NONLINEAR DYNAMICAL SYSTEMS PART 3 DYNAMICS IN PATTERN-FORMING SYSTEMS TUTORIAL 2

DANIELE AVITABILE[*]

Abstract. Themes of of this tutorial:
1. Inferring bifurcation diagrams using brute-force simulation (and its perils)
2. Numerical solution of boundary-value problems
3. Performing numerical continuation
4. Numerical study of linear stability of steady states

1. Introduction. In this tutorial we will continue working with the following Allen–Cahn PDE, subject to homogeneous Neumann (no-flux) boundary conditions

$$\begin{aligned} &\partial_t u = \nu \partial_{xx} u + \lambda u + \alpha u^2 + \beta u^3 - \gamma u^5, && (x,t) \in (-5,5) \times \mathbb{R}_{>0}, \\ &\partial_x u(-5,t) = \partial_x u(5,t) = 0, && t \in \mathbb{R}_{\geq 0}, \\ &u(x,0) = \varphi(x), && x \in [-5,5]. \end{aligned} \tag{1.1}$$

In Tutorial 1 we have derived, implemented, and time stepped a set of n coupled ODEs approximating (1.1),

$$\dot{U} = F(U,p), \qquad U(0) = \{\varphi(x_i)\}_{i=1}^n \tag{1.2}$$

where $U(t) \in \mathbb{R}^n$, contains an approximation to $u(x,t)$ at n grid points $\{x_i\}_{i=1}^n$ in $[-5,5]$, $p = (\nu, \lambda, \alpha, \beta, \gamma) \in \mathbb{R}^5$ is a vector containing all control parameters, and $F \colon \mathbb{R}^n \times \mathbb{R}^5 \to \mathbb{R}^n$.

Henceforth we will fix $\nu = \beta = \gamma = 1$, $\alpha = 0$, and use λ as principal continuation parameter. Owing to this convention we may write, with a slight abuse of notation, $F(u,\lambda)$ and $F(u,p)$ interchangeably.

Question 1 (Time simulations and bifurcation diagrams). Figure 1.1 presents a bifurcation diagram (without solution stability) for homogeneous equilibria of the Allen–Cahn equation. It was determined using analytical calculations involving the real-valued function G defined in Tutorial 1. We know from Tutorial 1 that heterogeneous equilibria are also supported by the system. Where are they located in parameter space? How do they emerge? Do they bifurcate from the trivial state $C = 0$, or from one of the other homogeneous steady states? How can we compute linear stability of PDE equilibria? We will address these questions in this tutorial. For now, we will uncover partial information using the time stepper coded above.

If a time simulation for $t \in [0,T]$ approaches an equilibrium, then we can compute the solution measure S_2 of the final state $u(x,T)$ (which is an approximation of the equilibrium $u_*(x)$), and transfer this information on the bifurcation diagram.

Reproduce Figure 1.1 by launching two simulations for $\lambda = -0.4$ and $\lambda = 0.7$, respectively. There will be some detective work involved in the process: you must select appropriate T and $\varphi(x)$ that produce the desired homogeneous equilibrium.

[*]Vrije Universiteit Amsterdam, Department of Mathematics, Faculteit der Exacte Wetenschappen, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.
Inria Sophia Antipolis Méditerranée Research Centre, MathNeuro Team, 2004 route des Lucioles-Boîte Postale 93 06902, Sophia Antipolis, Cedex, France.
(d.avitabile@vu.nl, www.danieleavitabile.com).

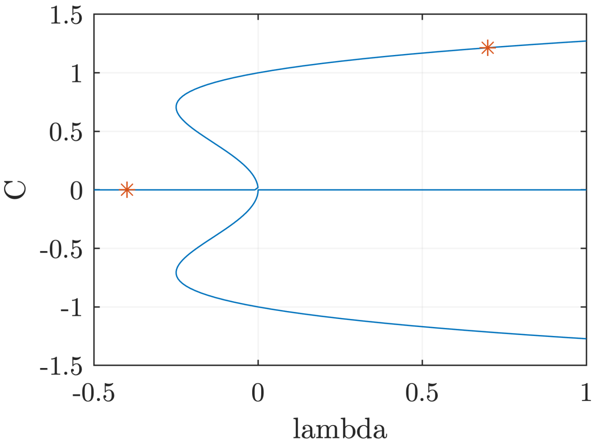

Fig. 1.1. *Equilibria of homogeneous steady states in the parameter λ and solution measure $S_1 = C$ for* (1.1) *with parameters $\mu = 1$, $\alpha = 0$, $\beta = 1$, $\gamma = 1$. The markers indicate steady states obtained by time simulations for $\lambda = -0.4, 0.7$.*

This is to be expected: from what you have seen in past questions, the system is nonlinear and supports coexisting stable states.

Question 2 (Brute-force continuation of steady state). Using the time stepper we can uncover selected branches of the bifurcation diagram, and “join the dots” between the two markers in Figure 1.1, by varying λ in small steps. The main idea is to daisy-chain numerical simulations: in each new simulation we perturb λ, and take as initial condition for the new simulation the equilibrium found for the previous value of λ. In this way we traverse the bifurcation diagram in small λ steps.

This approach, is called *brute-force* continuation of steady states, and we summarise it in the pseudocode below

```
lambdaValues = 0.7:-0.05:0.1;
u0 = ...

for lambda = lambdaValues
   Update p with the new value of lambda
   Prepare ODEs with the new p
   Time step with parameters p, and initial condition u0
   Store the final state in uF
   Check that uF is approximately an equilibrium
   Compute the solution measure of uF
   u0 = uF
end

Plot lambda versus solution measure along the branch
```

Your task for this question is to reflect on the algorithm above, implement it to perform brute-force continuation of homogeneous steady states, from $\lambda = 0.7$ to $\lambda = -0.5$, with steps $\Delta\lambda = -0.02$, and superimpose the brute-force bifurcation diagram to the one computed using G. The direction of the steps is important, as going from $\lambda = 0.7$ to $\lambda = -0.5$ gives different results than going from $\lambda = -0.5$ to

$\lambda = 0.7$ (can you guess why?).

The pseudocode above states on line 9 that one must check that the final state $u(x, T)$ is approximately an equilibrium. How do you plan to do this? One way is by using the function `AllenCahn`, which computes $F(u, p)$.

Give a brief description of your results. Which part of the bifurcation diagram do you uncover, and why?

Question 3 (Brute-force continuation of heterogeneous states). We can now use brute-force continuation to compute branches of stable heterogeneous states. We are aware that such solutions exist, from the numerical simulations in Question 10 of Tutorial 1. Compute a bifurcation diagram from $\lambda = 0.7$ down to $\lambda = -0.5$, as done in Question 2, and superimpose it to the bifurcation diagram of the previous questions. You will need an initial condition that leads to a heterogeneous equilibrium for $\lambda = 0.7$, and you can use the knowledge acquired in Question 10 of Tutorial 1 to this end. Describe your results. If your branch becomes unstable at a bifurcation point, give an interval of λ containing the bifurcation point (no need to characterise the bifurcation yet).

Question 4 (A sigmoidal patterned state). Time step the system for $\lambda = 0.7$, $\varphi(x) = \tanh(-x)/2$, and verify that the system supports a monotone decreasing, sigmoidal steady state. Looking at your bifurcation diagram you should now conclude that for $\lambda = 0.7$ the system has at least 3 coexisting stable steady states, which are selected depending on initial conditions.

Question 5 (Brute-force continuation of sigmoidal state). Repeat Question 3, to compute a branch of signoidal steady states.

SPOILER ALERT!

The next page contains a spoiler. Do not turn the page before attempting Question 5.

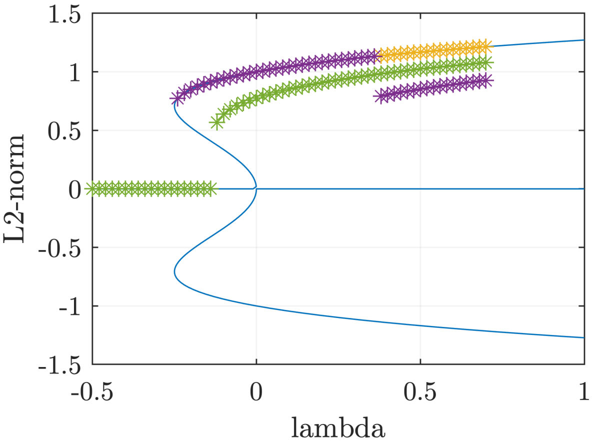

Fig. 1.2. *Brute-force bifurcation diagram after Question 4. Asterisks denote brute-force continuation started from a homogeneous steady state (yellow) a patterned "bump" state (green) or a patterned sigmoidal state (purple) for $\lambda = 0.7$, and continuing down to $\lambda = -0.5$. The continuation jumps at several points, and some branches overlap (a yellow and purple branch, for instance, lay underneath the green branch of trivial homogeneous states $u(x) \equiv 0$). See solutions for a more thorough description.*

Question 6. (Preparing for numerical continuation) You should now have a bifurcation diagram such as the one in Figure 1.2. We will now prepare to perform a better form of numerical continuation: more reliable, and able to compute and classify stable as well as unstable states. Let us start by preparing initial guesses for $\lambda = 0.7$. Run 3 time simulations for $t \in [0, 50]$ and $\lambda = 0.7$, so as to obtain approximations to: (i) a homogeneous steady state; (ii) a heterogeneous, stationary, bump state; (iii) a heterogeneous, stationary sigmoidal state. Save the solution profile $u(x, 50)$, as well as the parameters p in a suitable `.mat` file, which you will later reuse for continuation.

Question 7. (Solving boundary-value problems) As seen in the lectures, any equilibrium of the PDE (1.1) (irrespective of its stability), satisfies the boundary-value problem[1]

$$\begin{aligned} &\nu \partial_{xx} u + \lambda u + \alpha u^2 + \beta u^3 - \gamma u^5 = 0, \qquad x \in (-5, 5), \\ &\partial_x u(-5) = \partial_x u(5) = 0, \end{aligned} \tag{1.3}$$

or, in discrete form

$$0 = \nu D_{xx} U + N(U, \lambda, \alpha, \beta, \gamma) =: F(U, p) \tag{1.4}$$

For fixed $p \in \mathbb{R}^5$, one can approximate equilibria by solving the set of n nonlinear equations (1.4), using Newton's method, or one of its variants. This method of finding equilibria is alternative to time-stepping, and is at the heart of numerical continuation.

Newton's method require evaluations of the function $U \mapsto F(U, p)$, and of the Jacobian matrix $U \mapsto D_U F(U, p)$, where $p \in \mathbb{R}^p$ is fixed. Recall that we have already coded the function F in `AllenCahn.m`. We can now amend this function, so as to

[1] Recall that in our notation, ∂_{xx} indicates, with a slight abuse of notation, both the partial derivative of a bivariate function, say $u(x, t)$, and an operator that maps a single-variable function $u(x)$ to its second derivative. We are using the latter here.

return Jacobian evaluations when required. This new function can then be passed to Matlab's `fsolve` function, which implements a variant of Newton's method.

1. Compute with pen and paper the n-by-n Jacobian matrix $D_U F(U,p)$. Express this matrix using D_{xx} and a suitably-defined diagonal matrix containing the derivatives of N.
2. Modify the function `AllenCahn` so as to have the following interface

   ```
   function [F,DFDU] = AllenCahn(u,p,Dxx)
   ```

   If `nargout = 1`, the function returns only `F`, as the previous version of the function `AllenCahn`. If `nargout = 2`, the function returns `F` as well as the Jacobian matrix `DFDU`. This means that all function calls below are now possible

   ```
   >> F = AllenCahn(u,p,Dxx);
   >> [F,DFDU] = AllenCahn(u,p,Dxx);
   >> [~,DFDU] = AllenCahn(u,p,Dxx);
   ```

   This type of interface is also useful to pass functions to `fsolve`.
3. Use Matlab's in-built function `fsolve` to solve the discretised boundary-value problem (1.4). You should aim to obtain a sigmoidal steady state for $\lambda = 0.7$, so you must set the parameter vector p accordingly, and may select as initial guess the sigmoidal solution saved in Question 6, which is an approximation to the equilibrium. It is useful that the `.mat` file stores the profile u as well as the vector p, so you can pass p directly the correct parameters to `AllenCahn`. The function `fsolve` interfaces with functions of the type $U \mapsto F(U,p)$, and $U \mapsto D_U F(U,p)$, hence you should pass to it a function handle such as

   ```
     ACProblem = @(u) AllenCahn(u,p,Dxx);
   ```

   and set the option `'Jacobian'` to `'on'` to evaluate Jacobians. This will prompt `fsolve` to call the function handle using either one of the two expressions below:

   ```
     F = ACProblem(u), [F,DFDU] = ACProblem(u);
   ```

   thereby implementing $U \mapsto F(U,p)$ and $U \mapsto D_U F(U,p)$. It is useful to examine other options of `fsolve`, such as `Display`, `TolFun`, and `MaxIter`, for instance.
4. Plot the initial guess and solution to the BVP. Since the initial guess obtained by time simulation is close to an equilibrium, the functions should be very similar, and `fsolve` should use only few iterations to converge.

Question 8 (Perturb and find a new equlibrium). In the previous question we found an equilibrium for $\lambda = 0.7$. We can use the same initial guess and attempt to find an equilibrium for another value of λ. Use the same initial guess and procedure of Question 7 to find an equilibrium for $\lambda = 0.4$. Since the initial guess is now less accurate (we are using as initial guess an equilibrium for $\lambda = 0.7$ to solve the BVP for $\lambda = 0.4$), you should observe that convergence is achieved, but using more steps than before.

Intermezzo: recapitulation and remarks on numerical continuation. Let us pause for a moment to reflect on our results: we have now seen two strategies for

Listing 1
Continuation demo for a scalar nonlinear problem

```
% Clean
clear all, close all, clc

% Setup
u0 = 3;                      % Initial conditions
p0 = [2,1];                  % Parameters [lambda,beta]
prob = @(u,p) Poly135(u,p); % Problem definition

% Continuation parameter index:
% icp = 1 means continuation in p(1)
icp = 1;

% Parameter stepsize:
% a negative stepsize gives an initially decreasing parameter
ds = -0.05;

% Number of continuation steps
nSteps = 80;

% Options to pass to fsolve
opts = optimset('Display','off', 'TolFun',1e-10,...
                'MaxIter',50, 'Jacobian','on');

% Launch continuation
[bd,sol] = SecantContinuation(prob,u0,p0,icp,ds,nSteps,opts);

% Plotting
plot(bd(:,2), bd(:,3), '.-');
xlabel('lambda'); ylabel('2-norm');

% Right-hand side function
function [F,DFDU] = Poly135(u,p);
   F = p(1)*u + p(2)*u^3 - u^5;
   if nargout > 1
     DFDU = p(1) + 3*p(2)*u^2 -5*u^4;
   end
end
```

computing curves of equilibria to the Allen–Cahn problem: (i) brute-force continuation is based on time stepping an initial condition long enough, until we obtain an equilibrium, and then using the final condition to time step a new problem, in which λ is perturbed; (ii) the procedure outlined in Questions 7 and 8 substitutes time stepping with solving a BVP.

Numerical continuation adopts a strategy similar to (ii), that is, it computes equilibria using Newton's method. A full understanding of this tool is out of scope here: you are provided with a function implementing *secant continuation*.

The function `SecantContinuation` takes the function F, and several other parameters associated to the continuation (see explanations in Listing 1), and produces a branch of steady states for the problem $\dot{U} = F(U, p)$, in the parameter $p($`icp`$)$, using the Euclidean 2-norm of U as solution measure.

In the example in Listing 1 we show how to use `SecantContinuation` to continue steady states of the scalar ODE $\dot{U} = \lambda U + \beta U^3 - U^5$ in the parameter λ. The function returns:

- A matrix `bd` with `nsteps`+1 rows and 3 columns: `bd(1,:)` contains a label for points in the branch, from 0 to `nsteps`; `bd(2,:)` contains values of the continuation parameter $p($`icp`$)$; `bd(3,:)` contains values of $\|U\|$. This matrix is useful to plot bifurcation diagrams.
- A matrix `sol` with `nsteps`+1 rows and n columns: `sol(i,:)` contains the steady state U computed for $p($`icp`$)$ = `bd(i,2)`. This matrix is useful to fetch specific solutions on the bifurcation diagram, and post-process them.

Question 9 (Continuation of sigmoidal equilibria). We are now ready to recompute the bifurcation diagram in Figure 1.1 with more generality. Compute a branch of sigmoidal steady states, and plot the resulting bifurcation diagram. It may be useful to start from one of the `.mat` files saved in Question 6.

Question 10 (Continuation of bump equilibria). Compute a branch of bump steady states, and include it in the bifurcation diagram of Question 9.

Question 11 (Continuation of homogeneous states). Compute branches of homogeneous steady states, and include them in the bifurcation diagram of Question 9.

The next page contains a spoiler. Do not turn the page before attempting Question 11.

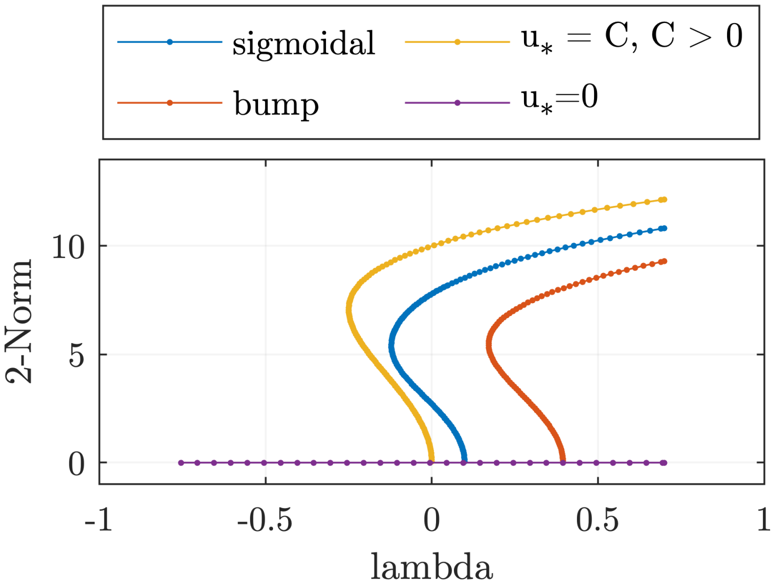

FIG. 1.3. *Selected solution branches of stead states of (1.1), obtained with numerical continuation.*

Question 12 (Stability of $u_* \equiv 0$ (analytic)). You should have obtained a plot with various branches, as shown in Figure 1.3, and we can now proceed to discuss linear stability. Using pen and paper, find an approximate linear evolution equation for small perturbations $v(x,t)$ of a steady state $u_*(x)$ of (1.1). Deduce that small perturbations around $u_*(x) \equiv 0$ evolve as

$$\begin{aligned} &\partial_t v(x,t) = \partial_{xx} v(x,t) + \lambda v(x,t), && (x,t) \in (-5,5) \times \mathbb{R}_{>0} \\ &\partial_x v(\pm,5) = 0, && t \in \mathbb{R}_{\geq 0}. \end{aligned} \tag{1.5}$$

Show that the branch of steady states $u_*(x) \equiv 0$ undergoes countably many bifurcations at $\lambda = \lambda_k$, $k \in \mathbb{N}$ (find an expression for λ_k), and with the corresponding critical modes $\psi(x) = \psi_k(x)$ (find an expression for $\psi_k(x)$).

Verify that the 3 branches of nontrivial steady states in Figure 1.3 bifurcate from the branch of trivial steady states ($u_*(x) \equiv 0$) at $\lambda = \lambda_0, \lambda_1, \lambda_2$, respectively. You can do so by comparing the analytical values $\lambda_{0,1,2}$ to the plot.

Question 13 (Stability of equilibria (numerical)). In this question we learn how to obtain information on the linear stability of a generic solution $u_*(x)$, by approximating eigenvalues and eigenfunctions of the linearised problem.

Formally, if one casts the (1.1) as

$$\partial_t u = \mathcal{F}(u,p) \quad \text{on } (-5,5) \times \mathbb{R}_{>0}, \qquad \partial_x u = 0 \quad \text{on } \{-5,5\} \times \mathbb{R}_{\geq 0},$$

then the linear stability of an equilibrium u_* attained for $p = p_*$ is studied via the linearised problem

$$\partial_t v = \partial_u \mathcal{F}(u_*,p_*)v \quad \text{on } (-5,5) \times \mathbb{R}_{>0}, \qquad \partial_x v = 0 \quad \text{on } \{-5,5\} \times \mathbb{R}_{\geq 0},$$

and, more specifically, by solving the eigenvalue problem

$$\partial_u \mathcal{F}(u_*(x),p_*)\psi(x) = \mu\psi(x) \quad \text{for } x \in (-5,5), \qquad \partial_x \psi(\pm 5) = 0. \tag{1.6}$$

In Question 12 we could solve the eigenvalue problem (1.6) analytically for the trivial solution $u_*(x) \equiv 0$, but this is the exception rather than the rule: in general one does not know $u_*(x)$ in closed form, and hence can not solve (1.6).

The work done in the previous questions, however, puts us in a favourable position to study stability numerically: we have an approximation $U_* \in \mathbb{R}^n$ to $u_*(x)$, and we can solve the eigenvalue problem which discretises (1.6), namely

$$D_U F(U_*, p_*)\Psi = \mu \Psi \tag{1.7}$$

where the n-by-n Jacobian matrix $D_U F(U_*, p_*)$ encapsulates boundary conditions, and where $\Psi \in \mathbb{C}^n$ is an approximation to the eigenfunction $\psi(x)$.

In fact, we already have a routine that provides an approximation to the matrix $D_U F(U_*, \lambda_*)$ (see Question 7): for a given solution `u` and parameter set `p`, this is obtained via the commands.

```
[~,DFDU] = ACProblem(u,p,Dxx);
```

All we need to do to solve (1.7) is calling Matlab's routine `eig` which computes eigenvalues and eigenfunctions of a matrix.

Use `eig` to approximate eigenvalues and eigenfunctions for the trivial steady state $u_*(x) \equiv 0$ for $\lambda = 0$. Use Matlab's command `sort` to order the eigenvalues according to the magnitude of their real part, in descending order.

$$\operatorname{Re}\mu_1 \geq \cdots \geq \operatorname{Re}\mu_{i-1} \geq \operatorname{Re}\mu_i \geq \operatorname{Re}\mu_{i+1} \geq \cdots \geq \operatorname{Re}\mu_n$$

Display the 5 most unstable eigenvalues $\{\mu_i\}_{i=1}^5$ and plot the corresponding eigenfunctions $\{\Psi_i\}_{i=1}^5$. Keeping in mind the results of Question 12, discuss with a colleague why these computations provide numerical evidence of the following statement: *the trivial stationary sate undergoes a bifurcation at $\lambda = \lambda_0 = 0$, with critical eigenfunction $\psi_0(x) \equiv c_0$, for $c_0 \in \mathbb{R}$.*

Question 14. Produce numerical evidence of the following statements:

1. the trivial stationary sate undergoes a bifurcation at $\lambda = \lambda_1 = (\pi/10)^2$; at this value of λ the spectrum of the linearised operator has one unstable eigenvalue, and one eigenvalue at 0; the equilibrium $u_*(x) \equiv 0$ for $\lambda = \lambda_1$ is unstable to perturbations $\psi_0(x) \equiv c_0$, and has critical eigenfunction $\psi_1(x) = c_1 \sin(\pi x/10)$, with $c_0, c_1 \in \mathbb{R}$.
2. the trivial stationary sate undergoes a bifurcation at $\lambda = \lambda_2 = (\pi/5)^2$; at this value of λ the spectrum of the linearised operator has two unstable eigenvalues, and one eigenvalue at 0; the equilibrium $u_*(x) \equiv 0$ for $\lambda = \lambda_2$ is unstable to perturbations $\psi_0(x) \equiv c_0$, $\psi_1(x) = c_1 \sin(\pi x/10)$, and has critical eigenfunction $\psi_2(x) = c_2 \cos(\pi x/5)$ with $c_1, c_2, c_3 \in \mathbb{R}$.

Question 15. With the tools built so far, you can explore in full the bifurcation diagram in Figure 1.3. This question is more open ended than the others and leaves room for your initiative. It is important that you think of a way of addressing the question at large (numerical simulation, numerical bifurcation analysis, analysis, a combination thereof), and possibly formulate new questions by yourself. These are a selection of open questions you may consider (they all conceal subtleties and directions you are invited to explore):

- Summarise what you have learnt about branches of steady states of the Allen–Cahn PDE.

- How many *stable* stationary states does the Allen–Cahn equation suport for $\lambda = 0.7$, and what is their spatial profile? What can we expect from a time simulation at $\lambda = 0.7$? Is your intuition backed-up by time simulations? What if $\lambda < 0.7$?
- Go back to Figure 1.1. The bifurcation diagram with solution measure $S_1 = C$ displays solutions with $C < 0$. What happened to them in Figure 1.3? And do branches of of heterogeneous states have this behaviour too?
- Throughout our study we have set $\alpha = 0$. Can you formulate any conjecture, on any solution branch, for the case $\alpha \neq 0$?
- As in any PDE, boundary conditions have a strong effect on the dynamics: what would change if instead of homogeneous Neumann one imposes homogeneous Dirichlet boundary conditions?