Numerical methods
Bachelor in Physics

Practicum 10: Partial differential equations IV: Implicit Methods

Contact: jeroen.mulkers@uantwerpen.be U.305

In practical 7, we used the FTCS to solve numerically the diffusion equation (a parabolic PDE). The FTCS is an explicit method. In this practical, we will study the time-dependent Schrödinger equation (also a parabolic PDE) using an implicit method.

1 The Schrödinger equation

The one-dimensional time-dependent Schrödinger equation is given by

$$i\hbar\frac{\partial\psi}{\partial t} = H\psi \tag{1}$$

with

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x) \tag{2}$$

The formal solution is

$$\psi(x,t) = \exp\left[-\frac{i}{\hbar}Ht\right]\psi(x,0) \tag{3}$$

Applying the FTCS on (1) yields

$$i\hbar\frac{\psi_j^{n+1} - \psi_j^n}{\tau} = -\frac{\hbar^2}{2m}\frac{\psi_{j+1}^n + \psi_{j-1}^n - 2\psi_j^n}{h^2} + V_j\psi_j^n$$

with $\psi_j^n \equiv \psi(x_j, t_n)$ and $V_j \equiv V(x_j)$. Because the Hamiltonian is a linear operator, we can rewrite previous equation:

$$i\hbar\frac{\psi_j^{n+1} - \psi_j^n}{\tau} = \sum_{k=1}^{N} H_{jk}\psi_k^n \tag{4}$$

with matrix $\mathbf{H}$ as the discrete form of the Hamiltonian:

$$H_{jk} = -\frac{\hbar^2}{2m}\frac{\delta_{j+1,k} + \delta_{j-1,k} - 2\delta_{j,k}}{h^2} + V_j\delta_{j,k}$$

Solving expression (4) for ψ_j^{n+1} yields a numerical scheme, which can be written down in matrix notation as

$$\Psi^{n+1} = \left(\mathbf{I} - \frac{i\tau}{\hbar}\mathbf{H}\right)\Psi^n \tag{5}$$

with column vector Ψ^n and the unity matrix $\mathbf{I}$. This is an explicit FTCS for the time-dependent Schrödinger equation. Since $e^{-z} \approx 1 - z$, we can interpret (5) as the first term in the Taylor expansion of (3), in order to compute the solution at the next time step.

2 Implicit schemes

The disadvantage of explicit schemes is the instability of the solution for large time steps. We will present an alternative method. Let's apply the Hamiltonian on the future values of ψ:

$$i\hbar\frac{\psi_j^{n+1} - \psi_j^n}{\tau} = \sum_{k=1}^{N} H_{jk}\psi_k^{n+1}$$

or

$$\Psi^{n+1} = \Psi^n - \frac{i\tau}{\hbar}\mathbf{H}\Psi^{n+1}$$

Solving for Ψ^{n+1} yields

$$\Psi^{n+1} = \left(\mathbf{I} + \frac{i\tau}{\hbar}\mathbf{H}\right)^{-1}\Psi^n \tag{6}$$

This is the *implicit FTCS method.* Both schemes are equivalent in the limit $\tau \to 0$ because (6) is an alternative way to represent the first term of the Taylor expansion of (3). ($e^{-z} = 1/e^z \approx (1+z)^{-1}$)

When using this new method, one has to compute the inverse of a matrix. This computation is always needed when using implicit schemes. However, calculating the inverse of the matrix pays off since every implicit scheme is stable.

Besides being stable, a numerical scheme has to be accurate. Remember, a solution which does not 'blow up', is not necessarily accurate. A more accurate scheme is the *Crank-Nicolson method.* This method uses the average of the explicit and the implicit FTCS scheme:

$$i\hbar\frac{\psi_j^{n+1} - \psi_j^n}{\tau} = \frac{1}{2}\sum_{k=1}^{N} H_{jk}(\psi_k^n + \psi_k^{n+1})$$

Or in matrix notation:

$$\Psi^{n+1} = \Psi^n - \frac{i\tau}{2\hbar}\mathbf{H}(\Psi^n + \Psi^{n+1})$$

Solving for Ψ^{n+1} yields

$$\Psi^{n+1} = \left(\mathbf{I} + \frac{i\tau}{2\hbar}\mathbf{H}\right)^{-1}\left(\mathbf{I} - \frac{i\tau}{2\hbar}\mathbf{H}\right)\Psi^n \tag{7}$$

A Padé-approximant for the exponential function is

$$e^{-z} \approx \frac{1 - z/2}{1 + z/2}.$$

Based on the expression of this approximant, we can interpret (7) as an alternative form to approximate the formal solution (3). Notice that for an imaginary z, $(e^{-z})^*$ is equal to the invers of e^{-z}, which means that e^{-z} is unitary. The operator $\exp[-iHt/\hbar]$ in (3) is also unitary. Out of the three given approximations of the exponential function, $1-z$,$1/(1+z)$ and $(2-z)/(2+z)$, only the Padé-approximant preserves the unitary property.

3 Wave packet of a free particle

Before studying the implementation of the Crank-Nicolson scheme for the time-dependent Schrödinger equation, we have to choose an initial wave function. We will use a normalized Gaussian wave packet as the initial wave function

$$\psi(x, t = 0) = \frac{1}{\sqrt{\sigma_0\sqrt{\pi}}} e^{ik_0x} e^{-(x-x_0)^2/2\sigma_0^2}$$

centered around x_0, width σ_0, and momentum $p_0 = \hbar k_0$. A Gaussian wave packet has the remarkable property that the product $\Delta x \Delta p$ is equal to the theoretical minimal value $\hbar/2$.

For a free particle ($V(x) = 0$), the wave function evolves over time as

$$\psi(x, t) = \frac{1}{\sqrt{\sigma_0\sqrt{\pi}}} \frac{\sigma_0}{\alpha} e^{ik_0(x - p_0t/m)} e^{-(x-x_0-p_0t/m)^2/2\alpha^2}$$

with $\alpha^2 = \sigma_0^2 + i\hbar t/m$. The probability density $P(x, t) = |\psi(x, t)|^2$ is equal to

$$P(x, t) = \frac{\sigma_0}{|\alpha|^2\sqrt{\pi}} \exp\left[-\left(\frac{\sigma_0}{|\alpha|}\right)^4 \frac{(x - x_0 - p_0t/m)^2}{\sigma_0^2}\right]$$

and stays a Gaussian function. The peak translates as $\langle x \rangle = \int_{-\infty}^{\infty} xP(x, t)dx = x_0 + p_0t/m$. The width increases over time as

$$\sigma(t) = \sigma_0\sqrt{\left(\frac{|\alpha|}{\sigma_0}\right)^4} = \sigma_0\sqrt{1 + \frac{\hbar^2t^2}{m^2\sigma_0^4}}$$

4 The program

The program schro.m solves the time-dependent Schrödinger equation, using the Crank-Nicolson scheme. The algorithm is as follows:

- Initialise the parameters $(h, \tau, \ldots)$
- Implement the matrix $\mathbf{H}$.
- Calculate the Crank-Nicolson matrix $(\mathbf{I} + \frac{i\tau}{2\hbar}\mathbf{H})^{-1}(\mathbf{I} - \frac{i\tau}{2\hbar}\mathbf{H})$.
- Choose the initial wave function.
- Plot the imaginary and real part of the initial wave function.
- Set the loop and plot variables.
- Iterate until the center of the wave packet is again at its starting position.
  - Calculate the new wave function using the Crank-Nicolson matrix.
  - Save data regularly (for the visualisation later)
- Plot the probability density $P(x,t)$ in function of the time.

The first and the last row of matrix $\mathbf{H}$ need special attention. If we choose to use periodic boundary conditions, we get:

$$\begin{aligned} H_{1,k} &= -\frac{\hbar^2}{2m}\frac{\delta_{2,k} + \delta_{N,k} - 2\delta_{1,k}}{h^2} \\ H_{N,k} &= -\frac{\hbar^2}{2m}\frac{\delta_{1,k} + \delta_{N-1,k} - 2\delta_{N,k}}{h^2} \end{aligned}$$

5 Tasks

1. Study the program schro.m. In order to get good results, you need a large number of grid points. However, this can lead to very large matrices and memory related problems. Furthermore, the computation time needed to invert a $N \times N$ matrix, scales as N^3. One can avoid these problems by taking into account the *sparsity* of the matrix $\mathbf{H}$. The computation of the inverse matrix can be avoided by rewriting the Crank-Nicolson scheme as

follows:

$$\begin{aligned}
\Psi^{n+1} &= \left(\mathbf{I} + \frac{i\tau}{2\hbar}\mathbf{H}\right)^{-1}\left(\mathbf{I} - \frac{i\tau}{2\hbar}\mathbf{H}\right)\Psi^n \\
&= \left(\mathbf{I} + \frac{i\tau}{2\hbar}\mathbf{H}\right)^{-1}\left[2\mathbf{I} - \left(\mathbf{I} + \frac{i\tau}{2\hbar}\mathbf{H}\right)\right]\Psi^n \\
&= \left[2\left(\mathbf{I} + \frac{i\tau}{2\hbar}\mathbf{H}\right)^{-1} - \mathbf{I}\right]\Psi^n \\
&= (\mathbf{Q}^{-1} - \mathbf{I})\Psi^n \\
&= \mathbf{Q}^{-1}\Psi^n - \Psi^n
\end{aligned}$$

with $\mathbf{Q} = \frac{1}{2}[\mathbf{I} + (i\tau/2\hbar)\mathbf{H}]$. The computation of the inverse matrix can now be avoided by, first, solving the system

$$\mathbf{Q}\chi = \Psi^n$$

and then, updating the solution:

$$\Psi^{n+1} = \chi - \Psi^n$$

We no longer have to compute the inverse of a matrix. Instead, every time step, we have to solve a system of eequations. This method does not speed op the computation, unless we consider the matrix $\mathbf{Q}$ as a sparse matrix.

Implement this method (taking into account the sparsity of the matrix $\mathbf{Q}$) and compare the computation time for $\tau = 0.1$ and $N = 1000$. Repeat for $N = 2000$.

2. Change the periodic boundary conditions to the Dirichlet boundary conditions $\psi(x = \pm L/2, t) = 0$. Note that this corresponds to the typical 'particle in a box' problem.

3. Change the initial wave packet to
   - the ground state of an infinite potential well
   - the sum of the ground state and the first excited state of the infinite well

4. Use the original program with periodic boundary conditions and add the potential $V(x) = U\delta(x - L/2)$. What happens with the result for different values of U (larger and smaller than the energy E of the particle).

5. Discuss the scattering of a particle at a potential barrier.