Solutions to Worksheet 11

Exercise 1:
We consider the Schrödinger equation

$$\begin{aligned} i\frac{\partial}{\partial t}\psi(x,t) &= (-\frac{1}{2}\frac{\partial^2}{\partial x^2} + (x-5)^m)\psi(x,t), \\ \psi(x,0) &= (\pi)^{-\frac{1}{4}} e^{-(x-x_0)^2} \end{aligned} \tag{0.1}$$

where $\psi : [0;10] \times \mathbb{R} \mapsto \mathbb{R}$ is defined for $x \in [0;10]$. To use Matlab's FFT method, we first have to rewrite this equation on the interval $[0;1]$. Thus, we introduce a new function

$$\tilde{\psi} : [0;1] \times \mathbb{R} \mapsto \mathbb{R} \quad \text{such that } \tilde{\psi}(x,t) = \psi(10x,t).$$

Let $x \in [0;1]$ and $y = 10x \in [0;10]$. With these definitions we find

- $\tilde{\psi}(x,0) = \psi(y,0) = (\pi)^{-\frac{1}{4}} e^{-(y-x_0)^2} = (\pi)^{-\frac{1}{4}} e^{-100(x-\frac{x_0}{10})^2}$,
- $\frac{\partial}{\partial t}\tilde{\psi}(x,t) = \frac{\partial}{\partial t}\psi(y,t)$,
- $\frac{\partial^2}{\partial x^2}\tilde{\psi}(x,t) = \frac{\partial^2}{\partial x^2}(\tilde{\psi}(y,t)) = \frac{\partial}{\partial x}(10\frac{\partial}{\partial y}\tilde{\psi}(y,t)) = 100\frac{\partial^2}{\partial y^2}\psi(y,0)$.

Equation (0.1) is valid for all $y \in [0;10]$, so we can rewrite (0.1) as

$$i\frac{\partial}{\partial t}\psi(10x,t) = -\frac{1}{2}\frac{\partial^2}{\partial x^2}(\psi(10x,t)) + (10x-5)^m\psi(10x,t).$$

for $x \in [0;1]$. This means, $\tilde{\psi}$ is the solution to

$$\begin{aligned} i\frac{\partial}{\partial t}\tilde{\psi} &= (-\frac{1}{200}\frac{\partial^2}{\partial x^2} + (10x-5)^m)\tilde{\psi}(x,t), \\ \tilde{\psi}(x,t) &= (\pi)^{-\frac{1}{4}} e^{-100(x-\frac{x_0}{10})^2}. \end{aligned} \tag{0.2}$$

Exercise 2:

We approximate the solution of the advection equation

$$\partial_t u + \partial_x f(u) = 0 \tag{0.3}$$

with boundary conditions $u(0,t) = u_0(t)$, $u(x_f,t) = u_f(t)$ and $f(u) = cu$ using the Crank-Nicolson method. For the approximated solution we write u_j^n. We discretize the position variable x with uniform cells of size Δx and denote the nodes with $x_0, x_1, \ldots, x_M, x_{M+1}$. We approximate the cell average via

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

for all inner nodes $u_1, \ldots, u_M$. For the nodes u_0 resp. u_{M+1} the boundary conditions impose $u_0^n = u_0(t_n)$ and $u_{M+1}^n = u_f(t_n)$.
Moreover, we also use a uniform time step size $t_{n+1} - t_n = \Delta t$. According to the trapezoidal rule, we define the approximated value for the next time step via

$$u_j^{n+1} = u_j^n + \Delta t\frac{1}{2}(\partial_t u_j^{n+1} + \partial_t u_j^n) \quad \text{for } j = 1, \ldots, M. \tag{0.4}$$

To estimate the time derivative, we use that $\partial_t u = -\partial_x f(u)$. Hence,

$$\partial_t u_j^n = -\frac{f(u_{j+1}^n) - f(u_{j-1}^n)}{2\Delta x} = \frac{c}{2\Delta x}(u_{j-1}^n - u_{j+1}^n) \quad \text{for } j = 1, \dots, M$$

since $f(u) = cu$. Inserting this into Equation (0.4) yields

$$\begin{aligned} u_j^{n+1} &= u_j^n + \Delta t \frac{1}{2} \cdot \frac{c}{2\Delta x}(u_{j-1}^{n+1} - u_{j+1}^{n+1} + u_{j-1}^n - u_{j+1}^n) \\ &= u_j^n + \frac{\Delta t}{\Delta x} \cdot \frac{c}{4}(u_{j-1}^{n+1} - u_{j+1}^{n+1} + u_{j-1}^n - u_{j+1}^n) \end{aligned}$$

for $j = 1, \dots, M$. For simplification we denote $k = \frac{\Delta t}{\Delta x} \cdot \frac{c}{4}$. Then, the matrix representation of the above equation reads

$$u^{n+1} = u^n + k \begin{pmatrix} 0 & -1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & 1 & 0 \end{pmatrix} (u^{n+1} + u^n) + k \begin{pmatrix} u_0(t_{n+1}) + u_0(t_n) \\ 0 \\ \vdots \\ 0 \\ -u_f(t_{n+1}) - u_f(t_n) \end{pmatrix}.$$

We can simplify this equation further to

$$\begin{pmatrix} 1 & k & & 0 \\ -k & \ddots & \ddots & \\ & \ddots & \ddots & k \\ 0 & & -k & 1 \end{pmatrix} u^{n+1} = \begin{pmatrix} 1 & -k & & 0 \\ k & \ddots & \ddots & \\ & \ddots & \ddots & -k \\ 0 & & k & 1 \end{pmatrix} u^n + k \begin{pmatrix} u_0(t_{n+1}) + u_0(t_n) \\ 0 \\ \vdots \\ 0 \\ -u_f(t_{n+1}) - u_f(t_n) \end{pmatrix}.$$

To deduce u^{n+1} we have to multiply this equation with the inverse of

$$\begin{pmatrix} 1 & k & & 0 \\ -k & \ddots & \ddots & \\ & \ddots & \ddots & k \\ 0 & & -k & 1 \end{pmatrix}.$$

In general, this matrix will be dense, take for example the two-dimensional example with $k = 1$, then

$$\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}^{-1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

Thus, u_j^{n+1} will (again, in general) depend on all u_i^n, $i = 1, ..., M$ and the region of numerical dependency is the whole grid. Furthermore, due to the CFL-condition no step size restriction is required.