Numerical bifurcation analysis of a 3D turing-type reaction–diffusion model

Song, Weiyan; Wubs, Fred; Thies, Jonas; Baars, Sven

Published in:
Communications in nonlinear science and numerical simulation

DOI:
10.1016/j.cnsns.2018.01.003

Publication date:
2018

Commun Nonlinear Sci Numer Simulat 60 (2018) 145–164

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

journal homepage: www.elsevier.com/locate/cnsns

Research paper

Numerical bifurcation analysis of a 3D turing-type reaction–diffusion model

Weiyan Song[a,b,*], Fred Wubs[b], Jonas Thies[a], Sven Baars[b]

[a] Facility of Simulation and Software Technology, German Aerospace Center (DLR), 51147 Cologne, Germany
[b] Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, 9700AK Groningen, The Netherlands

ARTICLE INFO

Article history:
Received 29 September 2017
Revised 20 December 2017
Accepted 7 January 2018
Available online 4 February 2018

Keywords:
Continuation
Pattern formation
Bifurcation diagram
Stability

ABSTRACT

We perform a numerical study of a two-component reaction–diffusion model. By using numerical continuation methods, combined with state-of-the-art sparse linear and eigenvalue solvers, we systematically compute steady state solutions and analyze their stability and relations in both two and three space dimensions. The approach gives a more reliable and complete picture than previous efforts based on time integration schemes and is also typically much more efficient in terms of computing time. We are therefore able to produce a rich bifurcation diagram showing a variety of solution patterns and transitions.

1. Introduction

Among the rich numerical bifurcation analysis toolbox, continuation techniques are efficient numerical schemes to compute solution manifolds of nonlinear systems and determine attractors as a function of model parameters for discretized partial differential equations (PDEs). The use of continuation methods for solving nonlinear systems of equations can be traced back to the work of Lahaye in 1934, for a detailed review see [1,2] and references therein. In order to avoid singularity of the Jacobian, e.g. in turning points, an additional equation is added to normalize the tangent along the branch, this leads to the well-known pseudo-arclength continuation algorithm first published in [3]. Compared to time-marching schemes, continuation methods avoid the potentially long integration process to equilibrium, hence, obtain generic and meaningful information in much shorter computing time [4]. Moreover, by just doing time integration to find a steady state, in general, one can not find unstable steady states and in some cases it cannot even find certain stable solutions, see Section 3.5.

In each continuation step, a large nonlinear system of equations has to be solved. The most frequently used methods are Newton–Raphson (Newton's method for short) and Picard-iteration, which lead to a sparse linear systems that have to be solved in each iteration. Both methods need a sufficiently accurate initial guess to achieve convergence.

Apart from continuation of solutions, one often also wants to determine their (linear) stability under variation of parameters. This leads to the solution of an eigenvalue problem. A general method for computing all eigenvalues can be used for small and moderate sized problems. Based on linear algebra, it can be used to study the stability of a fixed point with respect to small perturbations [5]. For large and sparse systems, there are a variety of iterative methods for computing a few eigenpairs near a selected target [6]. We will employ the Jacobi–Davidson method [7,8].

* Corresponding author at: Facility of Simulation and Software Technology, German Aerospace Center (DLR), 51147 Cologne, Germany.
E-mail address: Weiyan.Song@DLR.de (W. Song).

https://doi.org/10.1016/j.cnsns.2018.01.003
1007-5704/

Table 1
Some parameter values that will be fixed throughout this paper.

D	α	β	δ	r_1	L
0.516	0.899	−0.91	2	3.5	30

We consider a widely studied model for spatial pattern formation, proposed by Turing [9]. Turing showed that a system of two reacting and diffusing chemicals could produce spatial patterns in chemical concentrations from the destabilization of a homogeneous state. Many experimental results have illustrated the formation of striped and spotted patterns, as well as more complicated patterns [10]. The term diffusion-driven instability has appeared in studies of chemical and ecological processes. Turing models can exhibit most of those patterns and they can be found in many theoretical and experimental papers. For an overview, see [11–13]. Bifurcation and stability studies of the trivial solution have also been conducted: Callahan and Knobloch gave the bifurcation diagram including the solution patterns of the Brusselator model and the Lengyel–Epstein model [14,15] near the trivial steady state in 3D. Liu et al. [16] studied the occurrence of Hopf bifurcations departing from the trivial state in 1D and found conditions for which these exist. They illustrate their theoretical analysis by showing limit cycles. These theoretical results do not show what happens if one is following a branch of non-trivial solutions.

Recently, also the study of Turing patterns on complex networks has emerged. Nakao and Mikhailov [17] studied Turing patterns on large random networks. The problem of Turing instabilities for a reaction–diffusion system defined on a complex Cartesian product network is considered in [18]. Generally, Turing problems have been addressed using the time-dependent method; numerical continuation is used rarely. McCullen and Wagenknecht in 2016 investigated patterns on complex networks computationally using numerical continuation methods for 2D networks [19]. We have not seen numerical continuation with multigrid technique applied for 3D analyses in this field.

It is known that 3D solutions can display much richer behavior than 2D solutions. There are much greater possibilities for spatial multistability in three dimensions than in two. These are not only interesting from a theoretical point of view, but are possible in nature as is shown, e.g. by Bánsági et al. [20]. They also expect that in the near future evidence will be found of “3D self-regulated patterns in living systems”. A decade ago, not many 3D results existed, and the results that existed were on relatively coarse grids, e.g. [21,22]. Recently, however, quite a few 3D results that were generated on parallel computers were published using time integrations methods, e.g. [23,24]. Note that time integration methods, using Euler's method, are very well suited for computations on GPUs, as is done in these papers. However, using Euler's method, the time step must be chosen smaller and smaller when the grid is refined.

The general form of a Turing system for modeling the evolution of the concentrations of two chemicals is as follows:

$$\begin{aligned} \frac{\partial U}{\partial t} &= D_U \nabla^2 U + f(U,V), \\ \frac{\partial V}{\partial t} &= D_V \nabla^2 V + g(U,V), \end{aligned} \tag{1}$$

where $U = U(x,t)$ and $V = V(x,t)$ are the two concentrations, and D_U and D_V are the respective diffusion coefficients. The scalar functions f and g represent the reactions between the components, which are usually nonlinear.

There are various Turing models with different reaction kinetics depending on the application, for instance the Brusselator model [25], the Gray–Scott model [26] and the Lengyel–Epstein model [27]. In this paper, we study the model introduced by Barrio et al. [28] in 1999. As a general Turing model, it has applications in imitating the pattern formation on various fish species' skin [29]. The equations are obtained by expressing (1) in terms of a perturbation with respect to the stationary uniform solution (U_c, V_c), and then solving $f(U_c, V_c) = g(U_c, V_c) = 0$. Neglecting terms of order higher than 3, the equations are given by

$$\begin{aligned} \frac{\partial u}{\partial t} &= D\delta \nabla^2 u + \alpha u(1 - r_1 v^2) + v(1 - r_2 u), \\ \frac{\partial v}{\partial t} &= \delta \nabla^2 v + v(\beta + \alpha r_1 u v) + u(\gamma + r_2 v), \end{aligned} \tag{2}$$

where $u = U - U_c$ and $v = V - V_c$, so the point $(u, v) = (0, 0)$ is the stationary solution. The constant δ is a scaling factor and D is the ratio between the diffusion coefficients of the two chemicals. We note that D must not be equal to one in order to make the diffusion-driven instability occur. There are two parameters r_1 and r_2 in the nonlinear interactions, affecting a cubic and quadratic term, respectively. In [28], it is observed that the cubic term favors stripe patterns and the quadratic ones spot patterns. In our experiments, we use periodic boundary conditions, and to make the investigation as simple as possible, we set $\alpha = -\gamma$, so that (0, 0) is the only spatially uniform steady solution. In this paper we study only one set of parameters as indicated in Table 1. This set is one of the choices made in [28]. In the following, we will determine branches of steady solutions of the equations above as a function of r_2 on, respectively, an interval, a square and a cube with edge length $L = 30$.

In Section 2, we analyze the model theoretically in terms of the linear stability of the trivial solution and determine unstable modes. These modes will later be used to get onto non-trivial solution branches. Furthermore, we discuss some

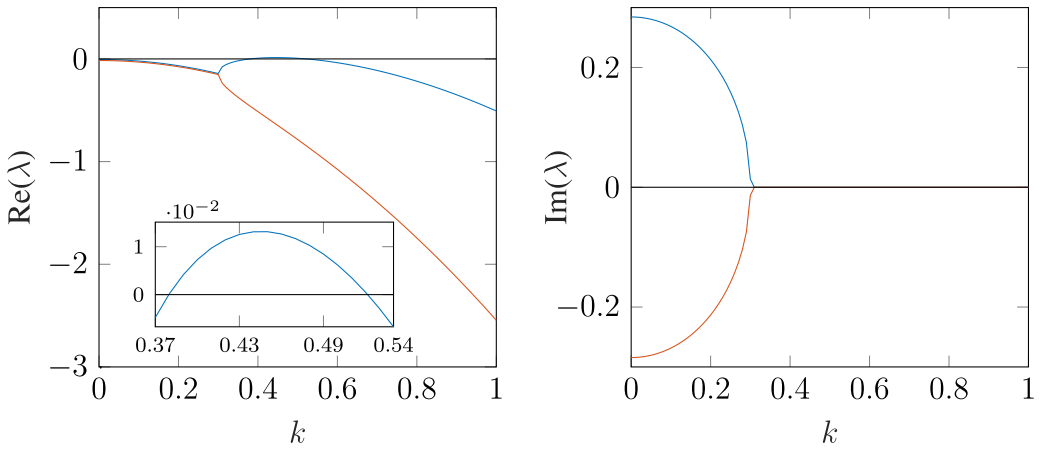

Fig. 1. Two eigenvalues of the linearized Eqs. (4) with respect to the wave number k.

properties of the non-trivial solutions, such as degrees of freedom, symmetry and similarity of the solutions. The main numerical methods we use are introduced in Section 3: the continuation methodology and the Jacobi–Davidson method for computing eigenpairs for the linear stability analysis. Both techniques require the solution of large and sparse linear systems. To solve these systems iteratively, a multigrid preconditioned Krylov solver is used.

In Section 4, we present numerical results in 1D, 2D and 3D, including bifurcation diagrams, stability of various solution patterns and an overview of the performance of the algorithms and the parallel implementation.

2. Model analysis

In the following part, a mathematical analysis will be carried out in order to get a better idea of the characteristics of the solutions of Eq. (2).

2.1. Linear stability analysis of the trivial solution

We follow the exposition in [28]. We start with noting that the nonlinear part of the right-hand side of (2) will not contribute to its Jacobian. Now, in the absence of diffusion, standard linear analysis predicts exponentially growing solutions of the form $(u, v) = (u_0 \exp(\lambda t), v_0 \exp(\lambda t))$ where λ is an eigenvalue, with

$$\lambda = \frac{1}{2}[(\alpha + \beta) \pm \sqrt{(\alpha + \beta)^2 - 4\alpha\beta - \gamma}]. \tag{3}$$

In the presence of diffusion, the spatial variation of the functions u and v is of the form $\exp(i\mathbf{k} \cdot \mathbf{x})$ where both $\mathbf{k}$ and $\mathbf{x}$ are vectors in R^d, d being the number of dimensions in which the problem (2) is considered, and the dispersion relation of the linearized equations is given by

$$\lambda^2 + B\lambda + C = 0, \tag{4}$$

with $B = -k^2\delta(1 + D) + \alpha + \beta$ and $C = (\alpha - \delta D k^2)(\beta - \delta k^2) + \alpha$, $k^2 = \mathbf{k} \cdot \mathbf{k}$.

Fig. 1 shows the real and imaginary parts of the two eigenvalues of Eq. (4) as a function of the wave number k with the parameter values given in Table 1. For $k \leq 0.34$, there is a complex pair of conjugate eigenvalues, the real part of which is less than zero, hence the trivial solution is stable there. For $k \approx 0.34$ these two eigenvalues turn into two real eigenvalues, one of which is positive in the range

$$[0.39, 0.52]. \tag{5}$$

So if the size of the d-dimensional domain allows wave numbers k in this interval then the trivial solution will be unstable, otherwise it will be stable. Next, we will study whether they can occur on generalized squares of size $L = 30$ in 1, 2 and 3 dimensions, respectively.

Unstable modes in 1D, 2D and 3D. For the parameters in Table 1, and a specific size of the domain, say $L = 30$, we can now specify in the various dimensions which modes become unstable. First we define the constant $\kappa = 2\pi/L \approx 0.2094$, which will be used through out the paper. In 1D the wave numbers are given by $k = \kappa m$, with $m = 1, 2, \ldots$. For $m = 2$ we find the sole wave number in the interval (5): $k = 0.4189$. In 2D, the wave numbers are given by $k = \kappa\sqrt{m^2 + n^2}$, with $m, n = 0, 1, 2, \ldots$ (and m and n not both zero) and in 3D similarly $k = \kappa\sqrt{m^2 + n^2 + l^2}$. Note that this relates to wave vectors

Table 2
Stable and unstable modes with respect to the trivial solution. Unstable modes occur in the interval (5).

Type	*l, m, n*	*k*	Name
1D, 2D, 3D			
Stable	1,0,0	0.2094	
Unstable	2,0,0	0.4189	S2
Stable	3,0,0	0.6283	
2D,3D			
Stable	1,1,0	0.2962	
Unstable	2,1,0	0.4683	S3,S4
Stable	2,2,0	0.5924	S1
3D			
Stable	1,1,1	0.3628	S6
Unstable	2,1,1	0.5130	S7,S8,S9
Stable	2,2,0	0.5924	
Stable	2,2,1	0.6283	

$\mathbf{k} = \kappa[m; n]$ and $\mathbf{k} = \kappa[m; n; l]$ in 2 and 3 dimensions, respectively. In Table 2, we list the wave numbers of the modes making the zero solution unstable together with those of surrounding stable modes.

Due to the fact that we are working on squares and cubes, the values may be randomly permuted over *m, n* and *l*. We also gave names to the unstable modes, because we will use these later to name the branches of solutions that they generate. Observe that the (2,1,0) mode has two names. This is due to the fact that for a square there is a rotated 1D mode and a genuine 2D mode, which have the same wave number. On one hand we have $\cos((2x+y)\kappa)$, which we will call S3, and on the other hand $\cos(2x\kappa)\cos(y\kappa)$, which we call S4. Note that due to the periodic boundary conditions all cosines may have a different phase shift. Note also that $\cos((2x+y)\kappa) = \cos(2x\kappa)\cos(y\kappa) - \sin(2x\kappa)\sin(y\kappa)$, which relates the two modes. The product of sines here is just a phase shifted version of the product of cosines. Moreover, in the right-hand side we could differentiate the kappas for the *x* and *y* direction to adjust to a non square domain. Since we cannot do this in the left-hand side, the rotated mode will not appear on a non-square domain. However, due to the combination in the right-hand side we are able to see something almost of that shape if the lengths of the sides differ not much from each other.

Similar phenomena occur in 3D. Observe that the mode (2,1,1) has three names: S7, S8 and S9, which are a truly 3D droplet pattern started with initial solution $\cos(2\kappa x)\cos(\kappa y)\cos(\kappa z)$, a tilted lamellae pattern started with $\cos(2\kappa x + \kappa y + \kappa z)$ and a tilted cylinder started with $\cos(2\kappa x)\cos(\kappa y + \kappa z)$, respectively. These are also related to goniometric identities, e.g.$\cos(2\kappa x + \kappa y + \kappa z) = \cos(2\kappa x)\cos(\kappa y + \kappa z) - \sin(2\kappa x)\sin(\kappa y + \kappa z)$.

We remark that the choice of *L* also determines how many unstable modes will occur. For instance, if *L* is doubled (as is done in [21,28]), the modes in the table reoccur but with wave number twice as big in all directions. So *l, m, n* in Table 2 become even numbers, which opens up the possibility that a combination with an odd number will drop into the interval (5). A simple computation reveals that this is indeed the case.

2.2. Magnitude estimate of non-trivial solution

A mode that makes the trivial solution unstable may turn into a stable solution of the nonlinear equations. By performing a Galerkin projection of the problem, an estimate of the magnitude of such a solution can be obtained. We show this process for the 1D case, but equally well it can be done in the 2D and 3D case.

We write the steady state system related to (2) as the vector equation $F(\mathbf{u}) = J\mathbf{u} + N(\mathbf{u}) = 0$, where *J* is the Jacobian matrix of *F* and *N* is the remaining nonlinear part. Here, the Jacobian is the same matrix for which the eigenvalues have been determined. Let $\mathbf{v}$ denote the eigenvector associated with the positive eigenvalue; note that eigenvalue and eigenvector are both real. Next we express $\mathbf{u}$ in this unstable mode and call this specific choice $\mathbf{u}_k$, so

$$\mathbf{u}_k \equiv \epsilon \mathbf{v} \sin(kx)$$

with $||\mathbf{v}|| = 1$. By the Galerkin approach, it is required that

$$\int_{x=0}^{L_x} \sin(kx)\mathbf{v}^T F(\mathbf{u}_k)dx = 0,$$

which results in

$$\int_{x=0}^{L_x} \sin(kx)\mathbf{v}^T[\lambda \mathbf{v}\sin(kx) + N(\mathbf{u}_k)/\epsilon)]dx = 0.$$

Suppose $N(\mathbf{u}) = Q(\mathbf{u}) + C(\mathbf{u})$, where *Q* and *C* are quadratic and cubic in $\mathbf{u}$, respectively. Then we can write

$$N(\mathbf{u}_k)/\epsilon = \epsilon \sin^2(kx)Q(\mathbf{v}) + \epsilon^2 \sin^3(kx)C(\mathbf{v}).$$

Suppose integrals over the domain of $\sin^2$, $\sin^3$, $\sin^4$ are, respectively, $a, 0, b$ then we end up with an equation of the form

$$\lambda a + \epsilon^2 b \mathbf{v}^T C(\mathbf{v}) = 0,$$

with

$$\epsilon = \sqrt{-\lambda a/(b\mathbf{v}^T C(\mathbf{v}))}.$$

In our case $C(\mathbf{u}) = \alpha r_1 u_1 u_2^2[-1; 1]$ and $\mathbf{v}^T C(\mathbf{v}) = \alpha r_1 v_1 v_2^2(v_2 - v_1)$, where $\mathbf{u} = [u_1; u_2]$ and similar for $\mathbf{v}$. To get a real solution, ϵ should be real. This is indeed the case for $k = 0.4189$ with the parameters specified in Table 1. For these values $\mathbf{v} = [0.8167, -0.5771]$ and $\epsilon = 0.1122$.

The knowledge gained in this section helps us to start the continuation and we will come back to this later.

2.3. Some properties of the solutions

Nonuniqueness of the solutions. Due to the periodic boundary conditions one can shift the solutions around in the plane. In principle, one has to apply a (phase) condition to prohibit this. This also results in zero eigenvalues of the Jacobian at the solutions. The number of zero eigenvalues depends on the solution. A 1D solution can only be shifted in one direction so it has a single eigenvalue zero. This also holds for generalizations of 1D solutions to 2D and 3D. A genuine 2D solution, i.e. one which cannot be found from a generalization of a 1D solution, can be shifted in two directions, each shift giving another solution, so here we have two zero eigenvalues. Similarly, in three dimensions there are three zero eigenvalues for a genuine 3D solution.

Symmetries and coinciding eigenvalues. Apart from multiple zero eigenvalues one also finds equal non-zero eigenvalues. For instance, if in 2D the solution corresponding to the (2,0) mode is studied, we will find an unstable mode of genuine two dimensional shape. This mode may be shifted arbitrarily in the y-direction and gives rise to an independent eigenvector. Hence, this builds a two-dimensional subspace of unstable modes and consequently leads to a double positive eigenvalue.

Similarly, the 3D mode (2,1,1) has in itself already three representations, as we have seen above. Moreover, the 2 can be at three positions, leading to at least 9 equal eigenvalues. In a numerical computation we found even up to 24 equal eigenvalues. We will get back to that in Section 4.3 where we study some of the 3D modes.

Symmetry in solutions is a much studied subject, see for instance [30–32]. It is possible to solve the problem on a small portion of the domain with various boundary conditions, whereupon various combinations of these solutions gives the whole range of solutions. We did not exploit this in this paper.

Similarity solutions. In Table 1, we fixed $r_1 = 3.5$. We will show now that if we compute the solutions for all r_2, then we do so for any pair (r_1, r_2).

Suppose at certain parameter values $(\hat{r}_1, \hat{r}_2)$ we have a steady state $(\hat{u}, \hat{v})$. Now we wonder for which values (r_1, r_2), $(\mu\hat{u}, \mu\hat{v})$ is a solution. If we substitute this into Eq. (2) we have that

$$\begin{aligned} 0 &= \mu D\delta\nabla^2\hat{u} + \alpha\mu\hat{u}(1 - r_1\mu^2\hat{v}^2) + \mu\hat{v}(1 - r_2\mu\hat{u}), \\ 0 &= \mu\delta\nabla^2\hat{v} + \mu\hat{v}(\beta + \alpha r_1\mu^2\hat{u}\hat{v}) + \mu\hat{u}(\gamma + r_2\mu\hat{v}). \end{aligned} \tag{6}$$

After dividing by μ we find

$$\begin{aligned} 0 &= D\delta\nabla^2\hat{u} + \alpha\hat{u}(1 - r_1\mu^2\hat{v}^2) + \hat{v}(1 - r_2\mu\hat{u}), \\ 0 &= \delta\nabla^2\hat{v} + \hat{v}(\beta + \alpha r_1\mu^2\hat{u}\hat{v}) + \hat{u}(\gamma + r_2\mu\hat{v}). \end{aligned} \tag{7}$$

So $(\mu\hat{u}, \mu\hat{v})$ is a solution if both $\mu^2 r_1 = \hat{r}_1$ and $\mu r_2 = \hat{r}_2$. The family of solutions defined by $\hat{r}_1$, $\hat{r}_2$ and μ has the same stability behavior, with exactly the same eigenvalues.

3. Numerical methods

We use a standard second order central finite difference scheme (3-point for 1D, 5-point for 2D and 7-point for 3D) to discretize (2) in space. An equidistant grid is used in all of our experiments. Rather than discretizing the time dimension as well, we focus on the direct computation of steady states using pseudo-arclength continuation [3]. An implementation of the algorithm is available in the Trilinos [33] library LOCA ("Library of Continuation Algorithms"). The arising linear systems are solved using the well-known GMRES method (Trilinos package Belos) with an algebraic multigrid preconditioner (ML package). Eigenvalues and eigenvectors are computed using PHIST, a recent implementation of the Jacobi–Davidson method [34] that allows for non-symmetric matrices and easy integration with Trilinos applications. Below, the methods are briefly outlined in order to make the paper more accessible.

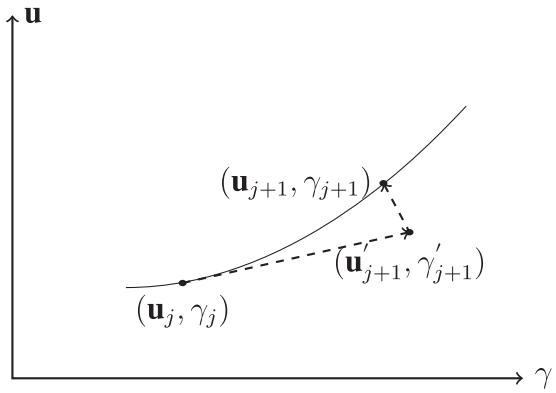

Fig. 2. Sketch of the continuation method: (i) prediction of the next solution in the direction of the tangent, (ii) correction by Newton's method.

3.1. Continuation approach

Our research is focused on the steady state of system (2). By continuation in a specific parameter, a series of approximate solutions can be generated by solving a system of parameterized nonlinear equations, $F(\mathbf{u}, \gamma) = 0$, where $\mathbf{u} = (u, v)$ is the solution and γ is the model parameter varying during the continuation. A predictor–corrector scheme is used, as illustrated geometrically in Fig. 2, where $(\mathbf{u}'_{j+1}, \gamma'_{j+1})$ is an approximation close to the branch obtained by the predictor step (Euler step or extrapolation) and $(\mathbf{u}_{j+1}, \gamma_{j+1})$ is the solution of the corrector step (Newton's method here) [35]. In order to avoid singularity of the Jacobian, e.g. in turning points, instead of increasing in the parameter we increasing in an approximation of the arclength by requiring that the increase in tangential direction along the branch should be equal to a user set step size. This leads to the well-known pseudo-arclength continuation algorithm first published in [3]. Compared to time integration methods, continuation is more efficient, especially when computing the whole bifurcation diagram or when slowly decaying modes lead to very slow approaches to equilibrium of the time dependent simulation. Additionally, it allows us to investigate the stability of various branches of solutions and capture also the unstable solutions.

3.2. Solution of linear systems

The linearized systems that arise in the continuation process take the form of two independent Laplace operators on the diagonal with off-diagonals for the coupling of the unknowns u and v. The coupling makes the matrix non-symmetric, so the solver of choice is the GMRES Krylov subspace method. We observe that the coupling terms are constant, independent of the grid size $\Delta x = x_{i+1} - x_i$. Therefore, at sufficiently high resolution, the Laplace terms dominate the convergence of the iterative solver. Multigrid preconditioning is an obvious choice to keep the number of linear solver iterations at bay. Being readily available in Trilinos, we choose the smoothed aggregation AMG solver ML, see [36] for documentation. Although the systems solved are all singular as discussed in Section 2.3, the multigrid method will converge to a particular solution in the solution space. This often occurs with iterative procedures, as long as matrices inverted in those procedures are non-singular. We have experienced that a direct method does indeed fail.

3.3. Linear stability analysis by eigenvalue computation

After computing the steady solution, we can analyze its stability by computing the right-most eigenvalues of the Jacobian matrix at this state. While many methods exist for solving this task, two characteristics of our algorithm guide our choice in particular: the availability of a preconditioner for the matrix in question, and the fact that a sequence of eigenvalue problems for a varying parameter is being solved.

The Jacobi–Davidson QR (JDQR) method [7,8] can be started with an approximate subspace (not just one vector) and requires the approximate solution of a linear system, for which in turn the existing solver/preconditioner combination can be used. We use the publicly available PHIST implementation [34] of the block Jacobi–Davidson QR method, which can straightforwardly be used in Trilinos applications. The basic implementation described in [37] was extended to allow using a (left) preconditioner as follows. The search space is extended in each outer iteration by n_b corrections obtained as the solutions of the independent left-preconditioned linear systems

$$P_{\tilde{Q}}^{-1}(A - \tilde{\lambda}_j I)\Delta q_j = -P_{\tilde{Q}}^{-1}(Aq_j - \tilde{\lambda}_j q_j), \tag{8}$$

where $\tilde{\lambda}_j, q_j, j = 1, \ldots, n_b$ are the current approximations to the next few eigenvalues and eigenvectors to converge, respectively, and $\tilde{Q} = [Q, q_1 \ldots q_{n_b}]$ also contains the already converged eigenspace. The right-hand side represents the preconditioned eigenvalue residual. The preconditioner $P_{\tilde{Q}}^{-1}$ is chosen such that it produces vectors in the orthogonal complement of

$\tilde{Q}$, i.e. if P^{-1} approximates the action of A^{-1},

$$P_{\tilde{Q}}^{-1} = (I - \tilde{Q}_P(\tilde{Q}^T\tilde{Q}_P)^{-1}\tilde{Q}^T)P^{-1}, \tag{9}$$

$$\tilde{Q}_P = P^{-1}\tilde{Q}. \tag{10}$$

We found that using an AMG cycle on the linear part of Eq. (2) for the action of P^{-1} yields good convergence behavior of the overall method. One argument for this choice is that the sought eigenvalues are close to 0, so that neglecting the shifts $\tilde{\lambda}_j$ in the preconditioner is a reasonable approximation. Furthermore, choosing a symmetric preconditioning operator precludes introducing non-normality in the operator, which may otherwise impair the quality of the updates Δq_j computed by the inner GMRES method.

3.4. Branch switching

Leaving the trivial branch. The general way to get on the branch of non-trivial solutions is to solve the eigenvalue problem for the Jacobian matrix of the discretized problem and select those modes that have a positive eigenvalue. With each of these one can create a Galerkin projection as indicated in Section 2.2 and find its approximate magnitude. At the end of that section a specific solution is given. Based on this our starting solution will have the form

$$0.1 \cdot \phi(x, y, z)[1; -1], \tag{11}$$

where ϕ is a combination of goniometric functions related to the unstable modes in Table 2. This guess appeared to be good enough to get convergence to the associated mode of the nonlinear system.

Switching between non-trivial branches. Once we are on a non-trivial branch we might also need to get onto another branch. The direction where to go is given by the eigenvector associated with the unstable eigenvalue. If u is the steady state solution on the current branch near the bifurcation point, we use as an initial guess $\mathbf{u} + \epsilon\mathbf{v}$ for some small values of ϵ for the next Newton iteration, which will then typically converge to a nearby solution on the new branch.

3.5. Differences with time integration approach

In the continuation approach, the time derivative disappears, but is implicitly still there, as we will explain here. In order to study the stability we want to find solutions of the form $\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \epsilon(t)\mathbf{v}(\mathbf{x})$, where $\bar{\mathbf{u}}(\mathbf{x})$ is a steady state solution. Inserting this in the equation $d\mathbf{u}/dt = F(\mathbf{u})$, we obtain for small $\epsilon(t)$ the equation

$$\frac{d\epsilon}{dt}\mathbf{v} = \epsilon J(\bar{\mathbf{u}})\mathbf{v}.$$

This linear equation allows for separation of variables, leading to

$$\frac{d\epsilon}{dt} = \lambda\epsilon, \quad \lambda\mathbf{v} = J(\bar{\mathbf{u}})\mathbf{v}.$$

The former has the solution $\epsilon(t) = \epsilon(0)\exp(\lambda t)$, and the latter has eigenvalues and eigenvectors as its solution. Once the eigenproblem is solved we find solutions of the specified form $\mathbf{u}(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}) + \epsilon(0)\exp(\lambda t)\mathbf{v}(\mathbf{x})$.

For a time integrator one can perform the analogous process. For instance, for the forward Euler method one finds a different equation for $\epsilon(t)$

$$\epsilon(t + \Delta t) = \epsilon(t) + \lambda\Delta t\epsilon(t) = (1 + \lambda\Delta t)\epsilon(t).$$

In this case, the solution is $\mathbf{u}(\mathbf{x}, n\Delta t) = \bar{\mathbf{u}}(\mathbf{x}) + \epsilon(0)(1 + \lambda\Delta t)^n\mathbf{v}(\mathbf{x})$. Note that $(1 + \lambda\Delta t)$ is a crude approximation of the exponential $\exp(\lambda\Delta t)$; it is called the amplification factor below.

So in reality, the perturbation damps out (amplifies) if $\mathcal{R}(\lambda) < 0$ ($\mathcal{R}(\lambda) > 0$), but for the Euler integrated version this occurs if $|1 + \lambda\Delta t| < 1$ ($|1 + \lambda\Delta t| > 1$), which is quite different.

Now, let us return to the original problem, where one gives an initial guess and performs a stable integration with the forward Euler method. Then by varying the initial conditions one may find various steady states. However, the consequence of the above is that only steady states can be found for which $|1 + \lambda\Delta t| < 1$, for all eigenvalues λ that occur. From this, it is immediately clear that an unstable steady state (at least one positive eigenvalue) cannot be found. Moreover, solutions which are stable but with an eigenvalue containing a relative large imaginary component, i.e. near a Hopf bifurcation, cannot be found if for that eigenvalue $|1 + \lambda\Delta t| > 1$. So even if we would start with such a steady solution we would be repulsed from it. This is a peculiarity one should be aware of when using the forward Euler method.

Another popular scheme is the backward Euler method. In that method, the amplification is $1/|1 - \lambda\Delta t|$, and one can take arbitrarily large time steps. However, one may find some stationary solutions even if they are unstable, wrongly concluding that they are stable since the numerical method converged.

As will be clear by now every time integration method has its own peculiarities, which can influence our conclusions with respect to stability. This cannot happen with the continuation approach, which makes it mathematically superior here.

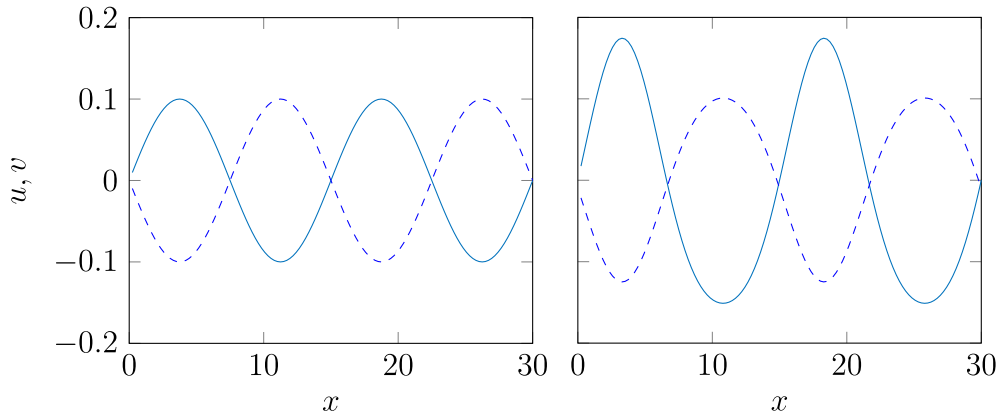

Fig. 3. Non-trivial 1D solution components u (solid) and v (dashed), left for $r_2 = 0$ and right for $r_2 = 1$.

4. Numerical experiments

In this section, we will show results for the 1D, 2D and 3D case, respectively. In all cases, we use a grid with 128 nodes in each direction. A justification for this is given in the paragraph "Refinement tests" at the end of Section 4.2.

4.1. Non-trivial solution in 1D

Table 2 shows that we have one stable nontrivial solution S2. If we choose $\phi(x) = \sin(2\kappa x)$ in (11) we indeed find it as a solution. In Fig. 3, this solution is shown for $r_2 = 0$ and $r_2 = 1$. Observe that u gets narrower tops and wider valleys where for v this is vice versa. Below we will only consider the interval [0,1], but here we remark that this solution is stable until about 1.58. At that point we find a Hopf-bifurcation leading to a stable periodic solution, with period $2\pi/0.1772$. This solution is stable for all values of r_2 between 0 and 1. It will be clear that this will also be a solution of the 2D and 3D case. However, we may not conclude that it is stable in the 2D and 3D case. But if it is unstable in the 2D case, then it must be due to a field that is genuinely 2D.

4.2. Bifurcation diagram of the 2D case

In our experiment, except for the uniform zero solution, five non-zero branches S1–S5 are found with appropriate initial vectors based on the model analysis. There are both stable and unstable stripe and spot patterns, see the bifurcation diagram in Fig. 4. The monitor we use here and in the following is defined by

$$M = \max_{x,y} \Delta u, \tag{12}$$

where Δ is the Laplacian. Being sensitive to curvature, M gives an insightful bifurcation diagram, but other choices are also possible.

The S2/S4 solution range. Below we start with a mode of the form S2 and will discover that also a mode of the form S4 is playing an important role. We show that the nonlinear behavior is occurring in a space built up by deformations of these two modes.

S2 Taking $\phi(x, y) = \sin(2\kappa x)$ gives the generalization of the 1D case, see Fig. 12. In 2D it is unstable, as expected, due to a mode with a genuinely 2D pattern resembling the S4 mode, see Fig. 13. Since this mode makes S2 unstable, we guess that a combination of the two becomes the stable one.

S5 By starting with the sum of the solution of $r_2 = 0.0$ on S2 and its eigenvector (an S4 mode), corresponding to the positive eigenvalue which makes the stripe pattern S2 unstable (see Fig. 13), the system converges to a solution on a stable branch which we will call S5, see Fig. 14. At $r_2 = 0.0$ it is a wavy striped pattern. Increasing r_2 to 0.21227, the spot (S4 mode) is getting more pronounced. At this value of r_2,there is a turning point which shows up by a new zero eigenvalue. See the bifurcation diagram for the 2D case (Fig. 4). When r_2 is decreasing after the turning point, the spot pattern (the S4 mode) is becoming even more pronounced in the solution; see Fig. 15. The continuation parameter r_2 goes back until 0.129, where the branch connects to the S4 branch, which we will study next.

S4 When choosing $\phi(x, y) = \cos(2\kappa x)\cos(\kappa y)$ in (11), the S4 mode with hexagonal spots is excited and Newton converges to a non trivial solution with the same pattern. It is unstable up to the bifurcation point $r_2 = 0.129$, where S5 splits off, and after that it is stable; see Fig. 16. Observe that the second plot, i.e. the one for $r_2 = 0.1$ is similar to the last of Fig. 15, the solution of S5 near the bifurcation point. Also note the symmetry that is growing in the y direction when

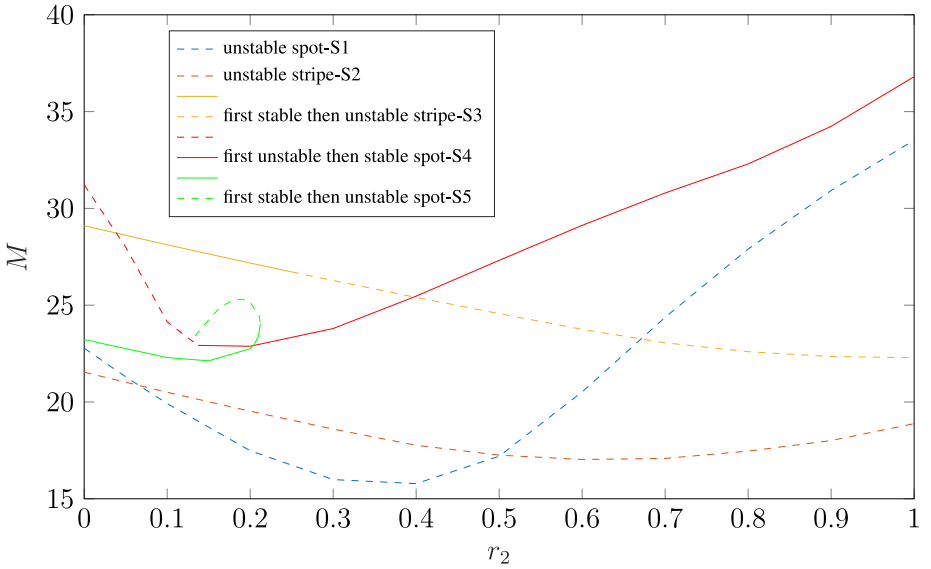

Fig. 4. Bifurcation diagram in 2D with parameter r_2 varying from 0.0 to 1.0. Stable and unstable branches are depicted by solid and dashed lines, respectively.

r_2 is decreasing (Fig. 15). It is clear that a sister of S5 would be the one where the yellow band is just going along the other sides of the spots. This will be the other solution emanating at the bifurcation point. Hence we have a pitchfork bifurcation here, and because the symmetry breaks, it is also a symmetry breaking bifurcation.

Summarizing, we see that the modes S2 and S4 and their combination give a partial solution to the problem. At $r_2 = 0$, a combination (S5) is stable until 0.212. When in a time-dependent simulation we increase r_2 from there, one will always converge to the stable S4 branch. The interesting thing is that S4 is also stable before 0.212, starting at 0.129, so there are two stable solutions (S4 and S5) on the interval [0.129, 0.212]. Before 0.129, S4 is unstable and hence S5 is the only stable solution there. We also understood why S4 changes stability at 0.129 and why S5 ceases to be stable at 0.212: S5 has a turning point at 0.212 and returns to S4 at 0.129, being unstable along that branch. For large r_2 only S4 is stable which also agrees with the observation in [28] that the quadratic term favors spots.

The S3 solution area. The S3 solution is a tilted version of the S2 solution. One might expect a similar behavior but that is actually not the case as we will see below

S3 When taking $\phi = \cos(\kappa(x + 2y))$ we find again a stripe solution which is oriented in the diagonal direction (Fig. 17). Note that the stripes can be tilted in any direction, depending on the initial solution. We observe that at the beginning all eigenvalues are negative, see Fig. 5. Around $r_2 = 0.25$, a pair of positive eigenvalues emerges having the same real value. The corresponding eigenvectors have a spot pattern and make the stripe solution unstable, see Fig. 18. Observe that the eigenvectors are the same up to a shift along the stripe direction. The bifurcation occurring at the double zero eigenvalue is difficult to address with our approach and we do not present results for the branches emerging at this point. In view of the eigenvectors shown in Fig. 18, we expect a stable mode splitting off which is non-unique since it can be shifted in stripe direction. We tried to get on the branch but the solution always jumped to S4, which is probably due to the singularity of the Jacobian.

Summarizing, the S3 solution is stable only for values of r_2 up to 0.25. This means that together with the previous solutions, there are three stable solutions on the interval [0.129, 0.212]. The eigenvectors appearing at 0.25 resemble a tilted S4, but we were not able to find a solution when starting with a tilted field resembling the eigenvector pattern: $\cos(\kappa(x/2 + y))\cos(\kappa(y - 3x/2)) = \cos(\kappa(x - 2y)) - \cos(2\kappa x)$. This field has the right periodicity conditions but is not an eigenmode of the linear part of the equation, since the two cosines in the last expression have different eigenvalues ($5\kappa^2$ and $4\kappa^2$ for the Laplace operator, respectively).

Refinement test. demonstrates the expected second-order accuracy of the discretization under grid refinement when following branch S4. The solution on the 256^2 grid is already very close that on the 128^2 grid. We therefore use 128 grid points in each direction for our experiments. To get an impression of the accuracy of the position of the bifurcation points, we perform a sensitivity analysis for the eigenvalues of S4 w.r.t. the number of grid points. We want to locate the bifurcation parameter r_2 at which the solution becomes stable. Close to the point where the positive eigenvalue crosses the imaginary axis, we compute the eigenvalues for different grids. In Table 3 shows the right-most non-zero eigenvalue for various r_2. In

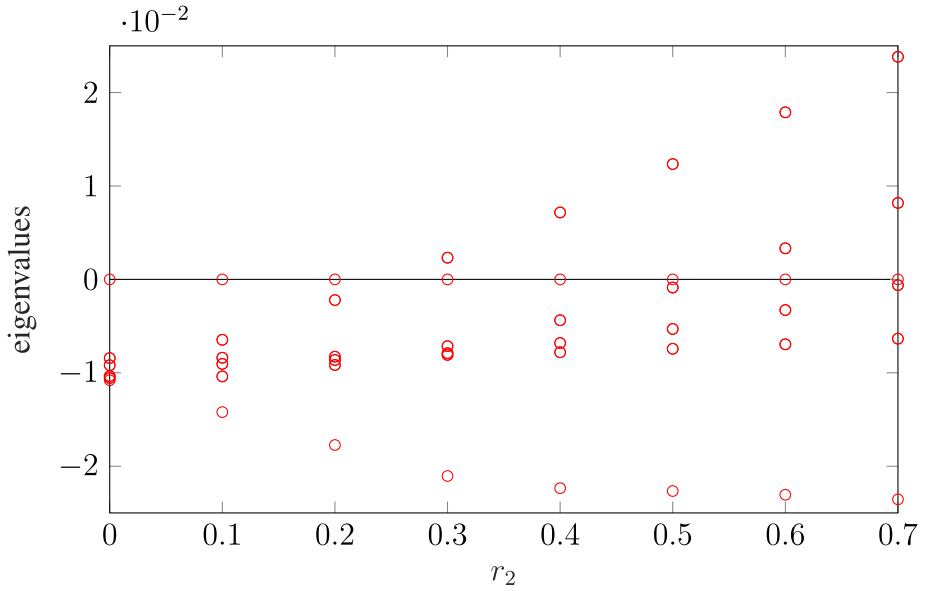

Fig. 5. Evolution of the right-most real eigenvalues as a function of r_2 on branch S3.

Table 3
The positive eigenvalue as a function of the grid size.

Grid	$r_2 = 0.14$	$r_2 = 0.145$	$r_2 = 0.15$	Critical r_2
32^2	1.308e−03	1.010e−03	7.146e−04	0.1621
64^2	2.476e−04	−4.953e−05	−3.409e−04	0.1442
128^2	−1.040e−04	−3.955e−04	−6.479e−04	0.1384
256^2	−2.010e−04	−4.900e−04	−7.30e−04	0.1369

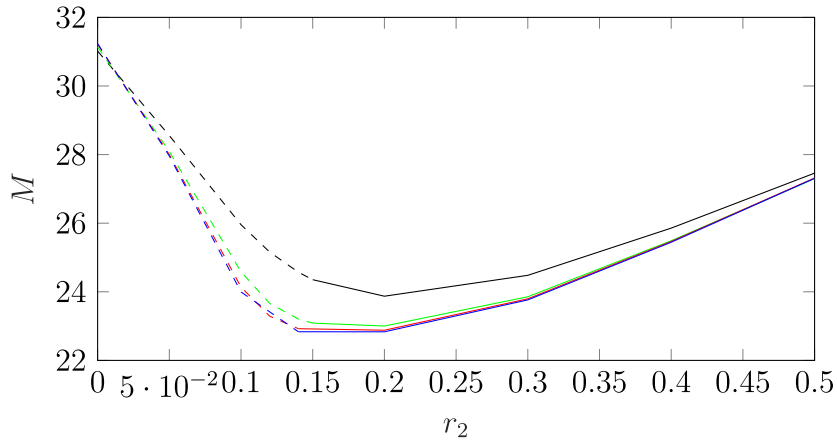

Fig. 6. Bifurcation diagram obtained on a 32^2 (black), 64^2 (green), 128^2 (red) and grid 256^2 (blue) grid on branch S4. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the columns, one observes the second-order convergence with the increasing mesh size. By quadratic interpolation in each row an estimate of the critical r_2 can be computed, which is presented in the last column. Fig. 6

Comparison to results from literature. Barrio et al. [28] and [38] also report 2D numerical results of Eq. (2). They started with random initial solutions and, after hundreds of thousands time steps using the Euler method, spot, stripe and hexagonal spot patterns are found for different parameter values r_1 and r_2. They observed that the cubic term favors stripe patterns while the quadratic term favors spot patterns. However, the cubic term (scaled by (r_1) does not completely suppress the occurrence of spot patterns. We also found the unstable spot patterns when $r_1 = 3.5$ and $r_2 = 0.0$, which is impossible by a time integration approach. Our results also show that when r_2 increases, the spot pattern is the only stable solution

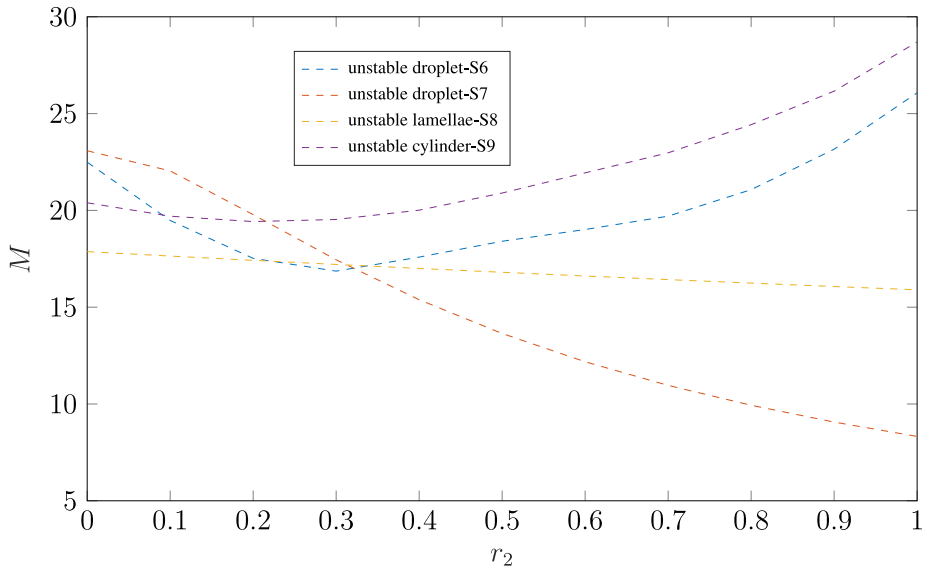

Fig. 7. Bifurcation diagram in 3D with parameter r_2 varying from 0.0 to 1.0, obtained on a 128^3 grid. For branches S1–S5, we refer to Fig. 4.

occurring and therefore it will be very robust in a time integration approach. Here, we have seen that there are regions in which 2 or even 3 stable solutions exist for the same parameter value.

4.3. Results for 3D

There are more morphologies in three dimensions, since besides the generalizations of the 2D solutions there will be new solutions. The 3D bifurcation diagram of branches S1–S5, including their stability properties, appeared to be exactly the same with those of 2D case. Therefore, in the bifurcation diagram for the 3D case we only plot the branches of genuine 3D solutions, i.e., in Fig. 7 we only plot the branches S6–S9, which are all unstable in the computed range. Next we discuss the classes of solutions separately.

Generalized 1D mode. We start off with the mode occurring already in the 1D case: S2. As in 2D, taking $\phi(x, y) = \sin(2\kappa x)$ again, we obtain the generalization of the 2D stripe pattern, which is also unstable, as expected. Since all constants have remained the same, the eigenvalues are also the same as those in 2D. But there are two additional independent eigenvectors appearing, i.e. four in total corresponding to the positive eigenvalue. In fact we find two pairs of eigenspaces, where each pair is an exact generalization of the 2D case: the S4 mode extended cylindrically. The pairs are just rotated 90° with respect to each other. Since these modes make S2 unstable, a combination of the two becomes the stable one, which is S5 explained below.

Generalized 2D modes. Next we considered the modes which were really of 2D shape. It appears that all modes S3, S4 and S5, have a similar behavior as in the 2D case. It seems that for all these cases the 3D-generalized 2D eigenvalues dominate over new eigenvalues corresponding to modes with truly 3D patterns. An example of that is shown in Fig. 8 for S4. The eigenvalues from the 2D case are indicated by a '$\circ$' while the new ones entering in the 3D case are indicated by a '+'.

The observation that for increasing parameter r_2 the spot pattern prevails does not seem to generalize to the 3D droplet solution. Instead the generalized S4 mode, which has a cylindrical pattern, prevails over a genuine 3D droplet pattern, most likely because the latter is not an unstable mode of the zero solution.

Genuine 3D modes. Finally, we consider the genuine 3D modes from Table 2. We found four different solutions S6–S9, but all of them are unstable in the parameter range studied. Table 4 lists the initial guesses used and points to the corresponding figures in the appendix. According to Table 2, we know that the zero solution is unstable with respect to modes S7–S9. The eigenvalue has a high geometric multiplicity: we computed it numerically and found 24 equal positive eigenvalues. To single out the stable branch originating from the bifurcation is therefore not trivial. To do so one should exploit the symmetry of the problem as indicated in the paragraph on symmetries in Section 2.

The results presented above agree with the observation by de Wit et al. [39], who studied the well known Brusselator reaction–diffusion model in 3D and demonstrated possible symmetry structures with high dimension, i.e. body centered cubic (BCC), hexagonally packed cylinders (HPC), and also lamellae structures. Our results provide richer information on the pattern formation and change depending on different parameters. In [21,38] model Eq. (2) introduced by Barrio et al. is

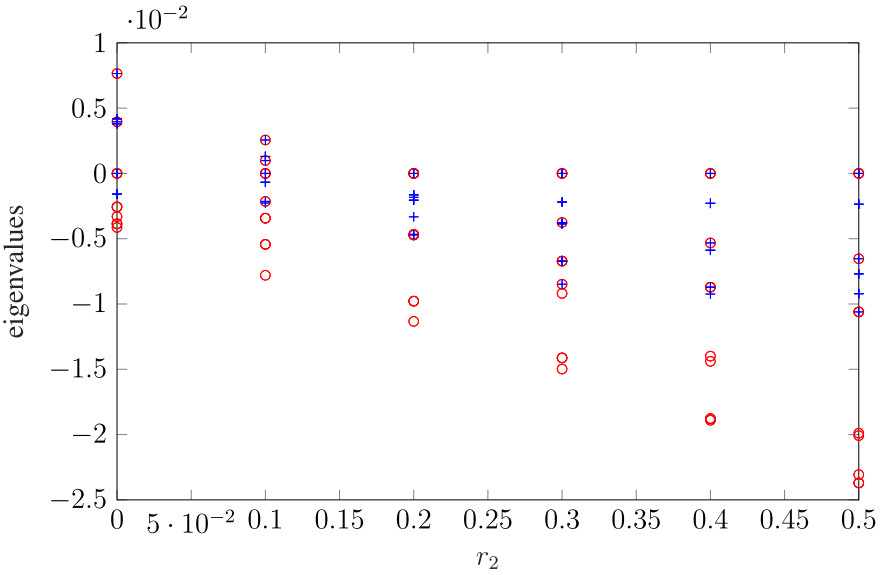

Fig. 8. 2D eigenvalues (○) and 3D eigenvalues (+) as a function of r_2 on the S4 branch.

Table 4
List of genuinely 3D solutions and initial conditions used to find them. All of these modes are unstable in the range $0 \le r_2 \le 1$.

Name	Function ϕ for initial guess (u, v)	Pattern
S6	$\sin(\kappa x)\sin(\kappa y)\sin(\kappa z)[1, -1]$	Droplet (Fig. 19)
S7	$\cos(2\kappa x)\cos(\kappa y)\cos(\kappa z)[1, -1]$	Droplet (Fig. 20)
S8	$\cos(2\kappa x + \kappa y + \kappa z)[1, -1]$	Tilted lamellae (Fig. 21)
S9	$\cos(2\kappa x + \kappa y)\cos(\kappa z)[1, -1]$	Tilted cylinder (Fig. 22)

investigated. Their results agree with ours as well. The HPC and lamellae patterns are generalizations of 2D solutions and have the same bifurcation diagram as in 2D.

4.4. Numerical performance

In this section, we want to give an impression on the effectiveness of the continuation process, and in particular the linear and eigenvalue solvers, which constitute most of the runtime in 3D. We use the number of sparse matrix-vector products (matvecs) as a rough indicator of the cost of our solvers. Equally roughly one could say that one time step with Euler's method, used by many other authors, would cost about one matvec per time step. We note that the number of time steps reported in papers such as [21,28] is on the order of hundred thousands or even millions for reaching a single steady state for this kind of problem.

In Fig. 9, the number of matvecs is shown for the solution of the nonlinear system (left) and the eigenvalue computation (right) at the different parameters. The number of Newton steps per parameter value is typically 3 or 4, and an adaptive tolerance of the inner GMRES solver is used to save some iterations. The GMRES method is restarted after 50 iterations to save memory and orthogonalization time. In the Jacobi–Davidson eigensolver we use a block size of 4 and allow at most 25 inner GMRES iterations to achieve an adaptively computed tolerance for the correction equation. Default settings are used for the smoothed aggregation (SA) AMG preconditioner ML from Trilinos version 11.12.1.

On the left one sees that the number of matvecs increases until the bifurcation point $r_2 = 0.25$, where the branch becomes unstable. After that the number of iterations decreases slightly, possibly indicating that we are getting away from the singularity. In the right panel, note that the first eigenvalue computation is particularly expensive. This is because no approximate eigenspace is available for starting the block JDQR method. Hence, it also shows that reusing the space from the previous step is advantageous, because in the end it reduces the amount of work by a factor 3. Comparing the left and right panel in Fig. 9 one observes that eventually the cost for the linear stability analysis (i.e. the eigenvalue computation) is about twice that of the actual solution of the non-linear problem. Of course this can be reduced by requiring fewer eigenvalues. Moreover, one could make the amount dependent on the situation, e.g. all the eigenvalues that are within a certain distance from the imaginary axis.

In order to give an impression of the performance of the linear solver, we followed a branch of steady states on a 128^3 grid, running on the 64 cores of an Intel Xeon Phi 7210 ("Knight's Landing") many-core processor (core frequency 1.3 GHz

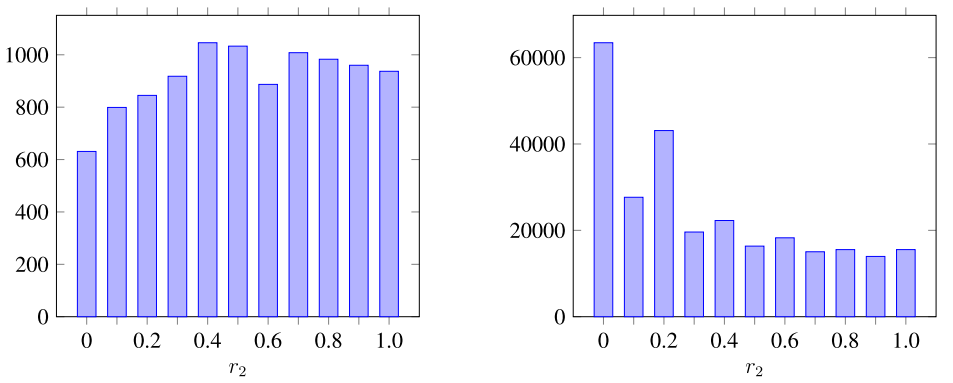

Fig. 9. Number of matvecs for computing a sequence of steady states (left) and their associated 10 right-most eigenpairs (right) on branch S3 with parameter r_2 varying from 0.0 to 1.0 on a 128^3 grid.

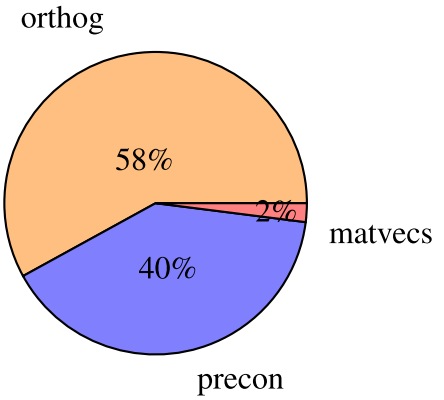

Fig. 10. The runtime percentage of the three main operations for computing a sequence of steady states on branch S3 with parameter r_2 varying from 0.0 to 1.0 on a 128^3 grid.

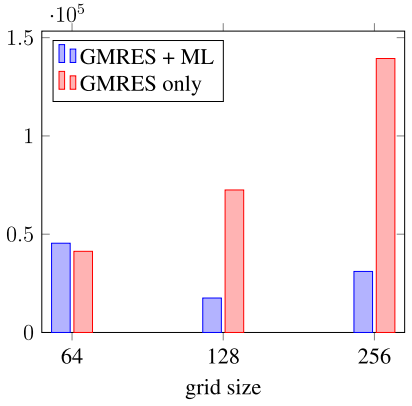

Fig. 11. Number of matvecs in an eigenvalue computation with and without preconditioning for different grid sizes.

and configured in cluster/cache mode). In Fig. 10, we give a breakdown of the actual runtime. A thorough performance analysis is not the goal here as all of the building blocks are freely available software and their performance has been studied elsewhere. The average time required for solving one linear system is 36 s. We display the runtime percentage of the three most expensive operations: orthogonalization of the subspace in GMRES(50) (orthog), preconditioner applications (precon) and matrix-vector products (matvecs). Most of the time is spent in orthogonalization. This can be reduced by decreasing the restart parameter m in GMRES(m) at the cost of more iterations and hence more matvecs and preconditioner applications. The matrix-vector product can be executed very efficiently in parallel here because of the simple matrix structure and therefore only has a minor contribution to the overall runtime.

Fig. 11 shows the effect of using the ML preconditioning in the Jacobi–Davidson eigensolver. We see a significant decrease of matvecs for large grid sizes. Note that the Jacobi–Davidson method performs some preconditioning of the equations solved by projecting out approximate and converged eigenmodes even if only GMRES is used as a correction solver. Therefore, we expect the gap between the unpreconditioned and preconditioned solver to be even larger when computing the steady states on the branch. The actual time spent solving both linear and eigenvalue problems depends on the balance between orthogonalization and preconditioner applications and can be optimized by tuning solver settings in production runs.

5. Conclusions and future work

We have shown that numerical continuation techniques, combined with efficient multigrid and Jacobi–Davidson solvers, are a very effective way to analyze nonlinear PDEs describing reaction/diffusion processes. Compared to previous results on this type of problem, we presented richer bifurcation diagrams in 2D and 3D with higher spatial resolution.

The numerical performance of the overall approach was demonstrated by reporting the number of operations required and giving an indication of the run time. We note that much optimization can be done to solve such problems more efficiently: geometric multigrid and matrix-free methods for structured grids, hybrid parallelization and SIMD usage (see [37]), exploiting the many symmetries in 3D, etc. In 3D, we did not investigate the complete bifurcation structure as systematically as in 2D. Instead we demonstrated the feasibility of such an investigation and leave it to domain scientists with a concrete application in e.g. computational biology or chemistry.

Our own future work will focus on developing scalable linear algebra algorithms and software to perform bifurcation analysis for other classes of nonlinear PDEs and apply them to e.g. realistic fluid flow problems.

Acknowledgments

The work of the third author was supported by the German Research Council (DFG) priority program 1648 ('Software for Exascale Computing'), project ESSEX. The contribution of S.B. is part of the Mathematics of Planet Earth research program, which is financed by the Netherlands Organization for Scientific Research (NWO).

Appendix

In the appendices, we present some of the solution patterns found in 2D and 3D. In all of the plots, the left column depicts the u-variable and the right column the v-variable. All results were obtained using a spatial resolution of 128 grid points per direction.

Appendix A. Patterns of 2D solutions

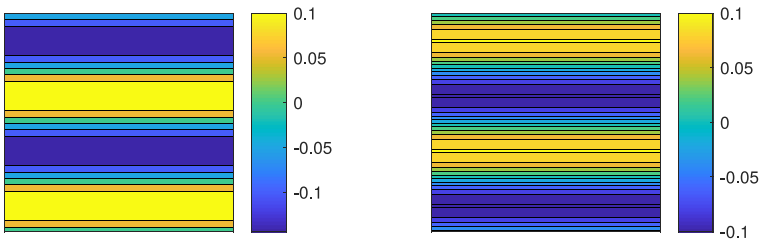

Fig. 12. S2 unstable stripe solution at $r_2 = 0.0$.

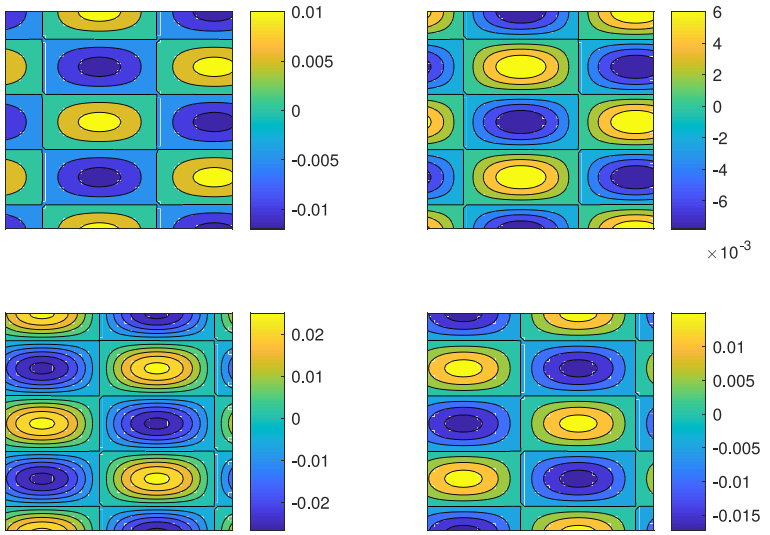

Fig. 13. Two independent eigenvectors corresponding to the positive eigenvalue of branch S2 with a pattern of an S4 mode, with $r_2 = 0.0$.

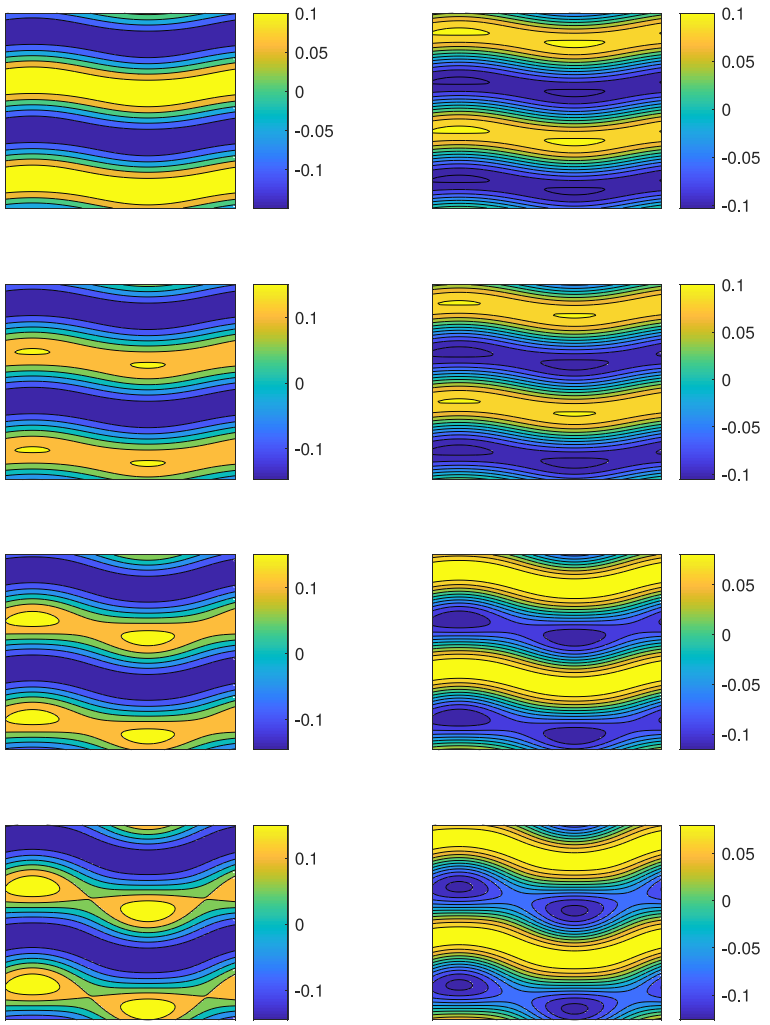

Fig. 14. S5 stable mixed stripe and spot solution at r_2=0.0, 0.1, 0.2, 0.21227 from top to bottom.

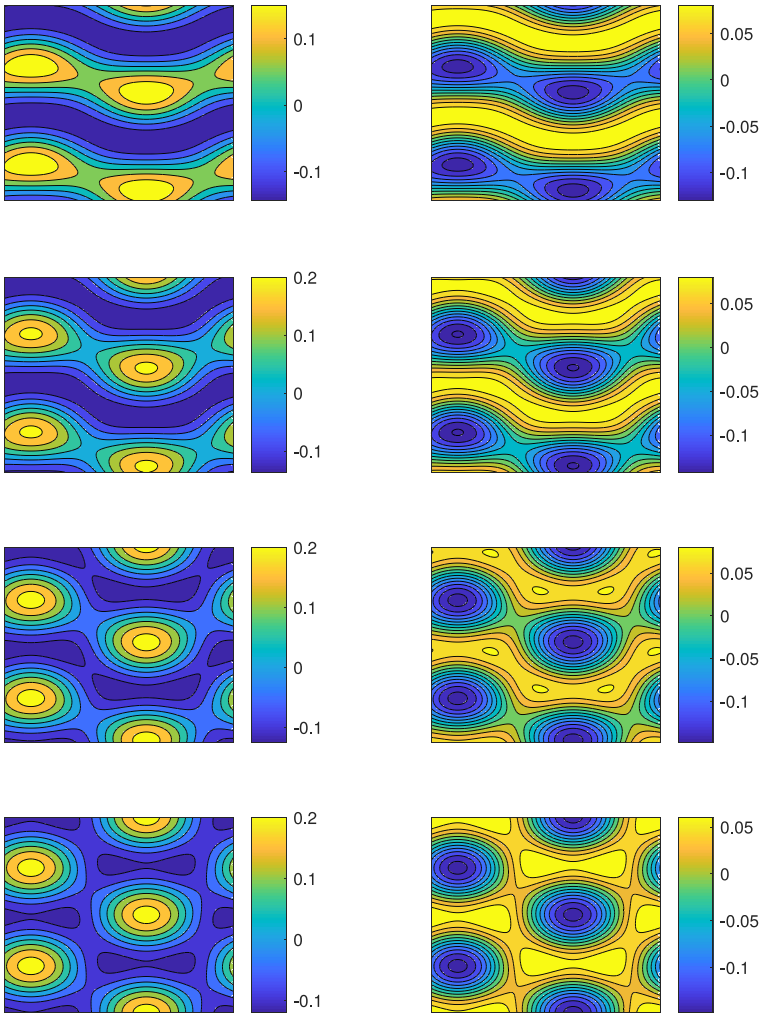

Fig. 15. S5 after turning point: unstable mixed stripe and spot solution at $r_2 = 0.21, 0.19, 0.15, 0.129$ from top to bottom.

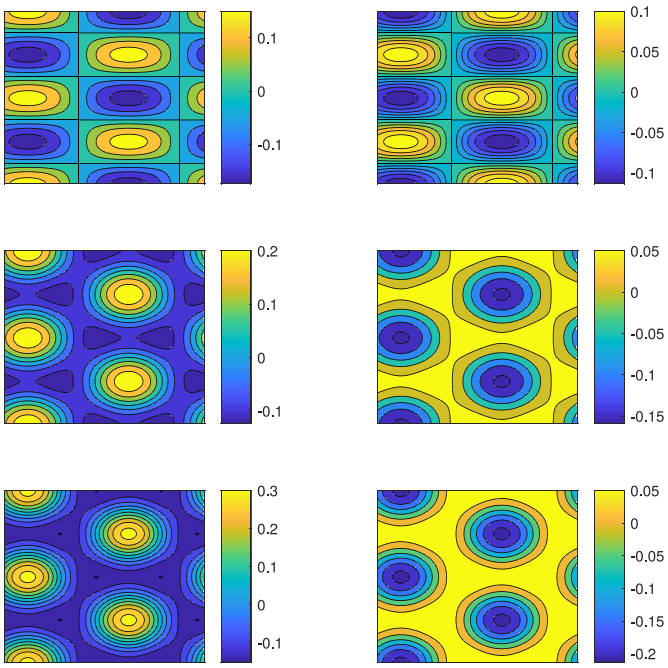

Fig. 16. S4 spot solution becoming stable (shown at $r_2 = 0.0, 0.2, 0.5$ from top to bottom).

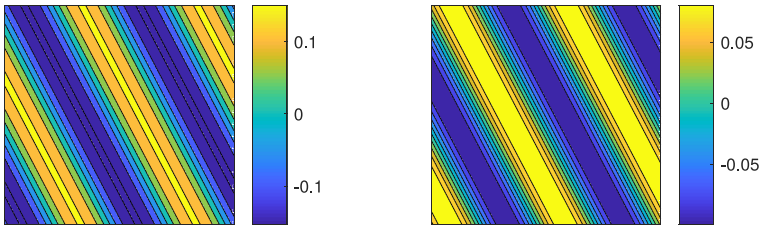

Fig. 17. S3 stable stripe solution at $r_2 = 0.0$ that will eventually lose stability.

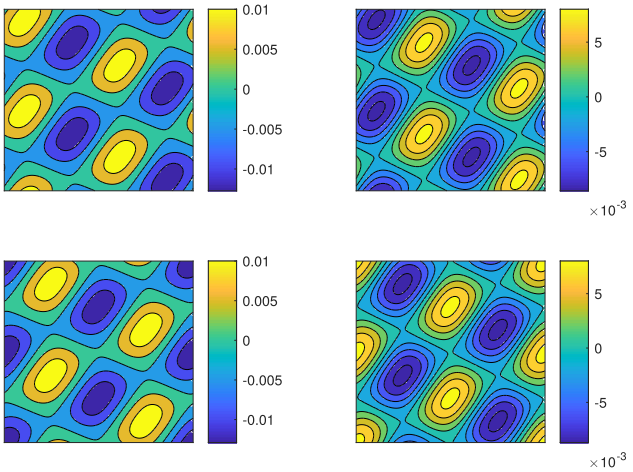

Fig. 18. Two independent eigenvectors corresponding to the positive eigenvalue of branch S3, $r_2 = 0.25$.

Appendix B. Patterns of 3D solutions

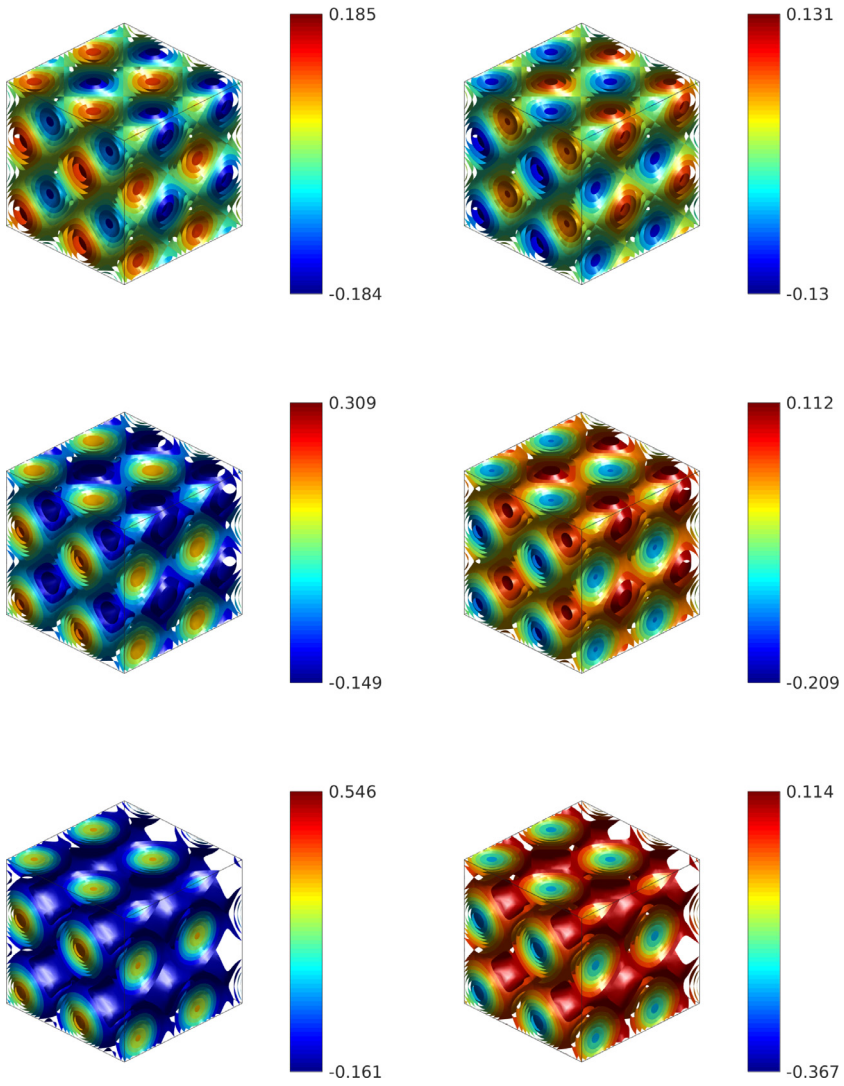

Fig. 19. Isosurface of 3D solution of pattern S6, with $r_2 = 0.0$, 0.5, 1.0 from top to bottom.

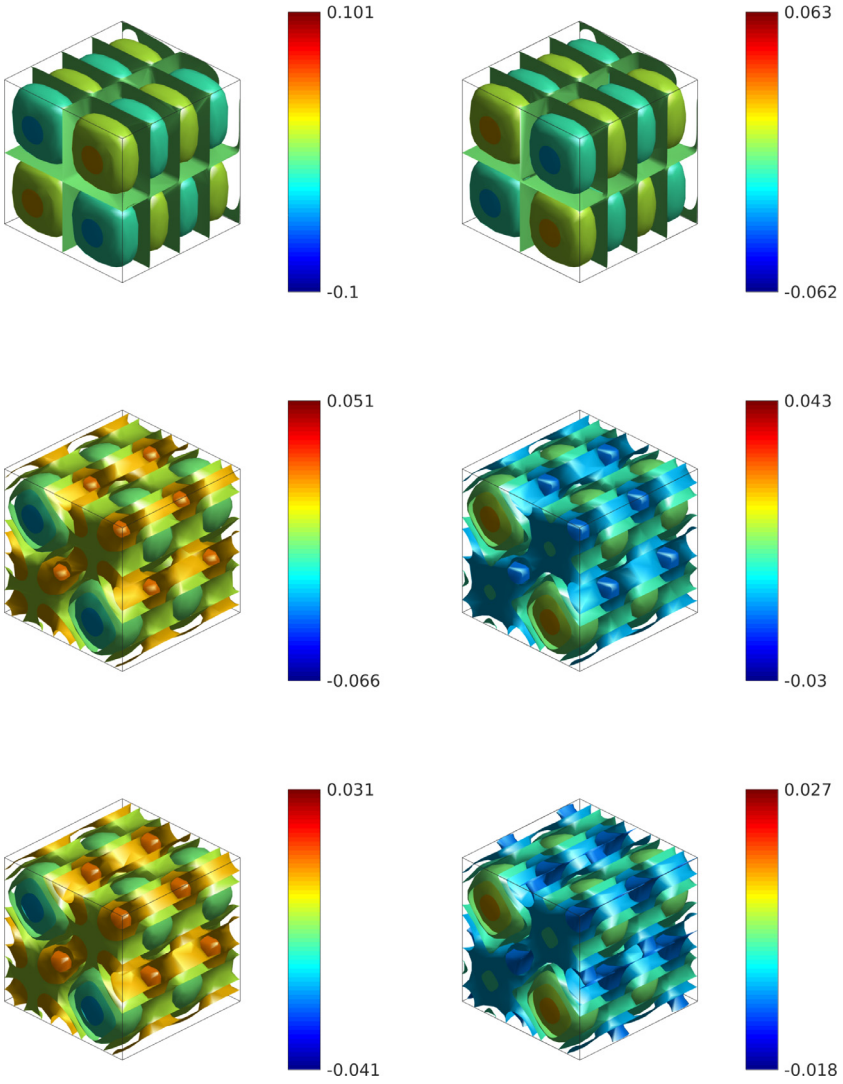

Fig. 20. Isosurface of 3D solution of pattern S7, with $r_2 = 0.0$, 0.5, 1.0 from top to bottom.

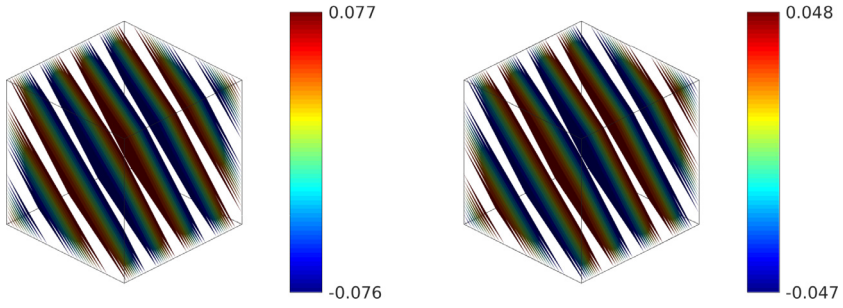

Fig. 21. Isosurface of 3D solution of pattern S8, with $r_2 = 0.0$.

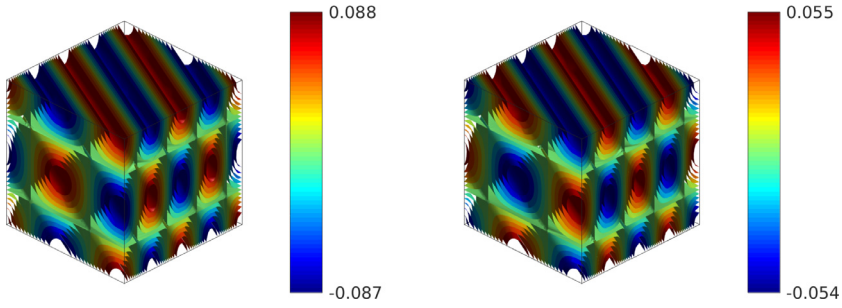

Fig. 22. Isosurface of 3D solution of pattern S9, with $r_2 = 0.0$.

References

[1] Alexandroff P, Hopf H. Topologie. Origedition. Berlin, Germany 1935: Chelsea Publication Co; 1965.
[2] Allgower E, Georg K. Introduction to numerical continuation methods. Soc Indust Appl Math; 2003. doi:10.1137/1.9780898719154.
[3] Keller H. Numerical solution of bifurcation and nonlinear eigenvalue problems. Appl Bifurc Theory 1977;1(38):359–84.
[4] Dijkstra H, Wubs F, Cliffe A, Doedel E, Dragomirescu I, Eckhardt B, et al. Numerical bifurcation methods and their application to fluid dynamics: analysis beyond simulation. Commun Comput Phys 2014;15(1):2–38. doi:10.4208/cicp.240912.180613a.
[5] Charru F. Hydrodynamic instabilities. Cambridge University Press; 2011. ISBN 9780511975172. Cambridge Books Online
[6] Saad Y. Numerical methods for large eigenvalue problems, 66. revised ed. SIAM; 2011.
[7] Sleijpen G, der Vorst HV. A Jacobi–Davidson iteration method for linear eigenvalue problems. SIAM Rev 2000;42(2):267–93.
[8] Fokkema D, Sleijpen G, der Vorst HV. Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils. SIAM J Sci Comput 1996;20(1).
[9] Turing A. The chemical basis of morphogenesis. Philos Trans R Soc Lond B Biol Sci 1952;237(641):37–72. doi:10.1098/rstb.1952.0012.
[10] Yang L, Dolnik M, Zhabotinsky A, Epstein I. Turing patterns beyond hexagons and stripes. Chaos 2006;16(3).
[11] Boissonade J, Dulos E, Kepper PD. Turing patterns: From myth to reality. Dordrecht: Springer Netherlands; 1995. p. 221–68.
[12] Dulos E, Boissonade J, Perraud J, Rudovics B, Kepper PD. Chemical morphogenesis: turing patterns in an experimental chemical system. Acta Biotheor 1996;44(3–4):249–61.
[13] Maini P, Painter K, Chau HNP. Spatial pattern formation in chemical and biological systems. J Chem Soc Faraday Trans 1997;93:3601–10.
[14] Callahan T, Knobloch E. Pattern formation in three-dimensional reaction–diffusion systems. Phys D 1999;132(3):339–62.
[15] Callahan T. Turing patterns with o(3) symmetry. Phys D 2004;188(1–2):65–91. doi:10.1016/S0167-2789(03)00286-0.
[16] Liu P, Shi J, Wang R, Wang Y. Bifurcation analysis of a generic reaction-diffusion turing model. Internat J Bifur Chaos Appl Sci Engrg 2014;24(4). doi:10.1142/S0218127414500424. 1450042, 12, ISSN 0218–1274
[17] Nakao H, Mikhailov AS. Turing patterns in network-organized activator–inhibitor systems. Nat Phys 2010;6:544–50.
[18] Asllani M, Busiello D, Carletti T, Fanelli D, Planchon G. Turing instabilities on cartesian product networks.. Sci Rep 2015;5(4517).
[19] McCullen N, Wagenknecht T. Pattern formation on networks: from localised activity to turing patterns. Sci Rep 2016;6(27397).
[20] Bánsági T, Vanag V, Epstein I. Tomography of reaction-diffusion microemulsions reveals three-dimensional turing patterns. Science 2011;331(6022):1309–12.
[21] Leppänen T, Karttunen M, Kaski K, Barrio R, Zhang L. A new dimension to turing patterns. Phys D Nonlinear Phenom 2002;168:35–44.
[22] Wit AD, Dewel G, Borckmans P, Walgraef D. Three-dimensional dissipative structures in reaction–diffusion systems. Phys D 1992;61(1):289–96. doi:10.1016/0167-2789(92)90173-K.
[23] Molnár F, Izsák F, Mészáros R, Lagzi I. Simulation of reaction–diffusion processes in three dimensions using Cuda. Chemometr. Intell. Lab. Syst. 2011;108(1):76–85. doi:10.1016/j.chemolab.2011.03.009. Cited By 10
[24] Shoji H, Ohta T. Computer simulations of three-dimensional turing patterns in the Lengyel–Epstein model. Phys Rev E 2015;91(3):032913,11. doi:10.1103/PhysRevE.91.032913.
[25] Nicolis G, Prigogine I. Self-Organisation in non-equilibrium chemical systems, 66. New York: Wiley; 1977.
[26] Gray P, Scott S. Autocatalytic reactions in the isothermal, continuous stirred tank reactor. Chem Eng Sci 1983;38(1):29–43. doi:10.1016/0009-2509(83)80132-8.
[27] Lengyel I, Rábai G, Epstein I. Batch oscillation in the reaction of chlorine dioxide with iodine and malonic acid. J Am Chem Soc 1990;112(11):4606–7.
[28] Barrio R, Varea C, Aragón J, Maini P. A two-dimensional numerical study of spatial pattern formation in interacting turing systems. Bull Math Biol 1999;61(3):483–505.
[29] Leppänen T. Computational studies of pattern formation in turing systems; 2004.
[30] Allgower EL, Georg K, Miranda R. Exploiting symmetry in applied and numerical analysis. American Mathematical Society; 1993.
[31] Golubitsky M, Stewart I, Schaeffer D. Singularities and groups in bifurcation theory: vol. II. Applied mathematical sciences, 69. Springer-Verlag, New York; 1988. ISBN 0-387-96652-8. doi:10.1007/978-1-4612-4574-2.
[32] Hydon P. Difference equations by differential equation methods. Cambridge monographs on applied and computational mathematics, 27. Cambridge: Cambridge University Press; 2014. ISBN 978-0-521-87852-4. doi:10.1017/CBO9781139016988.
[33] Heroux M, Bartlett R, Howle V, Hoekstra R, Hu J, Kolda T, et al. An overview of trilinos Technical Report. Sandia National Laboratories; 2003.
[34] Thies J.. et al. The PHIST software repository. accessed 2017-09-13. URL https://bitbucket.org/essex/phist/.
[35] Seydel R. Practical bifurcation and stability analysis. Springer; 2010.
[36] Gee M, Siefert C, Hu J, Tuminaro R, Sala M. ML 5.0 smoothed aggregation user's guide. Technical Report. Sandia National Laboratories; 2006.
[37] Röhrig-Zöllner M, Thies J, Kreutzer M, Alvermann A, Pieper A, Basermann A, et al. Increasing the performance of the Jacobi–Davidson method by blocking. SIAM J Sci Comput 2015;37(6):C697–722. doi:10.1137/140976017.
[38] Leppänen T, Karttunen M, Barrio R, Kaski K. Turing systems as models of complex pattern formation. Braz J Phys 2004;34:368–72.
[39] de Wit A, Borckmans P, Dewel G. Twist grain boundaries in three-dimensional lamellar Turing structures. Proc Natl Acad Sci USA 1997;94:2765–8.