The University of British Columbia
MECH 479

Module 4
Stability analysis & Temporal discretization

October 11, 2022

Stability analysis & Temporal discretization

This module introduces the stability analysis for the selected finite difference discretization schemes, which will be followed by popular temporal discretization techniques used in CFD.

1 Consistency, Stability and Convergence

1.1 Consistency

A numerical scheme is consistent if it recovers the exact partial differential equation as the grid spacing and time step size are reduced. In other words, the truncation error of the scheme must go to zero in the limit $\Delta x \to 0$ and $\Delta t \to 0$. This is usually the case. However, this is not always true. To demonstrate this we consider the Dufort-Frankel scheme for linear diffusion

$$u_i^{t+1} = u_i^{t-1} + \frac{2\beta\Delta t}{\Delta x^2}\left(u_{i-1}^t - u_i^{t+1} - u_i^{t-1} + u_{i+1}^t\right) + O\left(\Delta x^2, \Delta t^2, (\Delta t/\Delta x)^2\right)$$

While this scheme may have some useful properties, we note that it has a peculiar term in the truncation error of $O\left((\Delta t/\Delta x)^2\right)$. This can be obtained from a Taylor series expansion of the second derivative. This is concerning, as for any scheme to be consistent with the original partial differential equation we require the truncation error to go to zero. However, if we use a naive approach and simply refine Δx and Δt at the same rate, this term will not go to zero, and the scheme will not be consistent.

For example, if we reduce both the grid spacing and time step size to infinitesimal this $\left((\Delta t/\Delta x)^2\right)$ will remain the same. Hence, in order for the Dufort-Frankel scheme to be consistent with our original partial differential equation we should refine the grid spacing faster than the time step size. For example, if we reduce Δt by a half we should reduce Δx by a factor of four. This does not mean the Dufort-Frankel scheme is bad, per-se, but it does mean that care needs to be taken when using it.

1.2 Stability

If we can demonstrate our schemes are consistent, then we know that in the limit $\Delta x \to 0$ and $\Delta t \to 0$ we recover the exact partial differential equation. However, this is impossible to achieve in practice as it would require an infinite number of grid points and time steps. When considering stability, we are concerned with whether our numerical scheme will provide physical solutions when both Δx and Δt are finite. Let's start by considering what we mean by a physical solution.

In order to advance our solution in time, we start with some initial conditions. Then, by inserting this initial condition into our scheme we approximate the solution at the next timestep $t + \Delta t$. Then, we insert this approximation back into our scheme to approximate the solution at time $t + 2\Delta t$, and this process is repeated over and over again until we reach our final desired time. Hence, the way we advance our simulation in time is effectively a feedback loop, with the output of each time step being recycled back through the numerical scheme to get the solution at each consecutive time step. As an analogy, we can consider what happens in other simple feedback loops, such as a microphone and speaker. When a performer sings into a microphone their voice is amplified and played back through the speaker. We expect that this produces a physical replication of their voice, just at a louder volume for the audience. However, if the sound from the speaker is louder than the singer's voice at the microphone it will get amplified, and played through the speaker at a louder volume, and this cycle then repeats. This results in feedback noise, usually a high-pitched ringing that sounds nothing like the original performer, and usually happens when the performer moves too close to the speaker.

Since our numerical scheme is applied as a feedback loop, the exact same kind of thing can happen. If it amplifies our solution each time step, then the solution will continue to grow, eventually leading to non-physical values such as near-infinite density or pressure. This is colloquially referred to as the solution blowing up. In contrast, if the scheme damps our solution at each time step it will tend towards physical values, such as the background density or pressure. While this is perhaps not desirable in terms of accuracy, which will be explored later, it is a desirable property in that the solution remains stable and bounded between the initial condition and background state.

1.3 Convergence

If a scheme is consistent, recovering the exact partial differential equation in the limit $\Delta x \to 0$ and $\Delta t \to 0$, and stable, in that the approximate solution does not grow unbounded with time, then Lax's equivalence theorem can be applied. In summary, the theorem states that when given a properly-posed initial value problem, and a numerical scheme that is consistent, stability is the necessary and sufficient condition for convergence. In other words, if we can show that our numerical schemes are consistent and stable, then we can be sure that they will converge to the true solution of the partial differential equation in the limit $\Delta x \to 0$ and $\Delta t \to 0$.

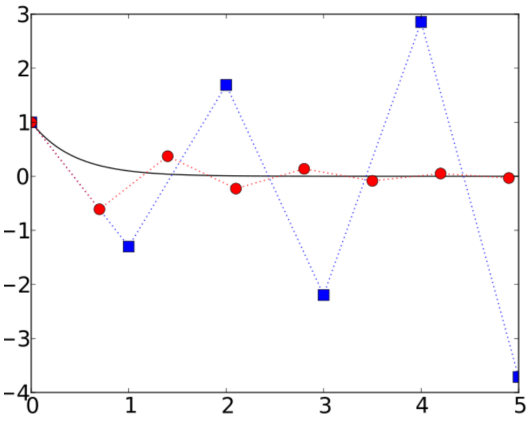


Figure 1: Numerical instability

2 Numerical instability

The numerical stability of a finite difference equation is linked with numerical error. According to the Lax Equivalence theorem, stability is the necessary and sufficient condition for convergence in a well-posed problem. An improper numerical discretization may lead to unbounded growth of errors for stable physical problems (as shown in Fig. (1)). This is referred to as numerical instability.

Numerical instability is usually characterized by the divergence of numerical solutions or local errors that create persistent unphysical or spurious oscillation. Since round-off errors are inevitable in discretization, the analysis of the stability nails down to the amplification of errors during the evolution of the solution. A stable discretization needs to ensure that any perturbation on the numerical solution should not be amplified without bounds.

We define the error as $\varepsilon_i^n = u_i^n - \overline{u}_i^n$, where u_i^n is the computed solution at node i and time step $t = n\Delta t$ and $\overline{u}_i^n$ denotes the exact solution of the finite difference scheme (not the exact solution of the PDE). The stability condition can be expressed as the error between u and $\overline{u}$ should remain uniformly bounded for $n \to \infty$ at fixed Δt. Mathematically:

$$\lim_{n\to\infty} |\varepsilon_i^n| \leq K \text{ at fixed } \Delta t$$

where K is independent of n. Note that the stability condition is a requirement solely on the numerical scheme and does not involve any condition on the differential equation.

3 Von Neumann stability analysis

The key innovation of the von Neumann stability analysis is to decompose errors with complex Fourier series. To make the Fourier series applicable, it's better to have a periodic function. Thus, we assume periodic boundary

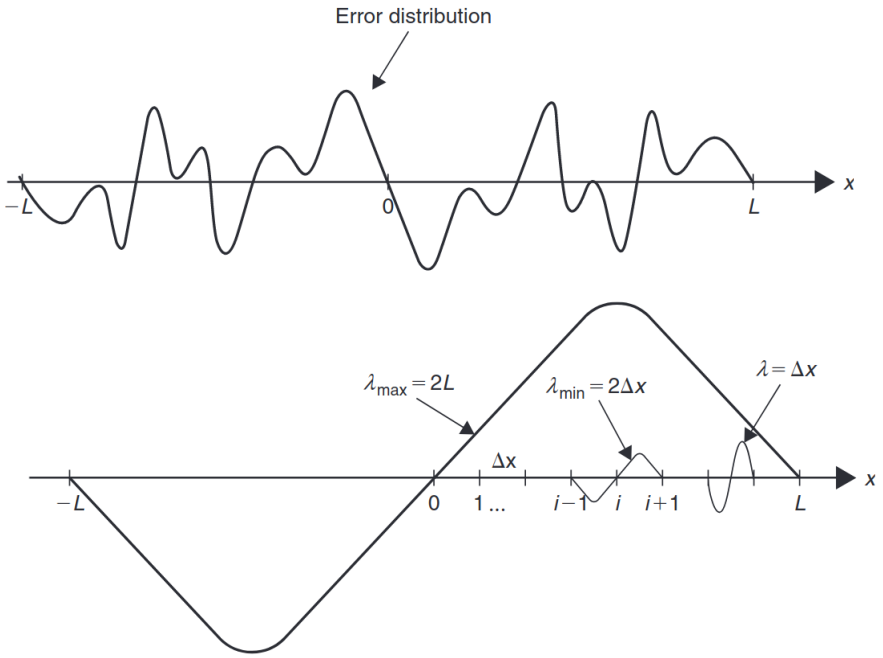


Figure 2: Errors and Fourier decomposition

conditions for the computational domain. Under this assumption, the error function can be decomposed as:

$$\varepsilon(x,t) = \sum_{m=-\infty}^{\infty} v_m(t) e^{ik_m x}$$

where the errors are decomposed as the summation of waves $e^{ik_m x}$ with wave number k_m, the magnitude of which is a function of time $v_m(t)$, as shown by Fig (2). The wave number is given by $k_m = \frac{2\pi}{\lambda_m}$, where λ_m is the wave length. For domain length L discretized by nodes x_i, $i = 0, 1, 2, \cdots, N$, which leads to $\Delta x = \frac{L}{N}$, the possible wave lengths are $\lambda_m = m2\Delta x$, $m = 1, 2, \cdots, N$ (Due to the mesh resolution, $\lambda = \Delta x$ gives the same constant on all node points, thus losing the wave-like behavior). The resulting wave numbers are $k_m = m\frac{\pi}{N\Delta x}$, $m = 1, 2, \cdots, N$. To further facilitate the analysis, the wave numbers can be transformed to phase angle: $\theta = k_m \Delta x = \frac{m\pi}{N}$, where the region close to $\theta = 0$ corresponds to the low frequency waves while the region close to $\theta = \pi$ is associated with high frequency waves. With the definition of phase angle, the stability of a discretization scheme can be characterized by the amplification factors of errors in different phase angles regardless of the domain length etc.

Complex Fourier series: The complex Fourier series is just a complex expression for the real Fourier series. The real Fourier series can be expressed

as:

$$\varepsilon(x) = \frac{a_0}{2} + \sum_{m=1}^{\infty} (a_m \cos(k_m x) + b_m \sin(k_m x))$$

Substituting the Euler's equation:

$$e^{i\theta} = \cos\theta + i\sin\theta$$
$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2},\ \sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}.$$

The resultant equation is the complex Fourier series:

$$\varepsilon(x) = \sum_{m=-\infty}^{\infty} v_m e^{ik_m x}$$

In other words, although imaginary number shows up in the decomposition, the final outcome after the superposition of all the waves will be a pure real wave existing in physical world. Note that

$$v_m = \begin{cases} \frac{1}{2}a_m - \frac{i}{2}b_m \text{ for } m \geq 1 \\ \frac{1}{2}a_0 \text{ for } m = 0 \\ \frac{1}{2}a_{|m|} + \frac{i}{2}b_{|m|} \text{ for } m \leq -1 \end{cases}$$

we observe that the phase and magnitude information given by the pairs of a_m, b_m in the real Fourier series:

$$a_m \cos(k_m x) + b_m \sin(k_m x) = \sqrt{a_m^2 + b_m^2}\cos\big(k_m x - \arctan(b/a)\big)$$

now is transferred to v_m and v_{-m} in the complex Fourier series. The frequency information is kept by k_m.

3.1 1D Diffusion Problem

We now apply the decomposition in the error analysis of the 1D diffusion equation:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} + f$$

where $\nu > 0$ is the diffusion coefficient. Applying the central difference scheme for spatial discretization and the forward Euler scheme for temporal discretization, we get the following FDE:

$$\overline{u}_i^{n+1} = \overline{u}_i^n + \frac{\nu\Delta t}{\Delta x^2}(\overline{u}_{i+1}^n - 2\overline{u}_i^n + \overline{u}_{i-1}^n) + f^n.$$

However, the round-off error will be introduced at the RHS of the equation, which leads to the calculated u_i^{n+1} as:

$$u_i^{n+1} = \overline{u}_i^n + \frac{\nu\Delta t}{\Delta x^2}(\overline{u}_{i+1}^n - 2\overline{u}_i^n + \overline{u}_{i-1}^n) + \overline{\varepsilon}_i^n + \frac{\nu\Delta t}{\Delta x^2}(\overline{\varepsilon}_{i+1}^n - 2\overline{\varepsilon}_i^n + \overline{\varepsilon}_{i-1}^n) + f^n$$

Thus, the error at $n+1$ accumulated from the previous time step can be calculated as:

$$\bar{\varepsilon}_i^{n+1} = u_i^{n+1} - \bar{u}_i^{n+1} = \bar{\varepsilon}_i^n + \frac{\nu \Delta t}{\Delta x^2}(\bar{\varepsilon}_{i+1}^n - 2\bar{\varepsilon}_i^n + \bar{\varepsilon}_{i-1}^n)$$

which shares the same FDE with the variable. Note that this is only applicable for linear PDE with constant coefficients. At this point, we decompose the error with Fourier series:

$$\sum_{m=-\infty}^{\infty} v_m^{n+1} e^{ik_m x} = \sum_{m=-\infty}^{\infty} \left(v_m^n e^{ik_m x} + \frac{\nu \Delta t}{\Delta x^2}(v_m^n e^{ik_m(x+\Delta x)} - 2v_m^n e^{ik_m x} + v_m^n e^{ik_m(x-\Delta x)}) \right)$$

where the superscript of v^n denotes the value of v at n time step. As a result of linear PDE with constant coefficients, the magnitudes of waves at each wave number evolve in the same manner. It is sufficient to analyse the magnitude evolution of wave at a single wave number:

$$v_m^{n+1} e^{ik_m x} = \left(v_m^n e^{ik_m x} + \frac{\nu \Delta t}{\Delta x^2}(v_m^n e^{ik_m(x+\Delta x)} - 2v_m^n e^{ik_m x} + v_m^n e^{ik_m(x-\Delta x)}) \right),$$

where m can be arbitrary integer. Now we can calculate the amplification of the error from the previous time step to the current time step:

$$\begin{aligned} G_m = \frac{v_m^{n+1}}{v_m^n} &= 1 + \frac{\nu \Delta t}{\Delta x^2}\left(e^{ik_m \Delta x} - 2 + e^{-ik_m \Delta x} \right) \\ &= 1 + \frac{2\nu \Delta t}{\Delta x^2}\left(\cos(k_m \Delta x) - 1 \right) \\ &= 1 - \frac{4\nu \Delta t}{\Delta x^2}\left(\sin(k_m \Delta x/2)^2 \right) \end{aligned}$$

$$\left[\cos(\theta) = \cos^2(\theta/2) - \sin^2(\theta/2)), \quad 1 = \cos^2(\theta/2) + \sin^2(\theta/2) \right]$$

The condition for the discretization scheme to be stable is that errors in all frequencies are not allowed to be amplified:

$$|G_m| \leq 1, \forall m$$

$$\begin{aligned} \left| 1 - \frac{4\nu \Delta t}{\Delta x^2}\left(\sin(k_m \Delta x/2)^2 \right) \right| &\leq 1 \\ -2 \leq -\frac{4\nu \Delta t}{\Delta x^2}\left(\sin(k_m \Delta x/2)^2 \right) &\leq 0 \\ \frac{\nu \Delta t}{\Delta x^2} &\leq \frac{1}{2} \end{aligned}$$

3.2 1D Convection Problem

Forward time central space (FTCS)

Similarly, consider the 1D convection problem:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

Using the central difference method for the spatial discretization and the forward Euler method for temporal discretization:

$$\overline{u}_i^{n+1} = \overline{u}_i^n - \frac{c}{2\Delta x}(\overline{u}_{i+1}^n - \overline{u}_{i-1}^n)$$

The error shares the same FDE:

$$\overline{\varepsilon}_i^{n+1} = \overline{\varepsilon}_i^n - \frac{c}{2\Delta x}(\overline{\varepsilon}_{i+1}^n - \overline{\varepsilon}_{i-1}^n)$$

After Fourier series decomposition:

$$v_m^{n+1} = v_m^n - \frac{c}{2\Delta x}(v_m^n e^{ik_m\Delta x} - v_m^n e^{-ik_m\Delta x})$$

The amplification factor can be calculated as:

$$G_m = 1 - \frac{c}{\Delta x} i \sin(k_m \Delta x)$$

Note that for pure convection case, G_m is a complex number. The stability condition is given by:

$$|G_m| \leq 1 \quad \forall m$$

where $|G_m| = \sqrt{Re(G_m)^2 + Im(G_m)^2}$ denotes the norm of the complex number and $Re(\cdot)$ and $Im(\cdot)$ denotes the real and imaginary part of the complex number. Thus, we have

$$|G_m| = \sqrt{1 + (\frac{c}{\Delta x}\sin(k_m\Delta x))^2} > 1$$

the scheme is unconditionally unstable.

Lax method

To solve the convection equation with a stable scheme, we can employ the Lax method. The corresponding FDE is given by:

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

The Fourier series decomposition leads to:

$$v_m^{n+1} = v_m^n \left(\frac{1}{2}(e^{ik_m\Delta x} + e^{-ik_m\Delta x}) - \frac{c}{2}\frac{\Delta t}{\Delta x}(e^{ik_m\Delta x} + e^{-ik_m\Delta x})\right)$$

$$G = \cos(k_m\Delta x) - i\sin(k_m\Delta x)\frac{c\Delta t}{\Delta x}$$

The stability condition is given by:

$$|G| = \sqrt{\cos^2(k_m\Delta x) + \sin^2(k_m\Delta x) + \left(\left(\frac{c\Delta t}{\Delta x}\right)^2 - 1\right)\sin^2(k_m\Delta x)}$$

$$= \sqrt{1 + \left(\left(\frac{c\Delta t}{\Delta x}\right)^2 - 1\right)\sin^2(k_m\Delta x)} \leq 1$$

$$\left(\frac{c\Delta t}{\Delta x}\right)^2 \leq 1$$

Since Δt and Δx are positive values, the stability condition is given by:

$$\frac{|c|\Delta t}{\Delta x} < 1$$

This is called the Courant criterion or Courant-Friedrich-Levy (CFL) number. In one dimension, it can be explained as that (1) the domain of dependence of the differential equation should be entirely contained in the numerical domain of dependence of the discretized equations. Or, (2) the information at x_j is only allowed to propagate in the discrete intervals between $[x_j, x_j + 1]$ when $c > 0$ or $[x_{j-1}, x_j]$ when $c < 0$. Propagation beyond the discrete interval or skipping a computational node is not allowed. These two statements are illustrated in Fig. (3).

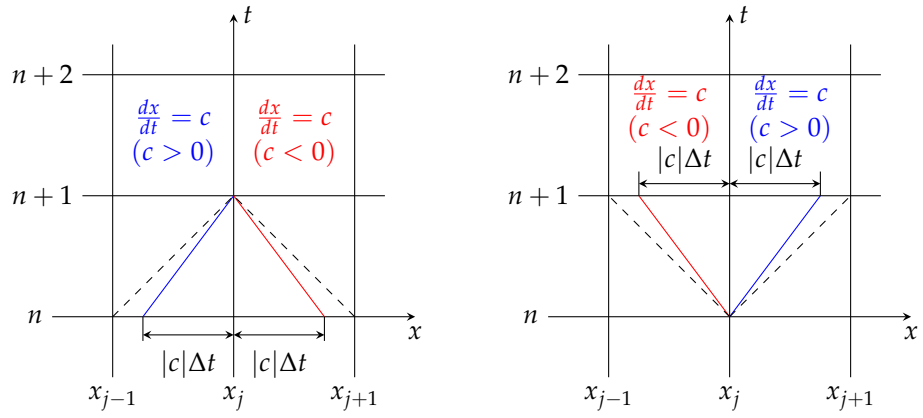


Figure 3: Illustration of the CFL condition statement one (left) and two (right)

3.3 2D Diffusion Problem

The von Neumann analysis is also applicable for higher dimension cases. Consider the 2D heat equation:

$$\frac{\partial T}{\partial t} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right)$$

Using the central difference method for spatial discretization and the forward Euler method for temporal discretization, we have the FDE as:

$$T_{i,j}^{n+1} = T_{i,j}^n + \frac{\alpha\Delta t}{h^2}\left(T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n - 4T_{i,j}^n\right)$$

where $h = \Delta x = \Delta y$ in uniform mesh, $\alpha > 0$ is the diffusion coefficient. Using Fourier expansion:

$$T_{i,j}^n = v_{m,n}^n e^{ik_m x_i} e^{ik_n y_i}$$

We have

$$v_{m,n}^{n+1} = v_{m,n}^{n}\left(1 + \frac{\alpha\Delta t}{h^2}\left(e^{ik_m h} + e^{-ik_m h} + e^{ik_n h} + e^{-ik_n h} - 4\right)\right)$$

The amplification factor can be calculated as:

$$\begin{aligned} G_{m,n} = \frac{v_{m,n}^{n+1}}{v_{m,n}^{n}} &= \left(1 + \frac{\alpha\Delta t}{h^2}\left(2\cos(k_m h) + 2\cos(k_n h) - 4\right)\right) \\ &= \left(1 + \frac{\alpha\Delta t}{h^2}\left(-4\left(\sin^2(k_m h/2) + \sin^2(k_n h/2)\right)\right)\right) \end{aligned}$$

According to the stable condition, the worst case is:

$$\begin{aligned} -1 \leq 1 - \frac{8\alpha\Delta t}{h^2} &\leq 1 \\ \frac{\alpha\Delta t}{h^2} &\leq \frac{1}{4} \end{aligned}$$

To summarize, with the assumption of *homogeneous linear PDE with constant coefficients and periodic boundary condition*, the von Neumann analysis can help us to analyze the stability of *sptial-temporal* discretization scheme in *one-dimensional or multi-dimansional* cases.

4 Modified equation analysis

In previous sections, we demonstrated that our numerical schemes only approximate the exact PDE we are trying to solve. Furthermore, via von Neumann analysis we can quantify the type of error, we will observe as either dissipation or dispersion. Modified equation analysis is another powerful technique that allows us to better understand how and why a particular numerical scheme behaves the way it does. Using modified equation analysis we can show that, while our numerical scheme does not exactly satisfy the PDE we are trying to solve, it does provide an exact solution to a similar PDE. By determining what this similar PDE is, we can gain further insight into the behavior of our scheme.

For example, given a FDE:

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

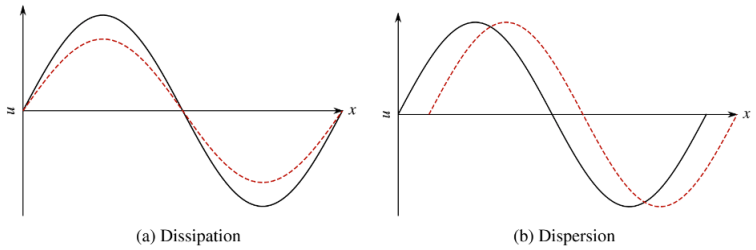
To find the corresponding PDE, we need to introduce the differential operators via Taylor expansion:

(1) $u_j^{n+1} = u_j^n + \frac{\partial u_j^n}{\partial t}\Delta t + \frac{\partial^2 u_j^n}{\partial t^2}\frac{\Delta t^2}{2} + \frac{\partial^3 u_j^n}{\partial t^3}\frac{\Delta t^3}{6} + \cdots$

(2) $u_j^{n-1} = u_j^n - \frac{\partial u_j^n}{\partial t}\Delta t + \frac{\partial^2 u_j^n}{\partial t^2}\frac{\Delta t^2}{2} - \frac{\partial^3 u_j^n}{\partial t^3}\frac{\Delta t^3}{6} + \cdots$

(3) $u_{j+1}^n = u_j^n + \frac{\partial u_j^n}{\partial x}h + \frac{\partial^2 u_j^n}{\partial x^2}\frac{h^2}{2} + \frac{\partial^3 u_j^n}{\partial x^3}\frac{h^3}{6} + \cdots$

(4) $u_{j-1}^n = u_j^n - \frac{\partial u_j^n}{\partial x}h + \frac{\partial^2 u_j^n}{\partial x^2}\frac{h^2}{2} - \frac{\partial^3 u_j^n}{\partial x^3}\frac{h^3}{6} + \cdots$



(a) Dissipation (b) Dispersion

Figure 4: Illustration of dissipation and dispersion errors

Substituting them into the FDE, we have:

$$\frac{\partial u}{\partial t} + \frac{\partial^3 u}{\partial t^3}\frac{\Delta t^2}{6} = D\frac{\partial^2 u}{\partial x^2} + D\frac{\partial^4 u}{\partial x^4}\frac{h^2}{12} + \dots$$

Rearranging the terms:

$$\frac{\partial u}{\partial t} - D\frac{\partial^2 u}{\partial x^2} = D\frac{\partial^4 u}{\partial x^4}\frac{h^2}{12} - \frac{\partial^3 u}{\partial t^3}\frac{\Delta t^2}{6} + \dots$$

The resulting equation shows that the scheme solves the diffusion equation, with errors of $O(h^2, \Delta t^2)$. Besides, the leading order errors behave in the form of $\frac{\partial^4 u}{\partial x^4}$ and $\frac{\partial^3 u}{\partial t^3}$.

In CFD the numerical error introduced by a scheme is typically classified into two general types, dissipation error and dispersion error. The dissipation error lets the amplitude of the solution remain the same or be reduced after each time step, while the dispersion error creates out-of-phase with the exact solution. Considering the errors in the equations, the former has an even-order derivative, then one would expect that the scheme is dominantly dissipative, which means that the amplitude of the wave will decrease in space. If the latter has an odd-order derivative then one would expect that the scheme is dominantly dispersive. Hence a phase shift will happen in time.

Similar to von Neumann analysis, modified equation analysis is a powerful tool to aid in understanding the general behavior of a numerical scheme, elucidate its dissipative and dispersive error properties, and identify its stability limits.

5 Temporal discretization

Now we turn our attention to the temporal discretization. Broadly speaking, the temporal discretization scheme can be classified into two categories: explicit methods and implicit methods. In explicit methods, only the data in the previous time step is used in the calculation. Mathematically, for a given PDE

$$\frac{\partial u}{\partial t} = f(t, u, \partial_x u, \cdots)$$

The analytical expression for the temporal evolution can be written as:

$$u(t^{n+1}) = u(t^n) + \int_{t^n}^{t^{n+1}} f(\tau, u(\tau), \partial_x u(\tau), \cdots) d\tau$$

In explicit methods, the time derivative $f(\tau, u(\tau), \partial_x u(\tau), \cdots)$ is evaluated according to the known data at the previous time step n. On the contrary, the evaluation of $f(\tau, u(\tau), \partial_x u(\tau), \cdots)$ in implicit methods include at least one value evaluated in the current time step $n+1$. Now taking the diffusion equation as an example and employing central difference scheme for the spatial discretization, we give several examples of temporal discretization in each category, and use von Neumann analysis to check their stability.

5.1 Explicit method

Forward Euler

The scheme is given by:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \frac{\alpha}{\Delta x^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)$$

which leads to the following evolution equation:

$$T_j^{n+1} = T_j^n + \frac{\Delta t \alpha}{\Delta x^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)$$

The corresponding modified equation is:

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = \frac{\alpha h^2}{12}(1 - 6s) T_{xxxx} + O\left(\Delta^2, h^2 \Delta t, h^4\right) T_{xxxxx}$$

where $s = \frac{\alpha \Delta t}{h^2}$. The accuracy is $O(\Delta t, h^2)$. When $s = 1/6$, the accuracy is $O(\Delta t^2, h^2)$. The amplification factor given by the von Neumann analysis is:

$$\left|\frac{v^{n+1}}{v^n}\right| = \left|1 - 4\frac{\alpha \Delta t}{h^2} \sin^2 k\frac{h}{2}\right| \leq 1$$

$$0 \leq \frac{\alpha \Delta t}{h^2} \leq \frac{1}{2}$$

As shown in Fig. (5), the calculation of the forward Euler scheme is simple and straight forward. However, due to the locality of the evolution, the dependency of the solution in the interior domain on the boundary values are quite lose, which may cause unphysical solution behavior and leads to instability when coarse grid is used.

Example of instability building up

Consider a 2D diffusion equation with boundary condition of $T_0 = 1, T_3 = 0$ and Initial condition $T_1^0 = 1, = T_2^0 = 0$.

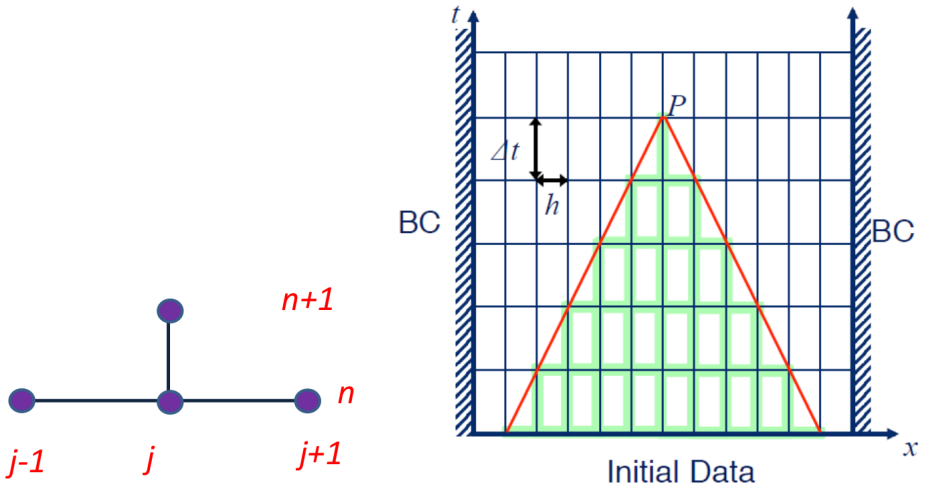


Figure 5: Forward Euler calculation for single node (left) and entire grid (right)

Considering FTCS scheme, we have:

$$T_1^1 = T_1^0 + \frac{\alpha \Delta t}{h^2}\left(T_0^0 - 2T_1^0 + T_2^0\right) = 1 - \frac{\alpha \Delta t}{h^2}$$
$$T_2^1 = T_2^0 + \frac{\alpha \Delta t}{h^2}\left(T_1^0 - 2T_2^0 + T_3^0\right) = \frac{\alpha \Delta t}{h^2}$$

When $\alpha \Delta t / h^2 = 0.2 < 0.5$, the scheme is stable. When $\alpha \Delta t / h^2 = 0.7 > 0.5$, we can observe that the result becomes oscillating and unphysical which violates the second law of thermodynamics (heat can never autonomously pass from low temperature to high temperature). The results are plotted in Fig. (6).

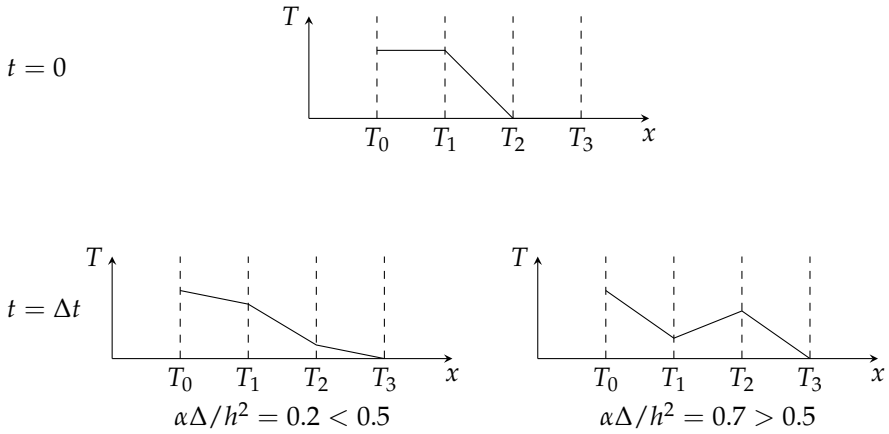


Figure 6: Building up of instability

5.2 Implicit method

Backward Euler

The backward Euler scheme is given by:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \frac{\alpha}{h^2}\left(T_{j+1}^{n+1} - 2T_j^{n+1} + T_{j-1}^{n+1}\right)$$

which can be rearranged as: $-sT_{j+1}^{n+1} + (2s+1)T_j^{n+1} - sT_{j-1}^{n+1} = T_j^n \quad \left(s = \frac{\alpha\Delta t}{h^2}\right)$
According to the equation, inversion of a tri-diagonal matrix is required to evolve solution from t^n to t^{n+1}. Compared to the explicit method, this requires more computational cost. On the other hand, this leads to stronger dependency between the solutions in the adjacent time step, thus being more stable. This can be shown by the von Neumann analysis. The amplification factor can be calculated as:

$$|G| = \left|\frac{v^{n+1}}{v^n}\right| = \left|\frac{1}{1 + 2s(1-\cos\theta)}\right| \leq 1$$

which is unconditional stable. The modified equation is given by:

$$\frac{\partial T}{\partial t} - \alpha\frac{\partial^2 T}{\partial x^2} = \frac{\alpha h^2}{12}(1+6s)T_{xxx} + O\left(\Delta t^2, h^2\Delta t, h^4\right)T_{xxxxx}$$

Crank-Nicolson scheme/Trapezoidal method

The Crank-Nicolson scheme is given by:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\alpha}{2}\left[\left(\frac{T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}}{h^2}\right) + \left(\frac{T_{i+1}^{n} - 2T_i^{n} + T_{i-1}^{n}}{h^2}\right)\right]$$

The corresponding evolution equation is:

$$-rT_{i+1}^{n+1} + (2r+1)T_i^{n+1} - rT_{i-1}^{n+1} = rT_{i+1}^n + (1-2r)T_i^n + rT_{i-1}^n$$

where $r = \frac{\alpha\Delta t}{2h^2}$. The Crank-Nicolson scheme makes use of trapezoidal differencing to achieve second-order accuracy with a T.E. of $O\left[(\Delta t)^2, (h)^2\right]$. The modified equation is:

$$\frac{\partial T}{\partial t} - \alpha\frac{\partial^2 T}{\partial x^2} = \frac{\alpha h^2}{12}T_{xxxx} + \left[\frac{1}{12}\alpha^3\Delta t^2 + \frac{1}{360}\alpha h^4\right]T_{xxxxxx}$$

with the error of $O(\Delta t^2, h^2)$. The amplification factor is

$$|G| = \left|\frac{v^{n+1}}{v^n}\right| = \left|\frac{1 - r(1-\cos\theta)}{1 + r(1-\cos\theta)}\right|$$

which is unconditionally stable.

θ scheme

Using weighting θ to combine the Forward Euler and Backward Euler method, we get the θ scheme:

$$\frac{T_i^{n+1}-T_i^n}{\Delta t} = \alpha\left[\theta\left(\frac{T_{i+1}^{n+1}-2T_i^{n+1}+T_{i-1}^{n+1}}{h^2}\right)+(1-\theta)\left(\frac{T_{i+1}^{n}-2T_i^{n}+T_{i-1}^{n}}{h^2}\right)\right]$$

$$\theta = \begin{cases} 0 & \text{Explicit (FTCS)} \\ 1 & \text{Implicit} \\ 1/2 & \text{Crank-Nicolson} \end{cases}$$

The modified equation is given by:

$$\begin{aligned}\frac{\partial T}{\partial t} - \alpha\frac{\partial^2 T}{\partial x^2} &= \left[\left(\theta-\tfrac{1}{2}\right)\alpha^2\Delta t + \tfrac{\alpha h^2}{12}\right]T_{xxx} \\ &+ \left[\left(\theta^2-\theta+\tfrac{1}{3}\right)\alpha^3\Delta t^2 + \tfrac{1}{6}\left(\theta-\tfrac{1}{2}\right) + \tfrac{1}{360}\alpha h^4\right]T_{xxxxx}\end{aligned}$$

when $\theta = \frac{1}{2} - \frac{r}{12}$, the error is $O(\Delta t^2, h^4)$. The scheme is unconditionally stable when $\frac{1}{2} \leq \theta \leq 1$

2D FTCS and Crank-Nicolson scheme

For 2D heat equation, the FTCS scheme is given by:

$$T_{i,j}^{n+1} = T_{i,j}^n + \frac{\alpha\Delta t}{h^2}\left(T_{i+1,j}^n + T_{i-1,j}^n + T_{i,j+1}^n + T_{i,j-1}^n - 4T_{i,j}^n\right)$$

The accuracy is $O(t, h^2)$. The stability condition can be calculated using Fourier expansion:

$$\begin{aligned} v^{n+1} &= v^n + \frac{\alpha\Delta t}{h^2}\left(e^{ik_x h} + e^{-ik_k h} + e^{ik_y h} + e^{-iky h} - 4\right)v^n \\ \frac{v^{n+1}}{v^n} &= 1 + \frac{\alpha\Delta t}{h^2}\left(2\cos k_x h + 2\cos k_y h - 4\right) \\ \frac{v^{n+1}}{v^n} &= 1 - \frac{4\alpha\Delta t}{h^2}\left(\sin^2\frac{k_x h}{2} + \sin^2\frac{k_y h}{2}\right)\end{aligned}$$

Consider the worst case scenario:

$$\begin{aligned} &-1 \leq 1 - \tfrac{8\alpha\Delta t}{h^2} \leq 1 \\ &\Rightarrow \tfrac{\alpha\Delta t}{h^2} \leq \tfrac{1}{4}\end{aligned}$$

Note that the stability condition of FTCS scheme in 1D and 2D is different. The stability condition for FTCS of 1D, 2D and 3D heat equation are given by $\frac{\alpha\Delta t}{h^2} \leq \frac{1}{2}, \frac{\alpha\Delta t}{h^2} \leq \frac{1}{4}$ and $\frac{\alpha\Delta t}{h^2} \leq \frac{1}{6}$ respectively.

$$\frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} = \frac{\alpha}{2}\left(\frac{\partial^2 T^{n+1}}{\partial x^2} + \frac{\partial^2 T^{n+1}}{\partial y^2} + \frac{\partial^2 T^n}{\partial x^2} + \frac{\partial^2 T^n}{\partial y^2}\right)$$

Taking $x = i\Delta x, y = j\Delta y, \Delta x = \Delta y = h, i = 1, 2, \cdots, m, j = 1, 2, \cdots, n$

$$T_{i,j}^{n+1} = T_{i,j}^{n} + \frac{\alpha\Delta t}{2h^2}\left(T_{i+1,j}^{n+1} + T_{i-1,j}^{n+1} + T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1} - 4T_{i,j}^{n+1}\right) + \frac{\alpha\Delta t}{2h^2}\left(T_{i+1,j}^{n} + T_{i-1,j}^{n} + T_{i,j+1}^{n} + T_{i,j-1}^{n} - 4T_{i,j}^{n}\right)$$

Note that in 2D, inversion of a matrix with number of entries $(nm)^2$ is required.

5.3 Alternating Direction Implicit (ADI)

Using the implicit methods, a system of equations requires substantially more computer time to solve, which is often solved by iterative methods. Can we avoid iterative solvers when attempting to solve the 2-D heat equation? This led to the development of alternating-direction implicit (ADI) methods by Peaceman and Rachford (1955) and Douglas (1955).

The 2D Alternating direction implicit scheme is composed of two steps of the backward Euler method, with each step taking the future information from one direction: Step 1 (X-direction):

$$T_{i,j}^{n+1/2} - T_{i,j}^{n} = \frac{\alpha\Delta t}{2h^2}\left[\left(T_{i+1,j}^{n+1/2} - 2T_{i,j}^{n+1/2} + T_{i-1,j}^{n+1/2}\right) + \left(T_{i,j+1}^{n} - 2T_{i,j}^{n} + T_{i,j-1}^{n}\right)\right]$$

Step 2 (Y-direction):

$$T_{i,j}^{n+1} - T_{i,j}^{n+1/2} = \frac{\alpha\Delta t}{2h^2}\left[\left(T_{i+1,j}^{n+1/2} - 2T_{i,j}^{n+1/2} + T_{i-1,j}^{n+1/2}\right) + \left(T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1}\right)\right]$$

To avoid bias, the sequence of step 1 and step 2 can be alternated. The error of the scheme is $O(\Delta t^2, h^2)$.

Now, the 2D problem is decomposed as multiple 1D problems (n 1D problems in step 1 and m 1D problems in step 2). Instead of inverting a matrix with a number of entries $(mn)^2$ as a 2D problem, we now need to invert a matrix with m^2 entries as a 1D problem for n times in step one, and invert a matrix with n^2 entries as a 1D problem for m times in step two, which significantly reduces the computational cost. Besides, the stability condition can be relaxed by reducing the dimension from 2D to 1D (see 5.2 2D FTCS). Different time steps can be applied in different directions according to the scheme of each direction

To analyze the scheme, the two steps can be combined as:

$$T_{i,j}^{n+1} - T_{i,j}^{n} = \frac{\alpha\Delta t}{2}\left[\underbrace{\frac{\partial^2 T^{n+1/2}}{\partial x^2}}_{\text{Mid-point}} + \underbrace{\frac{1}{2}\left(\frac{\partial^2 T^{n+1}}{\partial y^2} + \frac{\partial^2 T^{n}}{\partial y^2}\right)}_{\text{Trapezoidal}}\right]$$

Using Fourier expansion:

$$v^{n+1/2} = v^n + \frac{\alpha\Delta t}{h^2}\left[v^{n+1/2}\left(e^{ik_x h} + e^{-ik_2 h} - 2\right) + v^n\left(e^{ik_x h} + e^{-ik_x h} - 2\right)\right]$$

$$\frac{v^{n+1/2}}{v^n} = \frac{1 - 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_x h}{2}}{1 + 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_y h}{2}}$$

Similarly

$$\frac{v^{n+1}}{v^{n+1/2}} = \frac{1 - 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_y h}{2}}{1 + 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_x h}{2}}$$

Combining

$$\frac{v^{n+1}}{v^n} = \left(\frac{1 - 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_x h}{2}}{1 + 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_y h}{2}}\right)\left(\frac{1 - 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_y h}{2}}{1 + 2\frac{\alpha\Delta t}{h^2}\sin^2\frac{k_x h}{2}}\right) < 1$$

The scheme is unconditionally stable.

5.4 Approximate Factorization splitting

Consider Crank-Nicolson scheme for temporal discretization and central difference scheme for spatial discretization, the FDM for 2D heat euqation is given by:

$$\begin{aligned}\frac{T^{n+1} - T^n}{\Delta t} = &\frac{\alpha}{2\Delta x^2}\left(T^{n+1}_{i+1,j} + T^{n+1}_{i-1,j} - 2T^{n+1}_{i,j}\right) + \frac{\alpha}{2\Delta y^2}\left(T^{n+1}_{i,j+1} + T^{n+1}_{i,j-1} - 2T^{n+1}_{i,j}\right)\\ &+ \frac{\alpha}{2\Delta x^2}\left(T^{n}_{i+1,j} + T^{n}_{i-1,j} - 2T^{n}_{i,j}\right) + \frac{\alpha}{2\Delta y^2}\left(T^{n}_{i,j+1} + T^{n}_{i,j-1} - 2T^{n}_{i,j}\right)\\ &+ \mathrm{O}\left(\Delta t^2, \Delta x^2, \Delta y^2\right)\end{aligned}$$

Defining $\delta_{xx}() = \frac{1}{\Delta x^2}\left[()_{i-1,j} - 2()_{i,j} + ()_{i+1,j}\right]$, $\delta_{yy}() = \frac{1}{\Delta y^2}\left[()_{i,j-1} - 2()_{i,j} + ()_{i,j+1}\right]$, the Crank-Nicolson for heat equation becomes:

$$\begin{aligned}\frac{T^{n+1} - T^n}{\Delta t} =&\frac{\alpha}{2}\delta_{xx}\left(T^{n+1} + T^n\right) + \frac{\alpha}{2}\delta_{yy}\left(T^{n+1} + T^n\right)\\ &+ \mathrm{O}\left(\Delta t^2, \Delta x^2, \Delta y^2\right)\end{aligned}$$

which can be written as

$$\begin{aligned}\left(1 - \frac{\alpha\Delta t}{2}\delta_{xx} - \frac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n+1} = &\left(1 + \frac{\alpha\Delta t}{2}\delta_{xx} + \frac{\alpha\Delta t}{2}\delta_{yy}\right)T^n\\ &+ \Delta t\mathrm{O}\left(\Delta t^2, \Delta x^2, \Delta y^2\right)\end{aligned}$$

Noticing that $\left(1 - \frac{\alpha\Delta t}{2}\delta_{xx} - \frac{\alpha\Delta t}{2}\delta_{yy}\right)$ and $\left(1 + \frac{\alpha\Delta t}{2}\delta_{xx} + \frac{\alpha\Delta t}{2}\delta_{yy}\right)$ are linear

operator, we factorize them to reduce the complexity:

$$\left(1-\frac{\alpha\Delta t}{2}\delta_{xx}-\frac{\alpha\Delta t}{2}\delta_{yy}\right)=\left(1-\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left(1-\frac{\alpha\Delta t}{2}\delta_{yy}\right)-\left(\frac{\alpha\Delta t}{2}\right)^2\delta_{xx}\delta_{yy}$$
$$\left(1+\frac{\alpha\Delta t}{2}\delta_{xx}+\frac{\alpha\Delta t}{2}\delta_{yy}\right)=\left(1+\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left(1+\frac{\alpha\Delta t}{2}\delta_{yy}\right)-\left(\frac{\alpha\Delta t}{2}\right)^2\delta_{xx}\delta_{yy}$$

Note that by factorization, the original operator involving both X and Y directions now is split into two sequential operators which only contain X or Y direction. Similar to the ADI method, the factorization changes the 1-step scheme for the 2D problem into 2 steps scheme for the 1D problem, the advantage of which has been discussed in the ADI section. This is more clear in the resulting scheme:

$$\begin{aligned}&\left(1-\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left(1-\frac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n+1}-\left(\frac{\alpha\Delta t}{2}\right)^2\delta_{xx}\delta_{yy}T^{n+1}\\&=\left(1+\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left(1+\frac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n}-\left(\frac{\alpha\Delta t}{2}\right)^2\delta_{xx}\delta_{yy}T^{n}\\&+\Delta t\mathrm{O}\left(\Delta t^2,\Delta x^2,\Delta y^2\right)\end{aligned}$$

Or

$$\begin{aligned}&\left(1-\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left[\left(1-\frac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n+1}\right]\\&=\left(1+\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left[\left(1+\frac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n}\right]+\left(\frac{\alpha\Delta t}{2}\right)^2\delta_{xx}\delta_{yy}(T^{n+1}-T^{n})\\&+\Delta t\mathrm{O}\left(\Delta t^2,\Delta x^2,\Delta y^2\right)\end{aligned}$$

In fact, the ADI method can be expressed as

$$\begin{aligned}\left(1-\tfrac{\alpha\Delta t}{2}\delta_{xx}\right)T^{n+1/2}&=\left(1+\tfrac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n}\\\left(1-\tfrac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n+1}&=\left(1+\tfrac{\alpha\Delta t}{2}\delta_{xx}\right)T^{n+1/2}\end{aligned}$$

Eliminating $T^{n+1/2}$

$$\left(1-\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left(1-\frac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n+1}=\left(1+\frac{\alpha\Delta t}{2}\delta_{xx}\right)\left(1+\frac{\alpha\Delta t}{2}\delta_{yy}\right)T^{n}$$

Upto a factor:

$$+\frac{\alpha^2\Delta t^2}{4}\delta_{xx}\delta_{yy}\left(T^{n+1}-T^{n}\right)\approx\left(\frac{\alpha}{2}\right)^2\Delta t^3\delta_{xx}\delta_{yy}\frac{\partial T}{\partial t}=\mathrm{O}\left(\Delta t^3\right)$$

ADI is an approximate factorization of the Crank-Nicolson method