Institut für Mathematik, Universität Oldenburg, alexander.meiners@uol.de, hannes.uecker@uol.de

In [AKMU24] we revisited, in 1D with Neumann boundary conditions, the 2–component reaction system

$$\begin{aligned}\partial_t u_1 &= \partial_x^2 u_1 + \frac{u_2 - u_1}{(u_2 - u_1)^2 + 1} - \tau u_1,\\ \partial_t u_2 &= d\partial_x^2 u_2 + \alpha(j_0 - (u_2 - u_1)),\end{aligned} \tag{1}$$

from [MDWBS97], also considered in [Uec21a]. $u_1{=}u_1(t,x)$ is an interface charge, $u_2{=}u_2(t,x)$ is a voltage, and $(j_0, \alpha, D, \tau) \in \mathbb{R}^4$ is a parameter vector. Here we comment on the associated `pde2path` implementation in the folder `seco24`, available at [Uec24], and in particular the differences and extensions compared to the demo `pphome/demos/JBDMV/seco`, where `pphome` stands for the user's `pde2path` root–directory. For general background and usage of `pde2path` we refer to [Uec21b] and the tutorials at [Uec24].

For (1) we generally fix τ and D and use (α, j_0) as continuation parameters. The main difference between [Uec21a] and [AKMU24] is that in the latter we consider different parameter regimes, which are friendlier wrt branches of Turing–Hopf mixed modes and of localized standing waves, and hence more amendable to the discussed zipping–up and unzipping of such branches into snakes and stacks of isolas, respectively. For all (j_0, α, τ, D), (1) has the unique spatially homogeneous steady state

$$u^* = (u_1^*, u_2^*), \qquad u_1^* = \frac{j_0}{\tau(j_0^2+1)}, \quad u_2^* = j_0 + u_1^*, \tag{2}$$

For our parameter choice there are two codimension–2 Turing-Hopf points (C2TH points) $\mathrm{C2TH}_{1,2}$ near which u^* may undergo either a (steady) Turing or a Hopf bifurcation, see [AKMU24, Fig.1]. The main purpose of [AKMU24] is to study the interaction of both bifurcations: the "pure" modes take the form of time-periodic spatially homogeneous Hopf–branches (and long wavelength side–bands), and steady state spatially homogenous Turing branches, from which snaking branches of localized steady states (LSS) bifurcate in 2ndary bifurcations. On these, there are tertiary Hopf points (HPs) to Turing–Hopf mixed mode branches on which solutions consist of an essentially steady pattern in one part of the domain and an oscillating essentially spatially homogeneous background in the remainder. Additionally, on the LSS branches near $\mathrm{C2TH}_1$ there are also tertiary HPs to localized standing waves (LSWs).

The focus in [AKMU24] and the folder `seco24` is the continuation of these mixed mode branches. They can form long snaking branches where in every 2nd fold an additional spatial period is added (or taken away) from the spatially periodic part, or they can form of "short connections" between different BPs on the LSS snakes, or they can detach and form stacks of isolas. The connection between the three phenomena, "long snakes" vs "short connections" vs "isolas" is explored via 2–parameter continuation, for instance fold point continuation (FPC) and Hopf point continuation (HPC), and is complemented by some direct numerical integration (DNS). The implementation follows standard `pde2path` principles for (semilinear) reaction–diffusion systems, see, e.g., [Uec21b, §8], and no special tricks are needed. Table 1 lists the files.

[AKMU24] F. Al Saadi, E. Knobloch, A. Meiners, and H. Uecker. Localized steady and oscillatory states near a Turing–Hopf instability in a semiconductor model, 2024.

[MDWBS97] M. Meixner, A. De Wit, S. Bose, and E. Schöll. Generic spatiotemporal dynamics near codimension-two Turing-Hopf bifurcations. *Phys. Rev. E*, 55(6, part A):6690–6697, 1997.

[Uec21a] H. Uecker. Continuation and bifurcation for Nonlinear PDEs – algorithms, applications, and experiments. *Jahresbericht DMV*, 2021.

[Uec21b] H. Uecker. *Numerical continuation and bifurcation in Nonlinear PDEs.* SIAM, Philadelphia, PA, 2021.

[Uec24] H. Uecker. `www.staff.uni-oldenburg.de/hannes.uecker/pde2path`, 2024.

Table 1: Scripts and functions in `seco24`. Associated to most `cmds*` scripts are `cmds*plot` scripts for plotting; all figure numbers refer to [AKMU24]. First two blocks: scripts; 3rd block: problem describing functions; 4th block: overloads of `pde2path` library functions and convenience functions.

file	purpose, remarks
`cmds1a`	starting script near $C2TH_1$, generating Fig.2–4 (via `cmds1aplot`).
`cmds1FL`	details of first THMM bifurcation, with multipliers and spectral plots, Fig.5.
`cmds1DNS`	DNS with initial conditions near states from Figs.2–4, also including some pointwise time series and their fast Fourier transforms (FFT), Figs.6 and 7.
`cmds1e`	Similar to `cmds1` and `cmds1DNS` but on domain twice as large, Fig.8.
`cmds1b`	zipping–up by HPC in D of the THMMs, combined with FPC, and unzipping of the LSW snake, Figs.9 and 10 (via `cmds1bplot`).
`bdmov1`	generate “bifurcation diagram movie”; step through the BD and plot solutions and spectra/multipliers, Fig.21.
`cmds2a`	starting script near $C2TH_2$, generating Fig.11–13 (via `cmds2aplot`).
`cmds2b`	HPC and FPC of HPs and FPs from Fig.11–13, exploring recombination near $C2TH_2$, Figs.14–16.
`cmds2c`	periodic orbit continuation (POC) in D of POs from Fig.11–13, as these partly detach from the steady state branches and then form isolas, Fig.18.
`cmds2DNS`	similar to `cmds1DNS`, Fig.20.
`bdmov2`	simiar to `bdmov1` but near $C2TH_2$.
`secoinit`	initialization of problem struct `p` with standard parameter values, call of `stanpdeo1D` to generate a 1D PDE object (interval, with mesh), initialization of u with u^*, call of `oosetfemops` to generate the FEM matrices, and finally resetting of some `pde2path` parameters to problem adapted values.
`oosetfemops`	assemble and store the mass matrix M, and the (1-component) Neumann-Laplacian K.
`sG,sGjac`	rhs of (1), and Jacobian; these here have a simple standard structure.
`nodalf`	“nonlinearity”, i.e., terms without spatial derivatives, called in `hotintxs`.
`nodaljac`	Jacobian of “nonlinearity”, called in `sGjac`.
`spjac`	”spectral Jacobian”, implements $\partial_u(G_u\phi)$ for FPC, see [Uec21b, §3.6.1].
`hpjac`	”Hopf point Jacobian”, needed for HPC, again see [Uec21b, §3.6.1].
`secobra`	mod of library function `hobra`; subtract u^* from solution for branch output
`secobraHPC`	mod of `secobra` used for output during HPC.
`hotintxs`	(minor) mod of library function for simple linearly implicit DNS with stepsize δ and time–series output; system stiffness matrix K for DNS explicitly assembled here, followed by initial LU decomposition of $M + \delta K$.
`getss`	convenience function to compute steady state u^* from parameters.
`spufu`	”spectral” user function, used to plot dispersion relations.
`f1t`	”figure-1-title”; convenience function to set view and title for plots, and hence significantly shorten (plotting) scripts.