widehat{u}) - (-\bar{p}_t - D_u\Delta\bar{p} + 2\gamma\bar{u}\bar{v}(\bar{q}-\bar{p}) + \gamma\bar{p}), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.15)}
\end{aligned}$$

$$\begin{aligned}
&-(\delta q)_t - D_v\Delta(\delta q) + \gamma(\bar{u}^2\cdot\delta q + 2\bar{u}\bar{q}\cdot\delta u - \bar{u}^2\delta p - 2\bar{u}\bar{p}\cdot\delta u) - \beta_2\delta v \\
&\qquad = \beta_2(\bar{v}-\widehat{v}) - (-\bar{q}_t - D_v\Delta\bar{q} + \gamma\bar{u}^2(\bar{q}-\bar{p})), \quad \text{on } \Omega\times[0,T], \qquad \text{(A.16)}
\end{aligned}$$

$$(\delta p)(\mathbf{x},T) = \beta_{T,1}(\delta u)(\mathbf{x},T), \quad (\delta q)(\mathbf{x},T) = \beta_{T,2}(\delta v)(\mathbf{x},T), \quad \text{on } \Omega, \qquad \text{(A.17)}$$

$$\frac{\partial(\delta p)}{\partial\nu} = \frac{\partial(\delta q)}{\partial\nu} = 0, \quad \text{on } \partial\Omega\times[0,T]. \qquad \text{(A.18)}$$

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

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

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

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

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] A. Badugu, C. Kraemer, P. Germann, D. Menshykau, and D. Iber, *Digit patterning during limb development as a result of the BMP-receptor interaction*, Sci. Rep., 991 (2012), pp. 1–13.

[2] W. Bangerth, R. Hartmann, and G. Kanschat, *deal.II—a general-purpose object-oriented finite element library*, ACM Trans. Math. Software, 33 (2007), pp. Art. 24, 27.

[3] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer, 14 (2005), pp. 1–137.

[4] M. Benzi, E. Haber, and L. Taralli, *A preconditioning technique for a class of PDE-constrained optimization problems*, Adv. Comput. Math., 35 (2011), pp. 149–173.

[5] V. Castets, E. Dulos, J. Boissonade, and P. De Kepper, *Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern*, Phys. Rev. Lett., 64 (1990), pp. 2953–2956.

[6] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Monographs on Numerical Analysis, The Clarendon Press Oxford University Press, New York, 1989.

[7] A. D. Economou, A. Ohazama, T. Porntaveetus, P. T. Sharpe, S. Kondo, M. A. Basson, A. Gritli-Linde, M. T. Coburne, and J. B. A. Green, *Periodic stripe formation by a Turing mechanism operating at growth zones in the mammalian palate*, Nat. Genet., 44 (2012), pp. 348–351.

[8] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.

[9] R. Fletcher, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis (Proc. 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975), Springer, Berlin, 1976, pp. 73–89. Lecture Notes in Math., Vol. 506.

[10] M. R. Garvie, P. K. Maini, and C. Trenchea, *An efficient and robust numerical algorithm for estimating parameters in Turing systems*, J. Comput. Phys., 229 (2010), pp. 7058–7071.

[11] M. R. Garvie and C. Trenchea, *Identification of space-time distributed parameters in the Gierer-Meinhardt reaction-diffusion system*, SIAM J. Appl. Math., 74–1 (2014), pp. 147–166.

[12] M.W. Gee, C.M. Siefert, J.J. Hu, R.S. Tuminaro, and M.G. Sala, *ML 5.0 smoothed aggregation user's guide*, Tech. Report SAND2006-2649, Sandia National Laboratories, 2006.

[13] A. Gierer and H. Meinhardt, *A theory of biological pattern formation*, Biol. Cybernet., 12 (1972), pp. 30–39.

[14] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, third ed., 1996.

[15] E. Haber, U. M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Probl., 16 (2000), pp. 1263–1280.

[16] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–436 (1953).

[17] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*, Mathematical Modelling: Theory and Applications, Springer-Verlag, New York, 2009.

[18] C. Hogea, C. Davatzikos, and G. Biros, *An image-driven parameter estimation problem for a reaction–diffusion glioma growth model with mass effects*, J. Math. Biol., 56 (2008), pp. 793–825.

[19] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, vol. 15 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[20] Y. A. Kuznetsov, *Efficient iterative solvers for elliptic finite element problems on nonmatching grids*, Russ. J. Numer. Anal. M., 10 (1995), pp. 187–211.

[21] R. T. Liu, S. S. Liaw, and P. K. Maini, *Two-stage Turing model for generating pigment patterns on the leopard and the jaguar*, Phys. Rev. E, 74 (2006), pp. 011914–1.

[22] M. F. Murphy, G. H. Golub, and A. J. Wathen, *A note on preconditioning for indefinite linear systems*, SIAM J. Sci. Comput, 21 (2000), pp. 1969–1972.

[23] J. D. Murray, *Mathematical biology. Vol. 2: Spatial models and biomedical applications. 3rd revised ed.*, New York, NY: Springer, 2003.

[24] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[25] ———, *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, second ed., 2006.

[26] Q. Ouyang and H. L. Swinney, *Transition from a uniform state to hexagonal and striped Turing patterns*, Nature, 352 (1991), pp. 610–612.

[27] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

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

[29] 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 (2012), pp. 1126–1152.

[30] J. W. Pearson and A. J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numer. Linear Algebra Appl., 19 (2012), pp. 816–829.

[31] ———, *Fast iterative solvers for convection-diffusion control problems*, Electron. Trans. Numer. Anal., 40 (2013), pp. 294–310.

[32] Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[33] J. Schnakenberg, *Simple chemical reaction systems with limit cycle behaviour*, J. Theoret. Biol., 81 (1979), pp. 389–400.

[34] R. Sheth, L. Marcon, M. F. Bastida, M. Junco, L. Quintana, R. Dahn, M. Kmita, J. Sharpe, and M. A. Ros, *Hox genes regulate digit patterning by controlling the wavelength of a Turing-type mechanism*, Science, 338 (2012), pp. 1476–1480.

[35] F. Tröltzsch, *Optimal control of partial differential equations: theory, methods and applications*, American Mathematical Society, 2010.

[36] A. Turing, *The chemical basis of morphogenesis*, Philos. Trans. R. Soc. London, Ser. B, 237 (1952), pp. 37–72.

[37] A. J. Wathen and T. Rees, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2008), pp. 125–135.