Numerical Methods in ESP

Numerical Methods II

Graziano Giuliani

International Centre for Theoretical Physics

Second Semester 2023-24

`/afs/ictp.it/public/g/ggiulian/WORLD/num2_lesson5.pdf`

The Diffusion Equation

- Consider the linear diffusion equation:

$$\frac{\partial \phi}{\partial t} = K \frac{\partial^2 \phi}{\partial x^2} \tag{1}$$

  where K is a constant.

Assume the domain is periodic in x. If the initial condition is wavelike, $\phi(x, 0) = e^{ikx}$, then the exact solution can be found assuming $\phi(x, t) = \Phi(t)e^{ikx}$.

$$\frac{d\Phi}{dt} e^{ikx} = -k^2 K \Phi(t) e^{ikx} \tag{2}$$

The solution can be found integrating now in time, to obtain:

$$\Phi(t) = \Phi(0)e^{-k^2 Kt} \tag{3}$$

Considering that $\Phi(0) = 1$, the solution of the initial diffusion problem is:

$$\phi(x, t) = e^{-k^2 Kt} e^{ikx} \tag{4}$$

The solution is a stationary wave decreasing in amplitude, with damping faster for short waves than for long waves.

Diffusion Discretization and stability 1

- Using a second-order accurate centered difference scheme we can discretize [1]

$$\frac{\phi_j^{n+1} - \phi_j^{n-1}}{2\Delta t} = K\left(\frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{\Delta x^2}\right) \tag{5}$$

We will now analyze the stability of the numerical solution in the equation [5] with the now usual substitution of $\phi_j^n = A^n e^{ikj\Delta x}$:

$$\begin{aligned} A^{n+1}e^{ikj\Delta x} &= A^{n-1}e^{ikj\Delta x} \\ &\quad + \frac{2K\Delta t}{\Delta x^2}\left(A^n e^{ik(j+1)\Delta x} - 2A^n e^{ikj\Delta x} + A^n e^{ik(j-1)\Delta x}\right) \\ A^2 &= 1 + A\frac{2K\Delta t}{\Delta x^2}\left(e^{ik\Delta x} - 2 + e^{-ik\Delta x}\right) \end{aligned} \tag{6}$$

Diffusion Discretization and stability 2

Using some trigonometric relations:

$$A^2 + A\frac{2K\Delta t}{\Delta x^2}(2 - 2cos(k\Delta x)) - 1 = 0$$

$$A = -\frac{2K\Delta t}{\Delta x^2}(1 - cos(k\Delta x)) \pm \sqrt{\left[\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1 - cos(k\Delta x))^2 + 1\right]} \tag{7}$$

We can now note that:

- Both roots A^- and A^+ are real and their product is $A^+A^- = -1$
- $A^- \leq -1$ because $\sqrt{(1+y)} \approx 1 + 1/2y$ and therefore $|A| > 1$, so the mode corresponding to this root is unconditionally unstable and it is a computational one.
- A^+ lies between 0 and 1 and this corresponds to the physical mode.

Diffusion Discretization and stability 3

Instead of the second order discretization in [5], let us analyze a first-order forward difference for the time derivative.

$$\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} = K \left(\frac{\phi_{j+1}^n - 2\phi_j^n + \phi_{j-1}^n}{\Delta x^2} \right) \tag{8}$$

If we conduct a Von Neumann stability analysis now for this equation we obtain

$$A = 1 - \frac{2K\Delta t}{\Delta x^2}(1 - cos(k\Delta x)) \tag{9}$$

- Because $0 \leq (1 - cos(k\Delta x)) \leq 2$, we conclude that:

$$|A| \leq 1 \rightarrow \frac{2K\Delta t}{\Delta x^2} \leq 1 \rightarrow \Delta t \leq \frac{\Delta x^2}{2K} \tag{10}$$

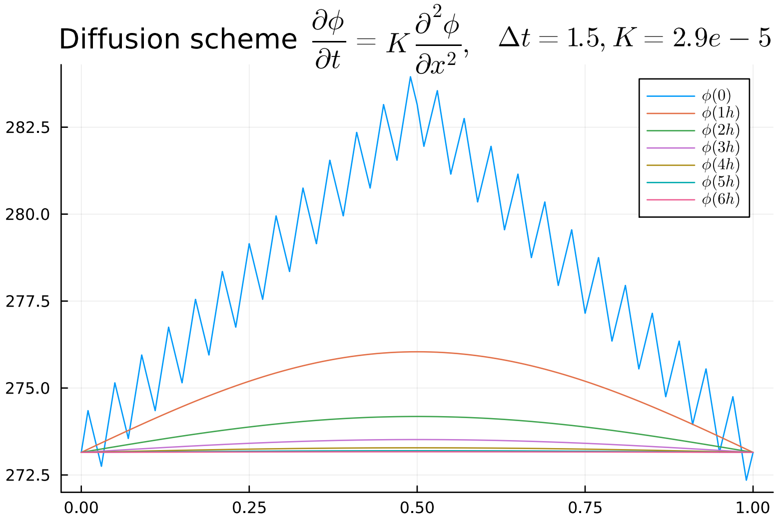
- The scheme is conditionally stable and because of the Δx^2 a small increase in resolution requires a big reduction of timestep.

Exercise on Diffusion

- Use the forward difference scheme in [8] to solve the diffusion problem in [1]. Use a spatial resolution of $\Delta x = 0.01m$ with a diffusion coefficient $K = 2.9E^{-5}$. Integrate for at least 6 hours and show the solution every hour. Let the initial condition be the following function, describing the temperature distribution along a $1m$ metal rod heated in the middle point and with extrema kept at a constant temperature of $T_0 = 273.15K$ (Dirichlet boundary condition):

$$
\begin{aligned}
\phi(x,0) &= \begin{cases} 273.15 + 20x + \sin(50\pi x) & for \quad 0 \le x \le 0.5 \\ 273.15 + 20 - 20x + \sin(50\pi x) & for \quad 0.5 < x \le 1 \end{cases} \\
\phi(0,t) &= 273.15, \quad \forall t \\
\phi(1,t) &= 273.15, \quad \forall t
\end{aligned}
\tag{11}
$$

Expected result



Julia Code

```
mult = K*dt/(dx^2)
function diffusion(phi_now,phi_new)
  phi_new[begin] = temp0;
  phi_new[end] = temp0;
  phi_new[begin+1:end-1] = phi_now[begin+1:end-1] + mult *
    (phi_now[begin+2:end]-2.0*phi_now[begin+1:end-1] +
     phi_now[begin:end-2]);
end;
phi_new = Array(phi)
t = t0;
while (t < t1)
  diffusion(phi,phi_new);
  global phi[:] = phi_new;
  global t = t + dt;
  if mod(t,tp) < dt
    it = Int(round(t)/3600.0)
    plot!(...);
  end;
end;
```