1 September 22nd, 2020

1.1 Three Time Level Scheme

Recalling the θ-method, when $\theta = \frac{1}{2}$, note that both the left and right hand sides are symmetric with respect to $n + \frac{1}{2}$, making it similar to the central difference. This is the intuition for why it has second order with respect to both time and space.

For the previous schemes, we use forward difference to expand u_t. Another way to express this is using the central difference instead:

$$\frac{U_j^{n+1} - U^{n-1})j}{2\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2}.$$

Note that both sides are symmetric with respect to n. This scheme is second order in both time and space, and is an explicit scheme. Note that it is a three-time level scheme, meaning we need the values at $n-1$ and n, and after that we can get the values for $n+1$.

Investigating the stability using Von Neumann stability analysis, we have:

$$U_j^n = \lambda^n e^{ik(j\Delta x)} \implies \frac{\lambda - \lambda^{-1}}{2\Delta t} = \frac{-4\sin^2(\frac{1}{2}k\Delta x)}{(\Delta x)^2}.$$

$$\implies \lambda^2 + 8\lambda\mu\sin^2(\frac{1}{2}k\Delta x) - 1 = 0.$$

This has two roots: λ_1, λ_2 with $\lambda_1 \cdot \lambda_2 = -1$. This means that $|\lambda_1| > 1$ or $|\lambda_2| > 1$, meaning that the scheme is always unstable. As such, this three level scheme is explicit and second order in both time and space, it is unstable.

> **Remark 1.1** — Note that since $n+1$ requires n and $n-1$, we'd have to use the forward difference explicit scheme for the first iteration, and then we can use the three time level scheme.

As a summary, we've investigated the following schemes:

- **Explicit Scheme**
  - Advantage: simple and easy to implement
  - Disadvantage: has a stability constraint of $\mu = \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$.
- **Implicit Scheme**
  - Advantage: unconditionally stable, meaning there is no restriction on Δt
  - Disadvantage: needs to solve a linear system at each time step

> **Remark 1.2** — In high dimensions, e.g. $d = 2, 3$ then implicit scheme is preferred, as it has a lower computational cost.

1.2 Boundary Conditions

Recall we are looking at the 1D heat equation:

$$u_t = u_{xx}, \quad 0 \leq x \leq 1.$$

Currently, we are solving with zero boundary conditions, i.e.:

$$u(0,t) = u(1,t) = 0.$$

and with initial conditions:

$$u(x,0) = u_0(x).$$

Recall for the finite difference methods before, we only calculate for the interior points, $j = 1, 2, \ldots, N-1$. This is because for $j = 0$ and $j = N$, the solution is known (since it is the boundary condition).

Note that we do use the boundary data for $j = 1$ and $j = N-1$, for example:

$$\frac{U_1^{n+1} - U_1^n}{\Delta t} = \frac{U_2^n + 2U_1^n + \cancel{U_0^n}}{(\Delta x)^2}.$$

If we instead consider the case where the boundary conditions are non zero but constant, i.e.:

$$u(0,t) = a, \quad u(1,t) = b, \quad a, b > 0.$$

Then, we can still calculate it, e.g.:

$$\frac{U_1^{n+1} - U_1^n}{\Delta t} = \frac{U_2^n + 2U_1^n + a}{(\Delta x)^2}.$$

Definition 1.3. The above boundary conditions are **Dirichlet boundary conditions**, as the function value is specified at the boundaries.

Definition 1.4. We can also have the **Neumann boundary conditions**, where we specify the derivative of the function at the boundaries

Example 1.5

An example of Neumann boundary condition would be

$$u_x(0,t) = 0, \quad u_x(1,t) = 0.$$

Remark 1.6 — The physical representation of Neumann boundary condition would be the heat flux, for example if $u_x(0,t) = u_x(1,t) = 0$, then there is no heat entering or exiting, meaning it is insulating.

Let us consider the insulating case with the explicit scheme:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta x)^2}.$$

In particular, consider $j = 1$, where:

$$\frac{U_1^{n+1} - U_1^n}{\Delta t} = \frac{U_2^n + 2U_1^n + U_0^n}{(\Delta x)^2}.$$

When $n = 0$, U_0^n is known (just the initial condition), but U_0^n is not known for $n = 1, 2, \ldots$ To fix this, we can use finite difference to discretize the boundary conditions.

To use central difference at the boundaries, we can introduce a ghost point at $x_{-1} = -\Delta x$ and $x_{N+1} = 1 + \Delta x$, meaning that at $x = 0$:

$$x_0 = 0 \approx \frac{U_1 - U_{-1}}{2\Delta x}.$$

which has second order accuracy. Meanwhile at $x = 1$, we have:

$$u_x = 0 \approx \frac{U_{N+1} - U_{N-1}}{2\Delta x}.$$

Now instead of only evaluating interior points, we also evaluate at the boundary points.

In particular when $j = 0$, we have:

$$\frac{U_0^{n+1} - U_0^n}{\Delta t} = \frac{U_1^n - U_0^n + U_{-1}^n}{(\Delta x)^2}.$$

With left boundary condition:

$$\frac{U_1^{n+1} - U_{-1}^{n+1}}{2\Delta x} = 0.$$

Thus, we first update the solution

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{(\Delta x)^2}\left(U_{j+1}^n - 2U_j^n + U_{j+1}^n\right), \quad j = 0, 1, , \ldots, N.$$

then we update the boundaries:

$$U_{-1}^{n+1} = U_1^{n+1}.$$
$$U_{N+1}^{n+1} = U_{N-1}^{n+1}.$$

We can do the same thing for the implicit scheme too. In summary, for Neumann boundary we need to introduce ghost points in order to discretize the boundary condition to maintain the second order accuracy.