LECTURE 4-2
DIFFUSION EQUATION

Min Sup Hur

Mar. 23 2023

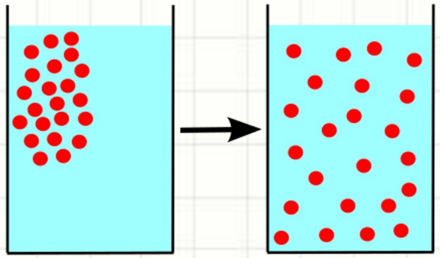
Outline

- Diffusion Equation
- Analytic Solution in 1D
- 1st Order Method
- 2nd Order Method
- Alternating-Direction Implicit Method for 2D (and higher).

Reference: Numerical Recipes in C

Diffusion Equation

Diffusion



- Fick’s law of diffusion $\mathbf{\Gamma} = -D\nabla u$ $\mathbf{\Gamma}$ is the particle flux
- Flux conservation $\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{\Gamma} = 0$
- Fick’s second law of diffusion: combining two equation above

$$\frac{\partial u}{\partial t} = D\nabla^2 u$$

- When there exist particle source, above equation is generalized to

$$\frac{\partial u}{\partial t} = D\nabla^2 u + S(\mathbf{x})$$

Analytic Solution in 1D

- Separation the variables as $u = T(t)S(x)$, where T and S are time-dependent-only and position-dependent-only functions.
- Put this into the diffusion equation to get

$$\frac{1}{D}\frac{T'}{T} = \frac{S''}{S} = -k^2$$

- The solution for S is

$$S = A \sin kx + B \cos kx$$

- The boundary condition is $S = 0$ at x=0 and L, where L is the length of the system. From this, $k = \frac{N\pi}{L}$ and B=0, where N is an integer.
- The time-part solution is, $T \propto e^{-t/\tau}$, where $\tau = \frac{1}{N^2\pi^2}\frac{L^2}{D}$, is the characteristic time of diffusion (i.e. how fast the particles diffuse).
- The solution for a specific initial mode k,

$$u = u_0 e^{-t/\tau} \sin kx$$

- When the system starts with more general profile of n,

$$u = \sum_{N=1}^{\infty} A_N e^{-t/\tau_N} \sin k_N x$$

- Note that higher harmonics diffuse faster.

Methods of 1st Order in Time

FTCS method

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\,\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}$$

$$u_j^{n+1} = u_j^n + \alpha\left[u_{j+1}^n - 2u_j^n + u_{j-1}^n\right] \qquad \alpha = \frac{D\Delta t}{\Delta x^2}$$

Amplification factor from von Neumann stability analysis $\xi = 1 - \frac{4D\Delta t}{(\Delta x)^2}\sin^2\left(\frac{k\Delta x}{2}\right)$

Stability condition $\frac{2D\Delta t}{(\Delta x)^2} \leq 1$

BTCS method

$$u_j^{n+1} = u_j^n + \alpha\left[u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right]$$

Amplification factor $\xi = \frac{1}{1 + 4\alpha\sin^2\left(\frac{k\Delta x}{2}\right)}$ Unconditionally stable

Methods of 1st Order in Time

Example 1. Write a FTCS diffusion solver. Set the initial shape of u to $\sin(\pi x/L)$, where L is the system length. Run it with xmax=1, dx=0.01, dt=0.001, D=0.05, imax=500.

Answer: Look at the code 'diffusion-1d-ftcs.py'.

Example 2. Add some higher harmonics to the initial condition of u. Observe the long-time behavior of u. How does the shape of u change? Explain it in terms of the characteristic diffusion time.

Answer: Look at the code 'diffusion-1d-ftcs-HH.py'.

Crank-Nicolson Method

- CN method is an implicit, second-order-in-time method.

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} = \frac{D}{2}\left[\frac{(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1})+(u_{j+1}^n-2u_j^n+u_{j-1}^n)}{(\Delta x)^2}\right]$$

- Average of $D\nabla^2 u$ at n and n+1 steps gives the effect of evaluating $\partial u/\partial t$ at n+1/2 step. Hence the left-hand-side is the 3-point derivative with the 2nd-order.

- The amplification factor:

$$\xi = \frac{1-2\alpha\sin^2\left(\frac{k\Delta x}{2}\right)}{1+2\alpha\sin^2\left(\frac{k\Delta x}{2}\right)} \quad \text{with} \quad \alpha = \frac{D\Delta t}{\Delta x^2}$$

Unconditionally stable!

- The discretized equation is

$$\beta u_j^{n+1} - \alpha u_{j+1}^{n+1} - \alpha u_{j-1}^{n+1} = \alpha u_{j+1}^n + \alpha u_{j-1}^n + \gamma u_j^n \equiv f_j^n$$

with $\beta = 2(1+\alpha)$ and $\gamma = 2(1-\alpha)$

Alternating-Direction Implicit Method

- To solve more than 2D systems, the ADI (alternating-direction implicit) method is quite powerful.
- For the first half-step, advance in one direction implicitly, keeping the other. In the next half-step, advance the other, keeping the first direction.

advance x-direction keep y-direction

$$u_{j,l}^{n+1/2} - u_{j,l}^{n} = \frac{\alpha}{2}\left[u_{j+1,l}^{n+1/2} - 2u_{j,l}^{n+1/2} + u_{j-1,l}^{n+1/2} + u_{j,l+1}^{n} - 2u_{j,l}^{n} + u_{j,l-1}^{n}\right]$$

keep x-direction advance y-direction

$$u_{j,l}^{n+1} - u_{j,l}^{n+1/2} = \frac{\alpha}{2}\left[u_{j+1,l}^{n+1/2} - 2u_{j,l}^{n+1/2} + u_{j-1,l}^{n+1/2} + u_{j,l+1}^{n+1} - 2u_{j,l}^{n+1} + u_{j,l-1}^{n+1}\right]$$

- The discrete equations used in the code are

$$\beta u_{j,l}^{n+1/2} - \alpha u_{j+1,l}^{n+1/2} - \alpha u_{j-1,l}^{n+1/2} = \alpha u_{j,l+1}^{n} + \alpha u_{j,l-1}^{n} + \gamma u_{j,l}^{n} = f_{j,l}^{n}$$
$$\beta u_{j,l}^{n+1} - \alpha u_{j,l+1}^{n+1} - \alpha u_{j,l-1}^{n+1} = \alpha u_{j+1,l}^{n+1/2} + \alpha u_{j-1,l}^{n+1/2} + \gamma u_{j,l}^{n+1/2} = f_{j,l}^{n+1/2}$$

$$\beta = 2(1+\alpha) \qquad \gamma = 2(1-\alpha)$$

- This method is unconditionally stable.
- The ADI method can be applied to the relaxation solver of the Poisson equation.

2D Diffusion in Python

Example 3. Create a python script for 2D simulation of the diffusion. Initialize u by $\sin(\pi x/L_x)\sin(\pi y/L_y)$ and its harmonics.

Answer: Look at the code 'diffusion-2d-adi.py'. Main part of the code is,

Half-step advance in x

Another half-step advance in y

```
...
while i<imax:
        f[1:-1,1:-1]=g*u[1:-1,1:-1]+a*u[0:-2,1:-1]+a*u[2:,1:-1]   # f^n
        j=0
        while j<Nx-1:   f[1:-1,j+1] -= f[1:-1,j]*(ll/d[j]); j+=1    # f-transform by diagonalization
        j=Nx-1
        while j>0:  u[1:-1,j] = (f[1:-1,j] - uu*u[1:-1,j+1])/d[j];  j -=1  # backsubstitution for u^n+1/2

        f[1:-1,1:-1]=g*u[1:-1,1:-1]+a*u[1:-1,0:-2]+a*u[1:-1,2:]     # f^n+1/2
        l=0
        while l<Ny-1:                  f[l+1,1:-1] -= f[l,1:-1]*(ll/d[l]); l+=1    # f-transform
        l=Ny-1
        while l>0:    u[l,1:-1] = (f[l,1:-1] - uu*u[l+1,1:-1])/d[l];  l -=1    # backsubsti. for u^n+1

        i+=1
```

END OF LECTURE 4-2