Numerical Programming 2 (CSE) 2015

Worksheet 11

Exercise 1 (Fourier Splitting)

We consider the Schrödinger equation with Gaussian initial condition

$$\begin{cases} i\dfrac{\partial}{\partial t}\psi(x,t) = \left(-\dfrac{1}{2}\dfrac{\partial}{\partial x^2} + (x-5)^m\right)\psi(x,t) \\ \psi(x,0) = \dfrac{1}{\pi^{1/4}}e^{-(x-x_0)^2} \end{cases}$$

on $x \in [0, 10]$. Remember (cf. Worksheet 6) that for a function $f(x) = \sum_{k=-N}^{N} a_k e^{2ik\pi x}$, $x \in [0,1]$, the coefficients a_k can be computed from the vector v, $v_j := f\left(\frac{j-1}{2N+1}\right)$, $j = 1, \ldots, 2N+1$ and vice versa using the MATLAB functions `fft` resp. `ifft`, where

$$\texttt{fft}(v) = \frac{1}{2N+1}\mathcal{F}_N v = \frac{1}{2N+1}\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \\ a_{-N} \\ \vdots \\ a_{-1} \end{pmatrix}$$

and

$$\texttt{ifft}(\texttt{fft}(v)) = v.$$

- Transform the above Schrödinger equation to an equation $i\widetilde{\psi}_t = \widetilde{H}\widetilde{\psi}$ on the interval $[0, 1]$ (this interval is the most conenient one for use with MATLAB's FFT).
- Implement the Fourier split-step algorithm for the transformed Schrödinger equation, i.e. for $\widetilde{H} = -c\Delta + \widetilde{V}$ and $u_j = \widetilde{\psi}(x_j)$, $\frac{j}{2N+1}$, $j = 1, \ldots, 2N+1$ compute in each step
  1. $u_j = e^{-i\tau/2\widetilde{V}(x_j)}u_j \ \forall j$
  2. $a = \mathcal{F}_N u$, $a_k = e^{-i\tau c(2\pi k)^2}a_k \ \forall k$, $u = \mathcal{F}_N^{-1}a$
  3. $u_j = e^{-i\tau/2\widetilde{V}(x_j)}u_j \ \forall j$
- Use the algorithm to solve the transformed Schrödinger equation with time step $\tau = 0.01$ and initial position $x_0 = 3$. Try both $m = 2$ and $m = 4$ as exponents in the potential. Visualize the evolution of the probability density $|\psi(\cdot, t)|^2$.

Exercise 2 (Crank-Nicolson)

The Crank-Nicolson method for the advection equation

$$\partial_t u + \partial_x (f(u)) = 0, \qquad f(u) = cu$$

uses cells of uniform size Δx with approximate cell averages

$$u_j^n \approx \frac{1}{\Delta x} \int_{x_j-1/2}^{x_j+1/2} u(x, t_n) dx$$

and the flux approximation

$$f(u(x_{j+1/2})) := v_{j+1/2} \approx \frac{1}{2} (f(u_j) + f(u_{j+1}))$$

$$\Rightarrow \partial_t u_j = \frac{1}{\Delta x} \left(v_{j-1/2} - v_{j+1/2} \right) \approx \frac{c}{2\Delta x} (u_{j-1} - u_{j+1})$$

together with the trapezoidal rule in time

$$u_j^{n+1} - u_j^n \approx \frac{\Delta t}{2} (\partial_t u_j(t_{n+1}) + \partial_t u_j(t_n)), \qquad t_{n+1} - t_n = \Delta t.$$

- Formulate the equation relating the vectors u^{n+1} and u^n in matrix form.
- Determine the region of numerical dependency. Does the CFL-condition impose a step size restriction?
- Use the Crank-Nicolson method with $\Delta t = \Delta x$ to compute approximate solutions $\widetilde{u}$ of the advection equation on $x \in [0, 10]$, $t \in [0, 10]$ with $c = 1$, initial condition $u(x, 0) = 0$, the condition $u_{M+1}(t) = 0$ on the right boundary (where cell $M + 1$ contains $x = 10$) and two different conditions on the left boundary:
  - $u_0(t) = \sin(t)$
  - $u_0(t) = (1 - e^{-t/3}) \sin(t)^2$

  Plot the error $\|u(\cdot, 10) - \widetilde{u}(\cdot, 10)\|_{L^2}$ as $\Delta t = \Delta x$ varies for both cases. (The exact solution is $u(x, 10) = u(0, 10 - x)$.)