Computational Physics – Lecture 16: Diffusion equation II

Kristel Michielsen
Institute for Advanced Simulation
Jülich Supercomputing Centre
Research Centre Jülich
k.michielsen@fz-juelich.de
http://www.fz-juelich.de/ias/jsc/qip

Contents

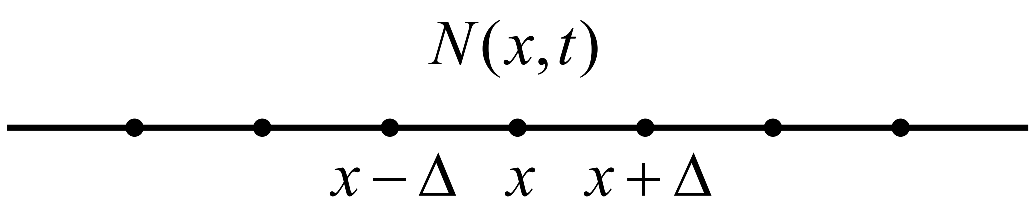
- How to solve the diffusion equation numerically
  - Random walk and diffusion equation
  - FTCS and Crank-Nicolson method
  - Diffusion equation in matrix notation
  - Product formula approach
- Exercise (Not for students who get < 10 ECTS for the course)

How to solve the diffusion equation numerically?

Random walk and diffusion

- We start by looking at a random walk of particles on a line
- Time proceeds in steps τ
- At each step, with probability ½, a particle jumps to the left or to the right by a distance Δ
- The number of particles at position x changes according to

$$N(x,t+\tau)=\frac{1}{2}N(x+\Delta,t)+\frac{1}{2}N(x-\Delta,t)$$



Random walk and diffusion

- Subtract $N(x,t)$ from both sides, divide both sides by τ and rearrange a little to obtain

$$\frac{N(x,t+\tau)-N(x,t)}{\tau}=\frac{\Delta^2}{2\tau}\frac{N(x+\Delta,t)-2N(x,t)+N(x-\Delta,t)}{\Delta^2}$$

- Next, we assume that the time step τ and the distance Δ are "small" and that $N(x,t)$ is sufficiently "smooth" such that its first and second derivatives exists
- Then, we may use

$$\frac{N(x,t+\tau)-N(x,t)}{\tau}\approx\frac{\partial}{\partial t}N(x,t)$$

$$\frac{N(x+\Delta,t)-2N(x,t)+N(x-\Delta,t)}{\Delta^2}\approx\frac{\partial^2}{\partial x^2}N(x,t)$$

Random walk and diffusion

- We find the diffusion equation (in 1D)

$$\frac{\partial N(x,t)}{\partial t} = \frac{\Delta^2}{2\tau}\frac{\partial^2 N(x,t)}{\partial x^2} \equiv D\frac{\partial^2 N(x,t)}{\partial x^2}$$

- Note on the continuum limit $\tau, \Delta \to 0$
  - The identification $D \Leftrightarrow \Delta^2 / 2\tau$ only makes mathematical sense if

$$0 < \lim_{\tau\to 0}\lim_{\Delta\to 0}\frac{\Delta^2}{2\tau} = \lim_{\Delta\to 0}\lim_{\tau\to 0}\frac{\Delta^2}{2\tau} < \infty$$

  - The discrete model is always well-defined

From continuum model to discrete model

- Diffusion equation

$$\frac{\partial N(x,t)}{\partial t} = D\frac{\partial^2 N(x,t)}{\partial x^2}$$

- We use

$$\frac{\partial}{\partial t}N(x,t) \rightarrow \frac{N(x,t+\tau)-N(x,t)}{\tau}$$

$$\frac{\partial^2}{\partial x^2}N(x,t) \rightarrow \frac{N(x+\Delta,t)-2N(x,t)+N(x-\Delta,t)}{\Delta^2}$$

to bring the equations in a form that is suitable for numerical simulation

Explicit forward in time central difference in space (FTCS) method

$$\frac{N(x,t+\tau)-N(x,t)}{\tau}=D\frac{N(x+\Delta,t)-2N(x,t)+N(x-\Delta,t)}{\Delta^2}$$

$$\Rightarrow \boxed{N(x,t+\tau)=\left(1-\frac{2\tau D}{\Delta^2}\right)N(x,t)+\left(N(x+\Delta,t)+N(x-\Delta,t)\right)}$$

- Stability criterion: $\tau \le \Delta^2/2D$
  - → a doubling of the spatial resolution requires a simultaneous reduction of the time step by a factor of four
- Numerical dispersion can approach exact dispersion for sufficiently fine gridding (see also lecture on the 1D Maxwell equation)

Crank-Nicolson method

$$\frac{N(x,t+\tau)-N(x,t)}{\tau}=D\frac{N(x+\Delta,t)-2N(x,t)+N(x-\Delta,t)}{\Delta^2}$$

$$\frac{N(x,t+\tau)-N(x,t)}{\tau}=D\frac{N(x+\Delta,t+\tau/2)-2N(x,t+\tau/2)+N(x-\Delta,t+\tau/2)}{\Delta^2}$$

$$\frac{N(x,t+\tau)-N(x,t)}{\tau}=D\frac{N(x+\Delta,t+\tau)-2N(x,t+\tau)+N(x-\Delta,t+\tau)+N(x+\Delta,t)-2N(x,t)+N(x-\Delta,t)}{2\Delta^2}$$

$$\boxed{-\frac{\tau D}{\Delta^2}N(x+\Delta,t+\tau)+2\left(1+\frac{\tau D}{\Delta^2}\right)N(x,t+\tau)-\frac{\tau D}{\Delta^2}N(x-\Delta,t+\tau)=\frac{\tau D}{\Delta^2}N(x+\Delta,t)+2\left(1-\frac{\tau D}{\Delta^2}\right)N(x,t)+\frac{\tau D}{\Delta^2}N(x-\Delta,t)}$$

known

Unconditionally stable implicit method

Tridiagonal linear system: $A\mathbf{N}=\mathbf{b}$

Diffusion equation in matrix notation

- Tremendous simplification of notation
- Example: $\dfrac{\partial^2}{\partial x^2} N(x,t) \rightarrow \dfrac{N(x+\Delta,t) - 2N(x,t) + N(x-\Delta,t)}{\Delta^2}$

$$\Delta^{-2} \underbrace{\begin{pmatrix} -2 & 1 & 0 & & & 0 \\ 1 & -2 & 1 & & & \\ 0 & 1 & -2 & & & \\ & & & \ddots & & 0 \\ & & & & -2 & 1 \\ 0 & & & 0 & 1 & -2 \end{pmatrix}}_{H} \begin{pmatrix} N(1,t) \\ N(2,t) \\ N(3,t) \\ \vdots \\ \vdots \\ N(L,t) \end{pmatrix} \equiv \Delta^{-2} H \underbrace{\begin{pmatrix} N(1,t) \\ N(2,t) \\ N(3,t) \\ \vdots \\ \vdots \\ N(L,t) \end{pmatrix}}_{\Phi(t)} \equiv \Delta^{-2} H \Phi(t)$$

Diffusion equation in matrix notation

- We keep time as a continuous variable

$$\frac{\partial \Phi(t)}{\partial t} = \alpha H \Phi(t) \quad (*)$$

- The square matrix H is a (linear) "operator" that acts on a vector
- The length of the vector $\Phi(t)$ is L, the number of lattice points that we use to represent the space in which the diffusion occurs
- As we have seen before, equation (*) is the generic form for most, if not all, basic equations of physics. Formal solution: $\Phi(t+\tau) = e^{\tau\alpha H}\Phi(t)$

Product formula approach

- In matrix form

$$\frac{\partial}{\partial t}\begin{pmatrix}\Phi_1(t)\\ \Phi_2(t)\\ \Phi_3(t)\\ \vdots\\ \vdots\\ \Phi_L(t)\end{pmatrix} = -D\Delta^{-2}\underbrace{\begin{pmatrix}2 & -1 & 0 & & & 0\\ -1 & 2 & -1 & & & \\ 0 & -1 & 2 & & & \\ & & & \ddots & & 0\\ & & & & 2 & -1\\ 0 & & & 0 & -1 & 2\end{pmatrix}}_{H}\begin{pmatrix}\Phi_1(t)\\ \Phi_2(t)\\ \Phi_3(t)\\ \vdots\\ \vdots\\ \Phi_L(t)\end{pmatrix}$$

- We decompose the matrix $H = A + B$ where

K. Michielsen, H. De Raedt, J. Przeslawski, and N. García, "Computer Simulation of Time-Resolved Optical Imaging of Objects Hidden in Turbid Media" Phys. Rep. 304, 89 – 144 (1998)

Product formula approach

$$A = -D\Delta^{-2}\left(\begin{array}{cc|cc|cc|c} 1 & -1 & 0 & & & & 0 \\ -1 & 1 & 0 & 0 & 0 & & \\ \hline 0 & 0 & \bullet & \bullet & 0 & & \\ & 0 & \bullet & \bullet & 0 & 0 & 0 \\ \hline & & 0 & 0 & \bullet & \bullet & 0 \\ & & & 0 & \bullet & \bullet & 0 \\ \hline 0 & & & 0 & 0 & 0 & \ddots \end{array}\right) \qquad B = -D\Delta^{-2}\left(\begin{array}{c|cc|cc|cc} 1 & 0 & 0 & 0 & & & 0 \\ \hline 0 & 1 & -1 & 0 & & & \\ 0 & -1 & 1 & 0 & 0 & 0 & \\ \hline 0 & 0 & 0 & \bullet & \bullet & 0 & \\ & & 0 & \bullet & \bullet & 0 & 0 \\ \hline & & & 0 & 0 & \ddots & \cdots \\ 0 & & & & 0 & \vdots & \ddots \end{array}\right)$$

- Second order formula: $e^{\tau H} = e^{\tau A/2} e^{\tau B} e^{\tau A/2}$
- The matrix exponential of a block-diagonal matrix is easy to compute: we need to compute

$$\exp\left(-\tau D\Delta^{-2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\right) \equiv \mathrm{e}^{aX}$$

Product formula approach

→We have to calculate e^{aX} with

$$X = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}; \quad a = -\tau D \Delta^{-2}$$

We use

$$e^{aX} = \sum_{k=0}^{\infty} \frac{a^k}{k!} X^k$$

Product formula approach

$$X^0 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \equiv I$$

$$X^1 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}^1 = X$$

$$X^2 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}^2 = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix} = 2X$$

$$X^3 = 2X^2 = 4X$$

$$X^4 = 4X^2 = 8X$$

$$X^5 = 8X^2 = 16X$$

$\cdots$

Product formula approach

$$\rightarrow \quad e^{aX} = I + aX + \frac{a^2}{2!}2X + \frac{a^3}{3!}4X + \frac{a^4}{4!}8X + \ldots$$

$$= I + aX + a^2X + \frac{2}{3}a^3X + \frac{1}{3}a^4X + \ldots$$

$$e^a = 1 + a + \frac{a^2}{2!} + \frac{a^3}{3!} + \frac{a^4}{4!} + \ldots$$

$$e^{2a} = 1 + 2a + 2a^2 + \frac{4a^3}{3} + \frac{2a^4}{3} + \ldots$$

$$\rightarrow \quad 2e^{aX} = 2I + 2aX + 2a^2X + \frac{4a^3}{3}X + \frac{2a^4}{3}X + \ldots$$

$$= 2I + X(2a + 2a^2 + \frac{4}{3}a^3 + \frac{2}{3}a^4 + \ldots) = 2I + X(e^{2a} - 1)$$

Product formula approach

$$\rightarrow e^{aX} = I + \frac{X}{2}(e^{2a}-1)$$

$$= I + \frac{1}{2}\begin{pmatrix} e^{2a}-1 & 1-e^{2a} \\ 1-e^{2a} & e^{2a}-1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1+e^{2a} & 1-e^{2a} \\ 1-e^{2a} & 1+e^{2a} \end{pmatrix}$$

- Special feature of this decomposition: unconditionally stable (independent of a) $\left\| e^{\tau H} \right\| \leq 1$ where $\left\| X \right\|$ denotes the eigenvalue of the matrix X of largest absolute value (eigenvalues 1 and e^{2a})

Product formula approach

- Implementation of the algorithm: example

$$\begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \vdots \\ \vdots \\ \Phi_L \end{pmatrix} \leftarrow e^{\tau B}\Phi = \left(\begin{array}{c|cc|cc|cc} e^{-\tau D\Delta^{-2}} & 0 & 0 & 0 & & & 0 \\ \hline 0 & \otimes & \oplus & 0 & & & \\ 0 & \oplus & \otimes & 0 & 0 & 0 & \\ \hline 0 & 0 & 0 & \otimes & \oplus & 0 & \\ & & 0 & \oplus & \otimes & 0 & 0 \\ \hline & & & 0 & 0 & \ddots & \cdots \\ 0 & & & & 0 & \vdots & \ddots \end{array}\right) \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \vdots \\ \vdots \\ \Phi_L \end{pmatrix}$$

- NEVER store the matrix!

Product formula approach

$$\begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \vdots \\ \vdots \\ \Phi_L \end{pmatrix} \leftarrow e^{\tau B}\Phi = \left(\begin{array}{c|cc|cc|cc} e^{-\tau D \Delta^{-2}} & 0 & 0 & 0 & & & 0 \\ \hline 0 & \otimes & \oplus & 0 & & & \\ 0 & \oplus & \otimes & 0 & 0 & 0 & \\ \hline 0 & 0 & 0 & \otimes & \oplus & 0 & \\ & & 0 & \oplus & \otimes & 0 & 0 \\ \hline & & & 0 & 0 & \ddots & \cdots \\ 0 & & & & 0 & \vdots & \ddots \end{array}\right) \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \vdots \\ \vdots \\ \Phi_L \end{pmatrix}$$

- We only need to pick a pair from Φ ,multiply them by a 2x2 matrix (except for Φ_1), store the resulting pair in the same memory location as the original pair and repeat for all possible pairs
- The same for $e^{\tau A/2}$, except that the pairs are different
- VERY SIMPLE to program, NEVER fails, EASY to generalize to 2D and 3D and to more complicated diffusion equations

Product formula approach

- Flexible
  - works entirely on vectors defined in real space (no FFT’s, ...)
  - no constraints on the shape of the sample
    - important for medical applications
- Efficient and accurate
  - CPU time scales linearly with the number of time steps and the total number of grid points
- Easy to program: $\begin{pmatrix}\bullet\\ \bullet\end{pmatrix} \leftarrow \begin{pmatrix}\bullet & \bullet\\ \bullet & \bullet\end{pmatrix}\begin{pmatrix}\bullet\\ \bullet\end{pmatrix}$

Exercise 1

- Use the 2nd-order product formula algorithm $e^{tH} \approx (e^{tA/2m} e^{tB/m} e^{tA/2m})^m$ to solve the one-dimensional diffusion equation.
  Perform simulations for the set of parameters:
  - $D = 1$ (only fixes the "scale")
  - $L = 1001$
  - $\Delta = 0.1$
  - $\tau = 0.001$
  - $m = 10000$

$$\left\langle x^p(t) \right\rangle = \frac{\int_0^{L\Delta} x^p N(x,t)dx}{\int_0^{L\Delta} N(x,t)dx} \approx \Delta^p \frac{\sum_{i=1}^{L} (i - i_0)^p \Phi_i(t)}{\sum_{i=1}^{L} \Phi_i(t)}$$

- For two different initial conditions:

$$\Phi_i(t=0) = \begin{cases} 1 & i = i_0 \equiv (L+1)/2 \\ 0 & i \neq i_0 \equiv (L+1)/2 \end{cases} \quad \text{and} \quad \Phi_i(t=0) = \begin{cases} 1 & i = 1 \\ 0 & i \neq 1 \end{cases}$$

- Plot $\Delta^{-2}\left(\left\langle x^2(t)\right\rangle - \left\langle x(t)\right\rangle^2\right)$ as a function of t
- Interpret the results!

Exercise 2: Diffusion comparison with random walk

- $N = 10000$ particles all start at the center $(L+1)/2$ of the system, making random walks as described earlier. $N_i(t)$ is the number of walkers at site i and discrete time t
- $L = 1001$
- $t = 0, 1, 2, \ldots, 10$

$$\langle x^p(t) \rangle = \frac{\sum_{i=1}^{L} (i - i_0)^p N_i(t)}{\sum_{i=1}^{L} N_i(t)}$$

- Plot $\langle x^2(t) \rangle - \langle x(t) \rangle^2$ as a function of t
- Compare with results of diffusion equation

$$\Delta = 0.1, \quad \tau = 0.001 \quad \Rightarrow D = \Delta^2 / 2\tau = 5$$

Not for students who get < 10 ECTS for the course

Report

Ms. Vrinda Mehta
v.mehta@fz-juelich.de
Dr. Fengping Jin
f.jin@fz-juelich.de
Dr. Madita Willsch
m.willsch@fz-juelich.de
Dr. Dima Nabok
d.nabok@fz-juelich.de

- Filename: Report_7_Surname1_Surname2.pdf, where Surname1 < Surname2 (alphabetical order).
  Example: Report_7_Jin_Willsch.pdf (Do not use "umlauts" or any other special characters in the names)
- Content of the report:
  - Names + matricle numbers + e-mail addresses + title
  - Introduction: describe briefly the problem you are modeling and simulating (write in complete sentences)
  - Simulation model and method: describe briefly the model and simulation method (write in complete sentences)
  - Simulation results: show figures (use grids, with figure captions !) depicting the simulation results. Give a brief description of the results (write in complete sentences)
  - Discussion: summarize your findings
  - Appendix: Include the listing of the program

Due date: 10 AM, June 26, 2023