446-2 Numerical Solution of Partial Differential Equations
Spring 2007
Hermann Riecke

Problem Set 3, April 20, 2007 due April 28, 2007

1 The 2-Dimensional Swift-Hohenberg Modell

In this homework you implement a pseudo-spectral code to solve the Swift-Hohenberg model in a two-dimensional square domain with periodic boundary condition,

$$\partial_t \psi = R\psi - \left(d\nabla^2 + 1\right)^2 \psi + \beta\psi^2 - \psi^3 \qquad 0 \le x \le L \quad 0 \le y \le L \tag{1}$$

This equation models a wide range of systems undergoing a steady bifurcation to patterns with a wavenumber close to $q_c = 1$. A classic example is convection of a thin layer of fluid heated from below for which you can think of the real quantity ψ as representing the temperature of the fluid in the mid plane.
First transform the system to the interval $[0, 2\pi]$, which amounts to a rescaling of the real coefficient d. Evaluate the derivatives using FFTs, remembering that in Fourier space $\nabla^2\psi = \sum_{kl}(-k^2 - l^2)\psi_{kl}e^{i(kx+ly)}$. For the time stepping use again the Crank-Nickolson/Adams-Bashforth scheme (CNAB). As discussed in class, to modify your previous CGL-code you need essentially only to define the $N \times N$-matrix for the derivative operator $\left(d\nabla^2 + 1\right)^2$ and modify the initial condition.
To plot a movie of the evolution of the two-dimensional solution $\psi(x, y, t)$ insert the sequence of commands at the end of the assignment after the call to CNAB (use Matlab's help to find out what the commands do if you are not familiar with them).

2 Testing of the Code

In this case there is no exact, non-trivial solution for the full equation. Therefore you want to test the critical, new derivative part by itself carefully. To do so compare the numerical solution with the exact analytical solutions of

$$\partial_t \psi = -\left(d\nabla^2 + 1\right)^2 \psi \qquad \text{with } d = 1$$

in a box of size $L = 20$ for the initial conditions

$$\begin{aligned}
\psi_1 &= \cos(qx) \\
\psi_2 &= \cos(2qx + qy) \\
\psi_3 &= \sin(3qy)
\end{aligned}$$

with $q = 2\pi/L$. For this test it is sufficient to test the growth or decay rate of the numerical solution at a single grid point. Since you will not perform a detailed convergence test for this assignment it is important that you check that the agreement with the exact solutions is good.

3 Evolution of Two-Dimensional Patterns

Use now your code to explore some properties of the Swift-Hohenberg model. It captures a number of phenomena that are observed in a variety of experiments, e.g. in Rayleigh-Benard convection or block co-polymers in thin films. For further information have a look at the following papers[1]

1. Bodenschatz *et al.*, *Transitions between Patterns in Thermal Convection,* Phys. Rev. Lett. 67 (1991) 3078.
2. Harrison *et al.*, *Mechanisms of Ordering in Striped Patterns*, Science 290 (2000) 1558.
3. Ouyang and Swinney, *Transition from a uniform state to hexagonal and striped Turing patterns,* Nature 352 (1991) 610

Do the following investigations:

1. For $\beta = 0$ start from random initial conditions with magnitude 0.05. Use $L = 40$ and at least $nx = 128$. What kind of final states do you obtain for $R = -0.2$, $R = 0.3$, $R = 0.9$, $R = 2$? How large do you have to choose t_{max} to reach a stationary state? How do the patterns change when you increase or decrease the magnitude of the random initial conditions by an order of magnitude? As always, **make sure** your observations and conclusions are not due to insufficient temporal or spatial resolution.
2. Investigate the behavior of the system for $\beta = 1$ using the same parameter values as for $\beta = 0$.

4 Appendix: Two-dimensional Movie

```
% Plot every ntplot steps
if (rem(nt,ntplot) == 1)
if (nt==1)
fig1=figure(1);
HandleM=surf(y);
shading interp
view(2)
axis([0 nx 0 nx]);
caxis([-1.5 1.5])
colorbar
hold on
set(gcf,'DoubleBuffer','on');
xlabel(' x ')
```

[1] If you cannot find the papers directly, there are links to them on the class web site.

```
ylabel(' y ')
else
set(HandleM,'ZData',y);
drawnow;
end
end
```