Westfälische
Wilhelms-Universität
Münster

Center for
Nonlinear Science

Münsteranian Torturials on Nonlinear Science

edited by Uwe Thiele, Oliver Kamps, Svetlana Gurevich

Continuation

SH and SHPER: Steady states of the Swift-Hohenberg equation

Uwe Thiele

with the support of

Frank Lengers

Version 1, Okt 2015

For updates of this text and the accompanying programme files see
www.uni-muenster.de/CeNoS/Lehre/Tutorials/auto.html

10 sh and shper : Steady states of the SH equation

In the tutorial SH and SHPER the dimensionless Swift-Hohenberg equation is analyzed. You will calculate homogeneous and periodic steady states. In particular, you will calculate localized periodic solutions using the boundary value mode of `auto07p` and observe their bifurcation behaviour. Additionally you will follow small amplitude modulations of the homogeneous state using the system size as a control parameter.

10.1 Model

This tutorial illustrates the calculation of steady homogeneous (part `sh`), periodic (parts `sh`, `shper`) and localized (part `shper`) solutions of the Swift-Hohenberg equation

$$\partial_t \phi = r\phi - (\partial_{xx} + q_c^2)^2\phi - \partial_\phi f(\phi) \tag{10.1}$$

studied by many authors, e.g. [1, 2, 3, 4].[4] Here we assume

$$\partial_\phi f(\phi) = -v\phi^2 + g\phi^3 \tag{10.2}$$

as in, e.g., Eq. (1) of [1, 2]. This gives $(r - q_c^4)\phi_0 - \partial_\phi f(\phi_0) = 0$ as condition for homogeneous steady states. Their linear stability is given by the dispersion relation

$$\beta(k) = -k^4 + 2q_c^2k^2 + r - q_c^4 - \partial_{\phi\phi} f(\phi_0). \tag{10.3}$$

Introducing $\beta_0 = \beta(k = 0) = r - q_c^4 - \partial_{\phi\phi} f(\phi_0)$, we obtain the critical wavenumbers that limit the band of unstable wavenumbers

$$k_\pm = \sqrt{q_c^2 \pm \sqrt{q_c^4 + \beta_0}} = \sqrt{q_c^2 \pm \sqrt{r - \partial_{\phi\phi} f(\phi_0)}}, \tag{10.4}$$

i.e., the wavenumber at onset ($r - \partial_{\phi\phi} f(\phi_0) = 0$) is $k_c = q_c$. Homogeneous states are linearly unstable for $r > \partial_{\phi\phi} f(\phi_0)$.

To study steady solutions, we set $\partial_t\phi = 0$ and obtain

$$0 = -\partial_{xxxx}\phi - 2q_c^2\partial_{xx}\phi + (r - q_c^2)\phi - \partial_\phi f(\phi) \tag{10.5}$$

that one can write as a system of first-order ordinary differential equations on the interval $\xi \equiv x/L \in [0, 1]$, where L is the physical domain size. Introducing $u_1 = \phi$, $u_2 = d\phi/dx$, $u_3 = d^2\phi/dx^2$ and $u_4 = d^3\phi/dx^3$, we obtain the autonomous system (NDIM = 4)

$$\begin{aligned} \dot{u}_1 &= Lu_2 \\ \dot{u}_2 &= Lu_3 \\ \dot{u}_3 &= Lu_4 \\ \dot{u}_4 &= L\left[-2q_c^2u_3 + (r - q_c^4)u_1 - \partial_\phi f(u_1)\right], \end{aligned} \tag{10.6}$$

where dots denote derivatives with respect to ξ. Actually, in the part `shper` u_1 is defined as $\phi - \bar{\phi}$ to facilitate the definition of a norm that is zero for all homogeneous states.

[4] Note that our notation for the nonlinearity is slightly different as we are guided by the formulation of Eq.(10.1) as gradient dynamics ($\partial_t\phi = -\frac{\delta F}{\delta\phi}$ with $F = \int_0^L \left[\frac{1}{2}(\partial_{xx}\phi)^2 - q_c^2(\partial_x\phi)^2 + \frac{q_c^4}{2}\phi^2 - \frac{r}{2}\phi^2 + f(\phi)\right] dx$ and $f = -\frac{\nu}{3}\phi^3 + \frac{g}{4}\phi^4$), i.e., our f is a local contribution to the free energy.

To study homogeneous and periodic steady states in the part `sh` we employ the modes for algebraic problems (IPS$=1$) and periodic solutions (IPS$=2$), and no further conditions need to be specified. However, the choice of runs one can do for periodic states in modus IPS$=2$ is somewhat limited and we provide a second part: `shper` that employs the boundary value mode IPS$=4$ of auto07p that allows for specification of boundary conditions and also of integral side conditions. In particular the latter can be used in a versatile way, e.g., to obtain important solutions measures as energies or self-defined norms.

In the basic set-up with IPS$=4$ used in part `shper`, we use periodic boundary conditions for u_1 to u_4 that take the form

$$u_1(0) = u_1(1), \tag{10.7}$$
$$u_2(0) = u_2(1), \tag{10.8}$$
$$u_3(0) = u_3(1), \tag{10.9}$$
$$u_4(0) = u_4(1), \tag{10.10}$$

i.e., $\mathrm{NBC} = 4$. We use an integral condition that corresponds to computational pinning (to break the translational symmetry that the solutions have for the considered homogeneous medium). That takes the form

$$\int_0^1 u_1 u_1' \, \mathrm{d}\xi = 0, \tag{10.11}$$

where u_1' is the derivative of u_1 along the solution curve in parameter space. This corresponds to imposing that the 'move' in parameter space does not contain a translation (is orthogonal to the translation eigenfunction).

In the provided code `shper` we may incorporate a second integral condition (i.e., $\mathrm{NINT} = 2$) that may, however, be removed without affecting the calculations. This integral condition

$$\int_0^1 (u_1 - \mathrm{PAR}(1)) \, \mathrm{d}\xi = 0 \tag{10.12}$$

is used to measure (as PAR(1)) the mean value of ϕ. It implies that whenever $\mathrm{NINT} = 2$ is set, PAR(1) has to be one of the continuation parameters in ICP. Similar integral conditions will be used in the tutorial `PFC`.

To follow homogeneous states in the demo `sh` we use as starting solution (provided in subroutine STPNT of file sh.f90) the trivial homogeneous state $\phi(x) = 0$, i.e., $u_1 = 0, u_2 = u_3 = u_4 = 0$, fix the domain size L at or somewhere near the wavelength at onset ($L_\mathrm{c} = 2\pi/k_c$, see below). Setting IPS$=1$ then allows one to follow the homogeneous steady states and to detect bifurcations of periodic states as Hopf bifurcation points (HB). Restarting at the HB points with IPS$=2$ switches branches and follows the periodic states.

For the part `shper` we first use numerical results of a time simulation (`shper_1.dat` and `shper_2.dat`) as starting solutions to follow different branches of localized steady states that connect to the branches we obtained in the part `sh`.
Alternatively, in the demo `shper` we follow periodic states starting with a small amplitude harmonic modulation of a homogeneous state $\phi(x) = \phi_0$. The modulation has wavelength $L_\pm = 2\pi/k_\pm$ with $i = 1, 2$ where $k_\pm$ is calculated by Eq. (10.4). Note that this even works well when choosing $L_0 = 2\pi/k_0$ at onset ($r = 0$).

For the demo `shper` (IPS$=4$), the number of free (continuation) parameters is given by NCONT $=$ NBC $+$ NINT $-$ NDIM $+1$ and is here equal to 3 (when including PAR(1), measured via the second integral condition).

Note that as in the tutorial `drop` we use an unfolding parameter ϵ as Eqs. (10.6) correspond to a conservative dynamical system. This parameter transforms the system into a 'virtual' dissipative system, thereby generating a one-parameter family of solutions. This 'virtual' dissipative system exhibits the same solution as the conservative one for $\epsilon \approx 0$ (check this!).

10.2 Runs for `sh`:

Python interface command line	**Terminal command line**
auto	
run 1 Starting with the trivial homogeneous state ($\phi = 0$) at $r = 0.4$ (for $v = 0.41$, $g = 0.1$ and $q_c = 0.5$), determine homogeneous solutions as a function of the parameter r (PAR(3)). **Continuation parameters:** r (PAR(3)) **Parameters:** IPS$=1$, ISP$=2$, ISW$=1$, ICP$=[3]$; Save output-files as `b.sh1, s.sh1, d.sh1`.	
r1=run(e='sh',c='sh.1',sv='sh1')	*@@R sh 1* *@sv sh1*
run 11 Restart at HB point of run 1; follow bifurcating periodic solution (branch switching: ISW$=-1$) in parameter r (PAR(3)). Now one needs to look at the obtained branches, plot $L2\_NORM$ over r and understand the bifurcation structure. **Continuation parameters**: r (PAR(3)), ϵ (PAR(2)); **Parameters:** IPS$=2$, ISP$=1$, ISW$=-1$, ICP$=[3,2]$; Starting from LAB4 of run 1; Save output-files as `b.sh11, s.sh11, d.sh11`	
r11=run(r1,e='sh',c='sh.11',sv='sh11')	*@@R sh 11 sh1* *@sv sh11*
clean()	*@cl*

Table 10.1: Commands for running part `sh`.

10.3 Remarks for `sh`:

- You may try `run 12` which starts the computation of the periodic branch at the other Hopf bifurcation to make sure your code works correctly.You should obtain the same branch as in `run 11`.

10.4 Tasks for `sh`:

After running the examples, you should try to implement your own adaptations, e.g.:

- Redo the runs for other values of the parameters q_c, v and g, E.g., choose values from [2].
- Redo run 11 starting at another Hopf bifurcation (HB) found in run 1.
- Restart at some periodic state obtained in run 11 and use another parameter (not r as in run 11) as principal continuation parameter.
- Replace the used $f(\phi)$ by a different one that you get from the literature or invent yourself.

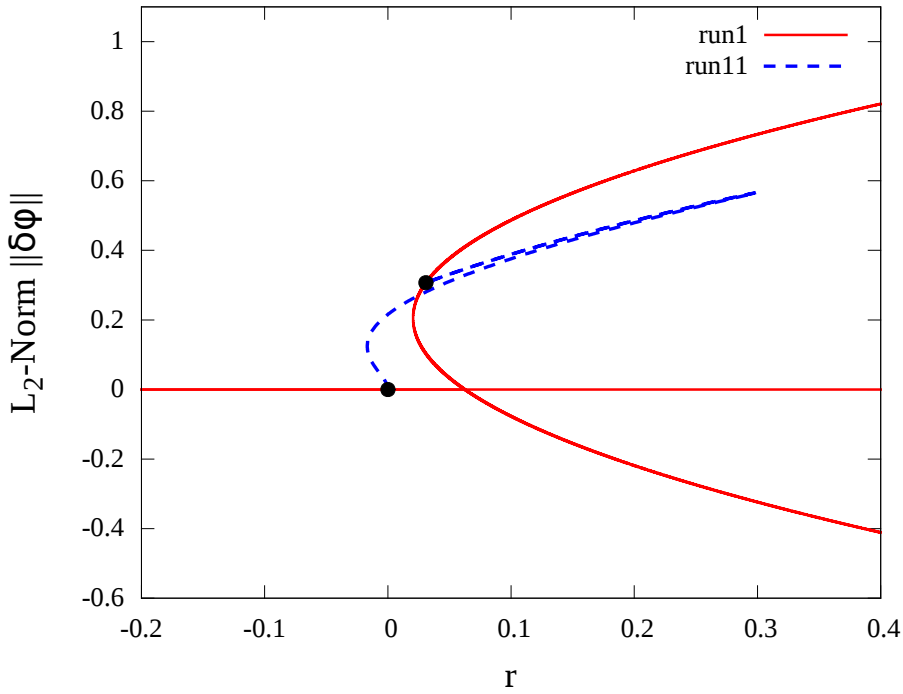

Figure 10.1: Bifurcation diagram resulting from `run1` and `run11` of part `sh` (see Table 10.1) showing the homogeneous solutions (straight lines) and the branch of periodic solutions (dashed lines) connecting two Hopf bifurcations (filled circles).

10.5 Runs for `shper`:

Python interface command line	Terminal command line
auto	
run 1: Starting with a given numerical solution of localized states obtained by time simulation (`shper_1.dat`), compute a branch of localized structures as a function of r (PAR(3)). **Continuation parameters:** r (PAR(3)), ϵ (PAR(2)); **Parameters:** IPS$=4$, ISP$=2$, ISW$=1$, ICP$=[3,2]$, NINT$=1$; Save output-files as `b.shper1, s.shper1, d.shper1`.	
r1 = run(e = 'shper', c = 'shper.1', sv = 'shper1')	*@@R shper 1* *@sv shper1*
run 11: Restart at LAB10 of run 1 (branching point BR), switch branches (ISW=-1) and follow the branch that connects the branch of the previous run to a yet unknown branch. **Continuation parameters:** r (PAR(3)), ϵ (PAR(2)); **Parameters:** IPS$=4$, ISP$=1$, ISW$=-1$, ICP$=[3,2]$, NINT$=1$; Starting from LAB10 of run1; Save output-files as `b.shper11, s.shper11, d.shper11`.	
r11 = run(r1, e = 'shper', c = 'shper.11', sv = 'shper11')	*@@R shper 11 shper41* *@sv shper11*
run 2: As in run 1, but with another starting solution (`shper_2.dat`) than in `run1`. Plot `run1`, `run11` and `run2` in one diagram, inspect the solution profiles, and understand how the solutions change along the branches. **Continuation parameters:** r (PAR(3)), ϵ (PAR(2)); **Parameters:** IPS$=4$, ISP$=1$, ISW$=1$, ICP$=[3,2]$, NINT$=1$; save output-files as `b.shper2, s.shper2, d.shper2`	
r2 = run(e = 'shper', c = 'shper.2', sv = 'shper2')	*@@R shper 2* *@sv shper 2*
run 3: Starting with a small amplitude sinusoidal solution with a period that equals the upper critical one $L_+ = 2\pi/k_+$ (change in `shper.f90` at the end of STPNT) in a domain of the same size ($L = L_+$, i.e. $n = 1$ (NN in STPNT)) for $\phi = 0$ at $r = 0.05$ (change in STPNT), determine periodic solutions as a function of the domain size L (PAR(5)). **Continuation parameters:** L (PAR(5)), ϵ (PAR(2)), $\bar{\phi}$ (PAR(1)) (set NINT=2); **Parameters:** IPS$=4$, ISP$=0$, ISW$=1$, ICP$=[5,2,1]$, NINT$=2$; Save output-files as `b.shper3, s.shper3, d.shper3`	
r3 = run(e = 'shper', c = 'shper.3', sv = 'shper3')	*@@R shper 3* *@sv shper3*
run 4: Starting with a small amplitude sinusoidal solution with a period that equals the lower critical one $L_- = 2\pi/k_-$ (change in shper.f90) in a domain of the same size ($L = L_-$,i.e. $n = 1$ (NN in STPNT)) for $\phi = 0$ at $r = 0.05$ ($v = 0.41$, $g = 1$, $q_c = 0.5$), determine periodic solutions as a function of the domain size L (PAR(5)). **Continuation parameters:** L (PAR(5)), ϵ (PAR(2)) and $\bar{\phi}$ (PAR(1)) **Parameters:** IPS$=4$, ISP$=0$, ISW$=1$, ICP$=[5,2,1]$; NINT$=2$, save output-files as `b.shper4, s.shper4, d.shper4`	

r4=run(e='shper',c='shper.4',sv='shper4')	*@@R shper 4* *@sv shper4*
run5: Starting with a small amplitude sinusoidal solution with a period that equals the upper critical one $L_+ = 2\pi/k_+$ (change again in `shper.f90`) in a domain of size $L = 6L_+$, i.e. $n = 6$ (NN in STPNT in `shper.f90`) for $\phi = 0$ at $r = 0.05$ (for $v = 0.41$, $g = 1.0$, $q_c = 0.5$), determine periodic solutions as a function of the domain size L (PAR(5)). **Continuation parameters:** L (PAR(5)), ϵ (PAR(2)), $\bar{\phi}$ (PAR(1)); **Parameters:** IPS$= 4$, ISP$= 0$, ISW$= 1$, ICP$= [5, 2, 1]$, NINT$= 2$; save output-files as `b.shper5, s.shper5, d.shper5`	
r5=run(e='shper',c='shper.5',sv='shper5')	*@@R shper 5* *@sv shper5*
clean()	*@cl*

Table 10.2: Commands for running part `shper`.

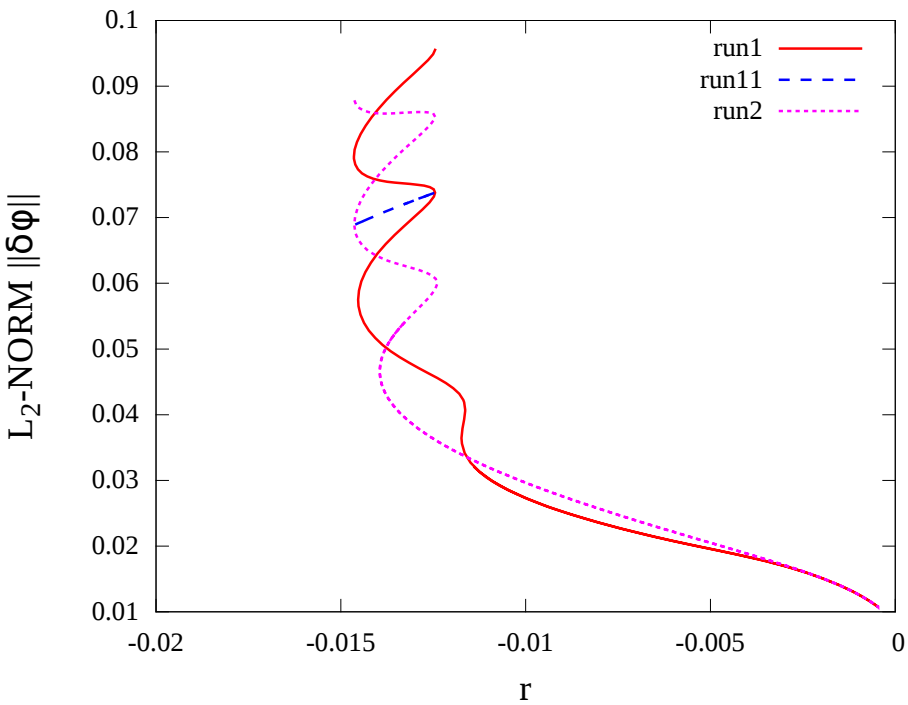

Figure 10.2: Bifurcation diagram showing the so called snaking of localized states from `run1` (solid lines), `run2` (dotted lines) and `run11` (dashed lines) of part `shper` (see Table 10.2) connecting two branching points

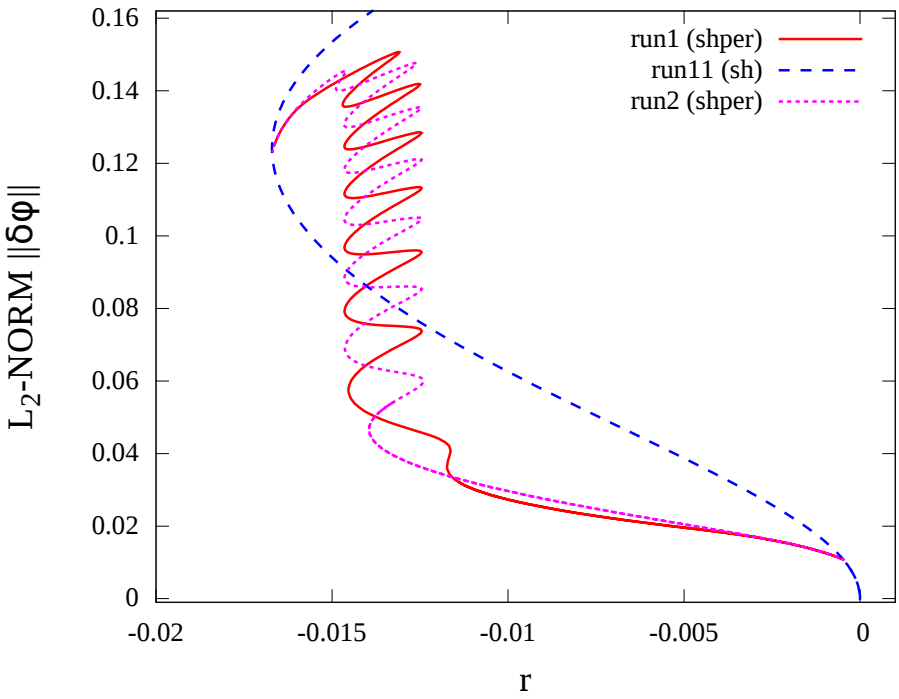

Figure 10.3: Bifurcation diagram showing the complete snaking connecting the branch of periodic steady solutions obtained in `run11` of part `sh` (dashed lines) and two branches of localized solutions obtained in `run1` (solid lines) and `run2` (dotted lines) of part `shper`

10.6 Remarks for `shper`:

- The starting solutions for localized structures were generated by using time simulation. The respective python code is given by `shper_start.py`. You may run the program by typing `python shper_start.py`. Wheather you end up at the starting solution for the branch of `run1` or `run2` is determined by the choice of initial conditions in line 53 and line 55. The name of the output file (initially `shper_1.dat`) is defined in line 116.

10.7 Tasks for `shper`:

After running the examples, you should try to implement your own adaptations, e.g.:

- Extent the runs for localized structures (`shper1, shper2`) so that the snaking connects to the branch of periodic solutions (`run11` of part `sh`). `NMX=330` for `c.shper.1` and `NMX=380` for `c.shper.2` should be sufficient for the same `DSMAX`.
- You may find other localized structures as starting points using the python-code `shper_start.py` by varying the parameters and start from these new starting solutions.
- Redo the runs for other values of the parameters q_c, v and g, E.g., choose values from [2].
- Start as in run 3 but another parameter (not L as in run 3) as principal continuation parameter.
- Deactivate the integral condition that measures the mean value of ϕ.
- Replace the used $f(\phi)$ by a different one that you get from the literature (e.g from [3]) or invent yourself.

References

[1] J. Burke and E. Knobloch. "Localized states in the generalized Swift-Hohenberg equation". In: *Phys. Rev. E* 73 (2006), p. 056211. DOI: 10.1103/PhysRevE.73.056211.

[2] A. Bergeon et al. "Eckhaus instability and homoclinic snaking". In: *Phys. Rev. E* 78 (2008), p. 046201. DOI: 10.1103/PhysRevE.78.046201.

[3] S. M. Houghton and E. Knobloch. "Swift-Hohenberg equation with broken cubic-quintic nonlinearity". In: *Phys. Rev. E* 84 (2011), p. 016204. DOI: 10.1103/PhysRevE.84.016204.

[4] D. J. B. Lloyd et al. "Localized Hexagon Patterns of the Planar Swift–Hohenberg Equation". In: *SIAM J. Appl. Dyn. Syst., 7(3), 1049–1100* 7(3) (2008), pp. 1049–1100. DOI: 10.1137/070707622.