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

Problem Set 3, April 28, 2007 due May 12, 2007

1 Exponential Time Differencing

In this homework you implement the exponential time differencing scheme evaluating the integral with 4^{th}-order Runge-Kutta (ETDRK4). The scheme is given in detail in the class notes.
Write a matlab function ETDRK4 that solves a general PDE of the form

$$\partial_t \psi = L\psi + N(\psi)$$

with L being a linear operator containing spatial derivatives and possibly a term without a derivative and N representing a nonlinear term without any derivatives. Write a separate function that computes the factors E_i of the ETDRK4 scheme. This function depends only on $c = dt \cdot \lambda$ with λ being the list of eigenvalues of L. Note that for small c the expressions for $E_i,\ i = 2 \ldots 5$ are very susceptible to round-off error due to cancelations. For those values of c you need to replace these expression with their Taylor expansions. **Without any modification,** your two functions implementing ETDRK4 should work in one, two, or arbitrary spatial dimensions. To do this efficiently use the FIND function of matlab. A typical use of FIND is, for instance,

index=find(A<1);

A(index)=0;

In a matrix of arbitrary dimension these two commands set all those entries to 0 that are smaller than 1.

2 Nonlinear Schrödinger Equation

Insert your ETDRK4 function into your standard code. Define λ, $N(\psi)$, and the initial conditions appropriately to solve the complex Ginzburg-Landau equation

$$\partial_t A = d\partial_x^2 A + aA + c|A|^2 A \qquad 0 \le x \le L. \tag{1}$$

Using the same parameter values and the same soliton as in homework 2 to perform a detailed convergence test of your ETDRK4. Now, however, use $L = 160$ and $t_{max} = 10$.

1. Do you obtain the correct order of accuracy with your code?

2. Can you now achieve an accuracy that is only limited by round-off error?

3. Provide a single figure showing the error as a function of Δt for all values of N that you used to demonstrate your conclusion.

Note: If you want to see a few more interesting solutions and regimes of the complex Ginzburg-Landau equations have a look at the papers

1. H. Chate, *Spatiotemporal intermittency regimes of the one-dimensional complex Ginzburg-Landau equation*, Nonlinearity 7 (1994) 185

2. I. Aranson and L. Kramer, *The world of the complex Ginzburg-Landau equation,* Rev. Mod. Phys. 74 (2002) 99.

There are links to these two papers on the class web site.

3 Swift-Hohenberg Model

Now change λ, $N(\psi)$, and the initial condition so as to solve the two-dimensional Swift-Hohenberg model as in homework 3 ($\beta = 0$),

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

Test the convergence of the ETDRK4 scheme now in a regime in which the evolution of the solution is very slow and study its behavior for relatively large time steps.

1. Use $L = 30$ and $R = 0.8$ and limit yourself to $N = 64$. First prepare a suitable initial condition for your convergence test by starting from random initial conditions and obtain a solution for $t_{max}^{(0)} = 400$. Save that solution ψ_{ic} to a file and use it as initial condition for the further tests.

2. Using the initial condition ψ_{ic} perform a convergence test by running the code to $t_{max} = 300$. Since you do not have an exact solution assess the accuracy of the code by measuring the difference $\Delta\psi(x_i, y_j)$ between two final states obtained with time steps Δt differing by a factor of 2. Plot $N^{-2}\sum_{ij}|\Delta\psi(x_i, y_j)|$ as a function of Δt. Start with the largest Δt allowed by the stability of the code and decrease Δt from there by factors of 2. For this homework it will be sufficient to plot 4 or 5 such data points.

3. Now perform the same convergence test with your CNAB code, i.e. start from the very same ψ_{ic} and run the code again to $t_{max} = 300$. Again start from the largest Δt that the stability of your code allows. Decrease Δt by factors of 2 until you reach an accuracy that is comparable to that obtained with your ETDRK4 code. Include these data in the same convergence plot for ETDRK4.