FYS4150 - Computational Physics
Project 5:
Diffusion of neurotransmitters in the synaptic cleft

Candidate numbers:

November 19, 2015

Abstract

In this project we study diffusion of neurotransmitters in the brain across the synaptic cleft. In particular, we look at the solution to the diffusion equation for a particular choice of boundary conditions. Our choice of boundary conditions allows for an analytical solution that is found and later compared to the numerical results. The numerical solutions are found by use of three different integration schemes for partial differential equations: the Forward Euler scheme, the Backward Euler scheme and the Crank-Nicolson scheme. Finally, we compare these deterministic schemes to an implementation based on Monte Carlo methods and random walks. We find that ...`blah, results, blah`...

- Github repository link with all source files and benchmark calculations: https://github.com/henrisro/Project5.

- A list of all the code files can be found in the code listing in the end of this document.

1   Introduction

Diffusion of signal molecules is the dominant way of transportation in the brain. In this project we are going to study the solutions to the diffusion equation with restriction to some special boundary conditions. The basic process is illustrated in figure 1. In (1) the vesicles inside the axon approach the presynaptic membrane and merge with it in (2). The vesicle contains neurotransmitters and release them across the synaptic cleft (of with d) in (3). If we denote the concentration of neurotransmitters at distance x and time t from the presynaptic membrane by $u(x,t)$, the dynamics can be described with the diffusion equation:

$$\frac{\partial u(x,t)}{\partial t} = D\nabla^2 u(x,t), \tag{1}$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2}$ in the one dimensional case. This equation will be the subject of investigations with restrictions to the interval $x \in [0,d]$ for $d = 1$ and diffusion constant equal to unity $D = 1$. We will apply the boundary conditions

$$u(0,t) = 1 \ \ \forall t > 0 \ \ \text{and} \ \ u(d,t) = 0 \ \ \forall t > 0, \tag{2}$$

corresponding to a constant release of neurotransmitters at the presynaptic membrane ($u(0,t)$) and a constant absorption of transmitters at the postsynaptic membrane (at $u(d,t)$) at all times. The initial conditions will be taken to be:

$$u(x,t) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \in (0,d] \end{cases} \tag{3}$$

This basically means that all the neurotransmitters are located at $x = 0$ initially when the vesicles are opened.
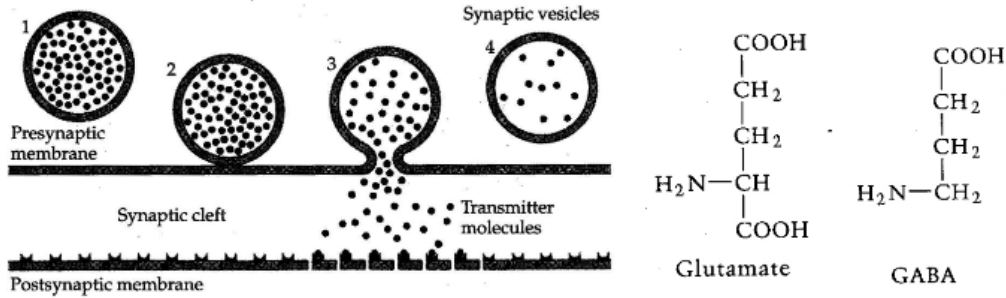

Figure 1: Left: Illustration of the release of vesicle from an axon. The neurotransmitters (two chemical structures are shown to the right) travel diffusively across the synaptic cleft and are absorbed in the postsynaptic membrane at a distance d away from the release point. The figure is taken from the project description [2], but originates from Thompson: "The Brain", Worth Publ., 2000.

The solutions will be studied by implementation of three different deterministic schemes in addition to a Monte Carlo based method. We will look in some detail at the Forward Euler scheme, the Backward Euler scheme and the Crank-Nicholson scheme for partial differential equations (PDEs). The methods are compared in terms of precision and stability.

2   Theory

In this section we derive the analytical solution to the diffusion problem outlined above. The closed form expression of the solution, although it will take the form of an infinite sum, will be valuable in terms of judging the numerical results later.

2.1   Analytical solution

To solve the one dimensional diffusion equation,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \tag{4}$$

with the boundary conditions in (2) and initial conditions in (3), we apply the standard *separation of variables* ansatz:

2

$$u(x,t) = X(x)T(t) + u_s(x) \tag{5}$$

We here separate out the steady-state solution $u_s(x) = 1 - x$, which trivially satisfies equation (4). When inserted into equation (4) we arrive at:

$$\frac{1}{X}\frac{d^2X}{dx} = \frac{1}{T}\frac{dT}{dt} = -k^2 \tag{6}$$

The last equality (i.e. both sides equal to a constant) follows since the left hand side is independent of t and the right hand side independent of x. Looking first at the $X(x)$ equation, we realize that it is simply the harmonic oscillator equation with solution $X(x) = a\cos(kx) + b\sin(kx)$. The boundary condition in equation (2) translates to (since we must have $X(0) = X(1) = 0$ after adding the steady-state solution) $a = 0$ and $\sin(k) = 0 \Rightarrow k_n = n\pi$ for $n \in \mathbb{N}$. We thus have:

$$X_n(x) = b_n \sin(n\pi) \tag{7}$$

The equation for $T(t)$ is a separable first order differential equation and have solutions:

$$T_n(x) = \exp(-(n\pi)^2 t) \tag{8}$$

The general solution is then a sum of (possibly all) the modes $X_n(x)T_n(t)$:

$$u(x,t) = 1 - x + \sum_{n=1}^{\infty} b_n \sin(n\pi x)e^{-(n\pi)^2 t} \tag{9}$$

The final step is to determine the coefficients b_n such that $u(x,0)$ fits the initial conditions in equation (3). This can be rewritten as

$$\sum_{n=1}^{\infty} b_n \sin(n\pi x) = u_0(x) = \begin{cases} 0 & \text{if } x = 0 \\ x - 1 & \text{if } x \in (0,1] \end{cases} \tag{10}$$

This is just as the Fourier series of the odd extension of the function $u_0(x)$ on the interval $[0,1]$. The coefficients are determined by Fourier's trick (see for instance [1]):

$$b_n = 2\int_0^1 dx \, (x-1)\sin(n\pi x) = -\frac{2}{n\pi} \tag{11}$$

We hence have arrived at the final closed form solution:

$$\boxed{u(x,t) = 1 - x - \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{\sin(n\pi x)}{n}e^{-(n\pi)^2 t}} \tag{12}$$

2.2 Stability of matrix iterations

Suppose we are given the iteration $\boldsymbol{V}_i = A\boldsymbol{V}_{i-1}$ for some matrix A and vectors $\boldsymbol{V}_i$. This implies that the vector after n iterations is given in terms of the initial one as: $\boldsymbol{V}_n = A^n \boldsymbol{V}_0$. By expanding $\boldsymbol{V}_0$ in the eigenvector basis of A, we can quickly understand the following theorem quite intuitively:

Theorem 1. *The solution to the iteration $\boldsymbol{V}_i = A\boldsymbol{V}_{i-1}$ will converge to a definite value if $\rho(A) < 1$, where $\rho(A)$ is the spectral radius of A:*

$$\rho(A) = \max\{|\lambda_n| : \det(A - \lambda_n \mathbb{1}) = 0\} \tag{13}$$

This is a central concept in Chaos theory when determining for example the stability of flows and maps[1].

3 Method / Algorithm

This section is devoted to the discussion of the three deterministic schemes for PDEs mentioned earlier. We derive the algorithms and investigate their stability properties expressed in terms of Δx and Δt. First some general nomenclature. We discretize the interval $[0, 1]$ with a step length determined by some integer n:

$$\Delta x = \frac{1}{n+1} \tag{14}$$

such that $x \mapsto x_i = i\Delta x$ for $i \in \{0, 1, \ldots, n+1\}$. A corresponding discretization in time is given by Δt (we will later discuss how this should be chosen depending on Δx, i.e. n). This means that $t \mapsto t_j = j\Delta t$ for $j \in \mathbb{N}$. The following discussion will be made in the context of the diffusion equation, here written compactly as $u_{xx} = u_t$ (the subscripts refer to derivatives). After discretization we use the notation $u(x, t) \mapsto u(x_i, t_j) \equiv u_{i,j}$. The following definition will also appear frequently in what follows:

$$\alpha \equiv \frac{\Delta t}{\Delta x^2} \tag{15}$$

The following discussion and derivations follow [2] closely.

3.1 Forward Euler scheme

In this scheme we use the two simplest possible approximations for the derivatives u_{xx} and u_t:

$$u_{xx} \approx \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} \mapsto \frac{1}{\Delta x^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \tag{16}$$

and

---

[1]This is for instance covered in Predrag Cvitanović's book on chaos theory: http://chaosbook.org/.

$$u_t \approx \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} \mapsto \frac{1}{\Delta t}(u_{i,j+1} - u_{i,j}) \tag{17}$$

This leads directly to an explicit scheme when equating the above expressions:

$$u_{xx} = u_t \Rightarrow \tag{18}$$

$$\frac{1}{\Delta x^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = \frac{1}{\Delta t}(u_{i,j+1} - u_{i,j}) \Rightarrow \tag{19}$$

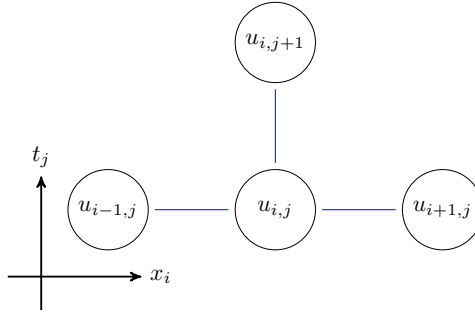$$u_{i,j+1} = \alpha u_{i+1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i-1,j} \tag{20}$$

Figure 2: Forward Euler computational grid scheme for the $(1 + 1)$-dimensional diffusion equation. $u_{i,j}$ is the discretized solution at position x_i and time t_j.

This scheme is explicit in the sense that it gives a recipe of how to find the solution at the next time step given the solution at the previous one. It should be initialized with the initial condition, $u_{i,0} = u_0(x_i)$ as given in equation (10). In terms of a matrix equation we can write:

$$\boldsymbol{U}_{j+1} = A\boldsymbol{U}_j \tag{21}$$

$$\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{n,j+1} \end{bmatrix} = \begin{bmatrix} 1 - 2\alpha & \alpha & 0 & \cdots & 0 & 0 \\ \alpha & 1 - 2\alpha & \alpha & \cdots & 0 & 0 \\ & & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha & 1 - 2\alpha \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{n,j} \end{bmatrix} \tag{22}$$

with the initial condition (in our case) $\boldsymbol{U}_0 = (u_0(\Delta x), u_0(2\Delta x), \ldots, u_0(1))^T = (1, 0, \ldots, 0)^T$. It is appropriate, also for later reference, to decompose the matrix as $A = \mathbb{1} - \alpha B$ where the elements of B can be expressed in terms of Kronecker deltas:

$$B_{i,j} = 2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j} \tag{23}$$

The subscripts here refer to matrix indices and have nothing to do with time and space coordinates as earlier.

Truncation errors
The discretized equations in (16) and (17) are easily seen to come from the Taylor expansions (with step lengths Δx and Δt respectively) around $u(x, t)$

$$u(x \pm \Delta x, t) = u(x, t) \pm \frac{\partial u(x, t)}{\partial x} \Delta x + \frac{\partial^2 u(x, t)}{\partial x^2} \Delta x^2 + \mathcal{O}(\Delta x^3) \tag{24}$$

$$u(x, t \pm \Delta t) = u(x, t) \pm \frac{\partial u(x, t)}{\partial t} \Delta t + \mathcal{O}(\Delta t^2) \tag{25}$$

and spell out for us that the Forward Euler scheme gives rise to errors of orders $\mathcal{O}(\Delta t)$ and $\mathcal{O}(\Delta x^2)$ when inserted into (16) and (17).

Stability condition
To determine for what combinations of Δt and Δx the Forward Euler scheme is stable, we investigate the spectral radius of the matrix B (the other part of $A = \mathbb{1} - \alpha B$ is proportional to $\mathbb{1}$ and gives rise to a simple unit contribution to the eigenvalues):

$$(B\boldsymbol{v})_i = \lambda_i v_i \tag{26}$$

$$\sum_{j=1}^{n} [2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}] v_j = \lambda_i v_i \tag{27}$$

We continue with expanding the vector components in a sine basis $v_i = \sin(i\theta)$ with $\theta = $ such that

$$\sum_{j=1}^{n} [2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}] v_j = \lambda_i v_i \tag{28}$$

$$2\sin(i\theta) - \sin((i+1)\theta) - \sin((i-1)\theta) = \lambda_i \sin(i\theta) \tag{29}$$

$$2(1 - \cos(\theta)) = \lambda_i \tag{30}$$

Hence in order to have $\rho(A) < 1$, we must have

$$|1 - 2\alpha(1 - \cos(\theta))| < 1 \Rightarrow \alpha = \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2} \tag{31}$$

The extremal eigenvalue occurs when $\theta = \pi$, at which the final inequality follows.

3.2   Backward Euler scheme

The Backward Euler scheme (just like for ordinary differential equations) exploits a time derivative approximation with a function evaluation backwards in time and the same approximation for u_{xx} as in equation (16). We (re)state both approximations to have a clear argument:

$$u_{xx} \approx \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} \mapsto \frac{1}{\Delta x^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \tag{32}$$

and

$$u_t \approx \frac{u(x,t) - u(x, t-\Delta t)}{\Delta t} \mapsto \frac{1}{\Delta t}(u_{i,j} - u_{i,j-1}) \tag{33}$$

Equating these leaves us with:

$$u_{xx} = u_t \Rightarrow \tag{34}$$

$$\frac{1}{\Delta x^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) = \frac{1}{\Delta t}(u_{i,j} - u_{i,j-1}) \Rightarrow \tag{35}$$

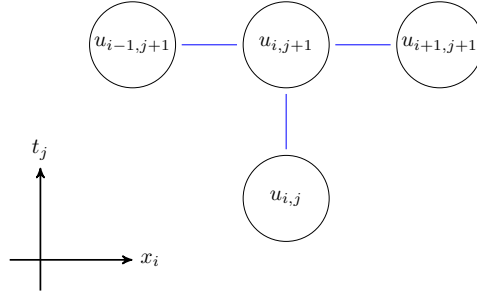$$u_{i,j-1} = -\alpha u_{i+1,j} + (1 + 2\alpha)u_{i,j} - \alpha u_{i-1,j} \tag{36}$$

Figure 3: Backward Euler computational grid scheme for the $(1 + 1)$-dimensional diffusion equation. $u_{i,j}$ is the discretized solution at position x_i and time t_j.

This recursion relation backwards in time does, as compared to the Forward Euler scheme, not give us the solution for the next time step in terms of the previous one. It instead takes the following form when written in matrix notation:

$$C\boldsymbol{U}_j = \boldsymbol{U}_{j-1} \tag{37}$$

$$\begin{bmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & \cdots & 0 & 0 \\ & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & -\alpha & 1+2\alpha \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{n,j} \end{bmatrix} = \begin{bmatrix} u_{1,j-1} \\ u_{2,j-1} \\ \vdots \\ u_{n,j-1} \end{bmatrix} \tag{38}$$

with $C = \mathbb{1} + \alpha B$ and B is the same as in (23). This asks for, at each time step j, nothing else than a set of linear equations to be solved. Phrased differently, a matrix inversion (meaning we must find C^{-1} with C tridiagonal as above) would give us the solution at each time step straight away: $\boldsymbol{U}_j = C^{-1}\boldsymbol{U}_{j-1}$. However, a matrix-vector multiplication at each time step is computationally consuming as it costs $\mathcal{O}(n^2)$ FLOPS and the matrix inversion even worse with its $\mathcal{O}(n^3)$ FLOPS. From Project 1, however, we know that solving such equations for a tridiagonal matrix can be done by an algorithm that takes only $\mathcal{O}(n)$ FLOPS; far superior when the dimension becomes large.

Truncation errors
Equations in (32) and (33), just as for the Forward Euler scheme, stem from the Taylor expansions (with step lengths Δx and Δt respectively) as in equation (24) and (25). It is straight

forwardly seen that this scheme gives identical truncation errors – of orders $\mathcal{O}(\Delta t)$ and $\mathcal{O}(\Delta x^2)$ – just as in the explicit method.

Stability condition

This time we need to consider the spectral radius of $C^{-1} = (\mathbb{1} + \alpha B)^{-1}$. However, the work done in the stability discussion of the Forward Euler method gives us the answer quickly. We know from the earlier discussion that the eigenvalues of C are strictly positive and larger than unity: $\mu_i = 1 + 2\alpha(1 - \cos(\theta)) > 1$. This implies that the eigenvalues of C^{-1} are strictly less than 1: $\mu_i^{-1} < 1$. Furthermore we have $\rho(C^{-1}) < 1$ for all combinations of Δt and Δx; a rather weak stability condition (i.e. no restrictions on the step lengths).

3.3 Crank-Nicolson scheme

The Crank-Nicolson scheme is, in a sense, an equally weighted compromise between the forward Euler scheme and the backward Euler scheme. To see this explicitly, let us rephrase and compare the two previous schemes:

$$\frac{1}{\Delta t}(u_{i,j} - u_{i,j-1}) = \frac{1}{\Delta x^2}(u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1}) \quad \text{(Forward Euler)} \tag{39}$$

$$\frac{1}{\Delta t}(u_{i,j} - u_{i,j-1}) = \frac{1}{\Delta x^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \quad \text{(Backward Euler)} \tag{40}$$

Adding the two equations gives the Crank-Nicolson scheme[2]:

$$\frac{2}{\Delta t}(u_{i,j} - u_{i,j-1}) = \frac{1}{\Delta x^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) \Rightarrow \tag{41}$$

$$-\alpha u_{i+1,j} + 2(1 + \alpha)u_{i,j} - \alpha u_{i-1,j} = \alpha u_{i+1,j-1} + 2(1 - \alpha)u_{i,j-1} + \alpha u_{i-1,j-1} \tag{42}$$
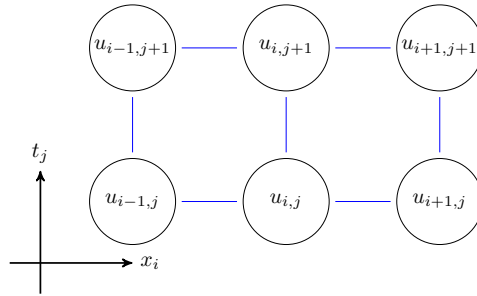
Figure 4: Crank-Nicolson computational grid scheme for the $(1 + 1)$-dimensional diffusion equation. $u_{i,j}$ is the discretized solution at position x_i and time t_j.

or in matrix notation:

---

[2] A more general approach would be to assign different weights the two contributions in what is known as a θ-rule [2].

$$(2\mathbb{1} + \alpha B)\boldsymbol{U}_j = (2\mathbb{1} - \alpha B)\boldsymbol{U}_{j-1} \tag{43}$$

$$\begin{bmatrix} 2+2\alpha & -\alpha & 0 & \cdots & 0 & 0 \\ -\alpha & 2+2\alpha & -\alpha & \cdots & 0 & 0 \\ & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & -\alpha & 2+2\alpha \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{n,j} \end{bmatrix} = \tag{44}$$

$$\begin{bmatrix} 2-2\alpha & \alpha & 0 & \cdots & 0 & 0 \\ \alpha & 2-2\alpha & \alpha & \cdots & 0 & 0 \\ & \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & \alpha & 2-2\alpha \end{bmatrix} \begin{bmatrix} u_{1,j-1} \\ u_{2,j-1} \\ \vdots \\ u_{n,j-1} \end{bmatrix} \tag{45}$$

Thus $\boldsymbol{U}_j = (2\mathbb{1} + \alpha B)^{-1}(2\mathbb{1} - \alpha B)\boldsymbol{U}_{j-1}$. This looks very similar to the Backward Euler scheme with the only difference that we must at each time step first calculate $\tilde{\boldsymbol{U}}_{j-1} = (2\mathbb{1} - \alpha B)\boldsymbol{U}_{j-1}$ before sending $(2\mathbb{1} + \alpha B)\boldsymbol{U}_j = \tilde{\boldsymbol{U}}_{j-1}$ to the tridiagonal matrix solver.

Truncation errors

The Crank-Nicolson scheme was presented above as the sum of two equally weighted contributions to u_t – at steps $j+1$ and j (figure 4). The formulas for u_{xx} were not altered, so the truncation errors in x are still of the same order $\mathcal{O}(\Delta x^2)$. To see what happens to the truncation in t, we can Taylor expand the involved quantities around $t' = t + \Delta t/2$ and only care about the errors in powers of Δt:

$$u(x \pm \Delta x, t) = u(x, t') \pm \frac{\partial u(x, t')}{\partial x}\Delta x + \frac{\partial^2 u(x, t')}{\partial x^2}\Delta x^2 + \tag{46}$$

$$+ \frac{\partial u(x, t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x, t')}{\partial t^2}\frac{\Delta t^2}{4} \pm \frac{\partial^2 u(x, t')}{\partial x \partial t}\frac{\Delta x \Delta t}{2} + \mathcal{O}(\Delta t^3) \tag{47}$$

$$u(x \pm \Delta x, t) = u(x, t') \pm \frac{\partial u(x, t')}{\partial x}\Delta x + \frac{\partial^2 u(x, t')}{\partial x^2}\Delta x^2 + \tag{48}$$

$$- \frac{\partial u(x, t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x, t')}{\partial t^2}\frac{\Delta t^2}{4} \mp \frac{\partial^2 u(x, t')}{\partial x \partial t}\frac{\Delta x \Delta t}{2} + \mathcal{O}(\Delta t^3) \tag{49}$$

$$u(x, t + \Delta t) = u(x, t') + \frac{\partial u(x, t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x, t')}{\partial t^2}\frac{\Delta t^2}{4} + \mathcal{O}(\Delta t^3) \tag{50}$$

$$u(x, t) = u(x, t') - \frac{\partial u(x, t')}{\partial t}\frac{\Delta t}{2} + \frac{\partial^2 u(x, t')}{\partial t^2}\frac{\Delta t^2}{4} + \mathcal{O}(\Delta t^3) \tag{51}$$

Putting these expansions carefully into equation (42) (with the discretized labellings such as $u(x + \Delta x, t) = u_{i+1,j}$ etc.) and collecting the surviving terms, one does in fact see that the leading order truncation error in time is $\mathcal{O}(\Delta t^2)$ – an improvement as compared to the Forward and Backward Euler schemes.

Stability condition

We need to look at the spectral radius of $(2\mathbb{1} + \alpha B)^{-1}(2\mathbb{1} - \alpha B)$. We can also this time reuse some earlier derived results. Recall the eigenvalues of B: $\lambda_i = 2(1 - \cos(\theta)) \in (0, 4]$. The spectral radius condition thus translates to the following in this case:

$$\rho\left((2\mathbb{1}+\alpha B)^{-1}(2\mathbb{1}-\alpha B)\right) = \frac{|2-\alpha\lambda_i|}{|2+\alpha\lambda_i|} < 1 \tag{52}$$

This inequality is quite trivially seen to be satisfied since $\lambda_i > 0$. It is even more clearly seen by considering the two cases $\alpha\lambda_i < 2$ and $\alpha\lambda_i > 2$ separately. In total, we arrive at the same condition as for the Backward Euler scheme: all combinations of Δt and Δx results in a stable solution.

3.4 Monte Carlo methods and random walks

4 Results

5 Discussion

6 Conclusion

7 Appendix

References

[1] M. L. Boas. *Mathematical Methods in the Physical Sciences.* Kaye Pace, 2006.

[2] M. Hjort-Jensen. Computational Physics, Lecture Notes Fall 2015. Department of Physics, University of Oslo, 2015.

Code listing