Implicit RD Method for Unsteady Advection Problem

Group of Computational Fluid Dynamics
Universiti Sains Malaysia
Presenter : Neoh Soon Sien
18th November 2015.

Implicit RD Method for Unsteady Advection Problem

- **The Consistent Upwind Mass Matrix**
  - Semi-discrete Finite Difference Equation
  - Implicit solver
  - The mass matrix for Implicit Solver

Consistent Upwind Mass Matrix

- Semi-discrete Finite Difference Equation
- Implicit solver
- The mass matrix for Implicit Solver

Semi-discrete Finite Difference Equation

In the method of lines (semi-discrete method), both the spatial and temporal discretisation are separable.

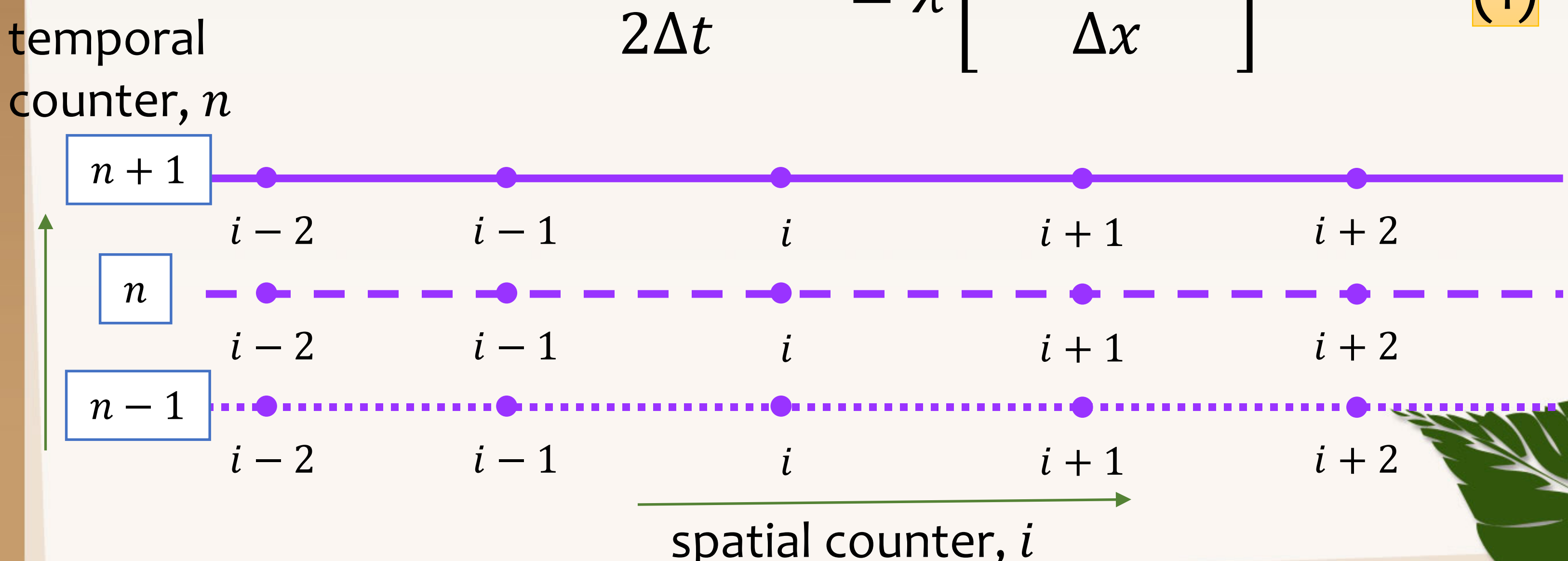
The spatial derivative is to be discretised at a chosen time slab, and only after that the time derivative is considered.

This leads to a system of coupled ordinary differential equation.

Take 1D advection model equation as an example, by using:
forward-differencing scheme for spatial derivative
three-point central-differencing scheme for time derivative

$$\frac{\partial u}{\partial t} = \lambda \frac{\partial u}{\partial x}$$

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \lambda \left[\frac{u_{i+1}^n - u_i^n}{\Delta x} \right] \quad (1)$$



Implicit Solver

For unsteady scalar advection problem, a more general equation to represent the problem is

$$\frac{\partial u}{\partial t} = u_t = F(u, \vec{x}, t)$$

where F is a function of the dependent variable u and two independent variables $\vec{x}$ and t.

By introducing the n superscript that refers to a discrete time level,

$$(u_t)^n = F(u^n, \vec{x}^n, t^n)$$

After the discretisation of the equation (using the above mentioned semi-discrete or method of lines method), the PDE is reduced to ODE.

The discretised ODE usually forms linear combinations of the dependent variable and its derivative at various time steps.

$$u^{n+1} = f(a_1 u_t^{n+1}, a_0 u_t^n, a_{-1} u_t^{n-1}, \cdots, b_1 u^{n+1}, b_0 u^n, b_{-1} u^{n-1}, \cdots)$$

If u^{n+1} is to be determined, the method is said to be explicit if

$$a_1 = 0 \quad \&\& \quad b_1 = 0$$

and implicit otherwise.

u^n means the flow solution at the present time t.
Consequently, u^{n+1} represents the solution at the time $t + \Delta t$

Explicit means u^{n+1} depends solely on values already known.
Implicit formulation includes some unknown variables at time step $(t + \Delta t)$.

Examples of explicit and implicit solver for linear advection.

Explicit

$$(\Omega \bar{M})_i \frac{\Delta u_i^n}{\Delta t} = -\phi_i^n \qquad (2a)$$

Implicit

$$(\Omega \bar{M})_i \frac{\Delta u_i^n}{\Delta t} = -\theta \phi_i^{n+1} - (1-\theta)\phi_i^n \qquad (2b)$$

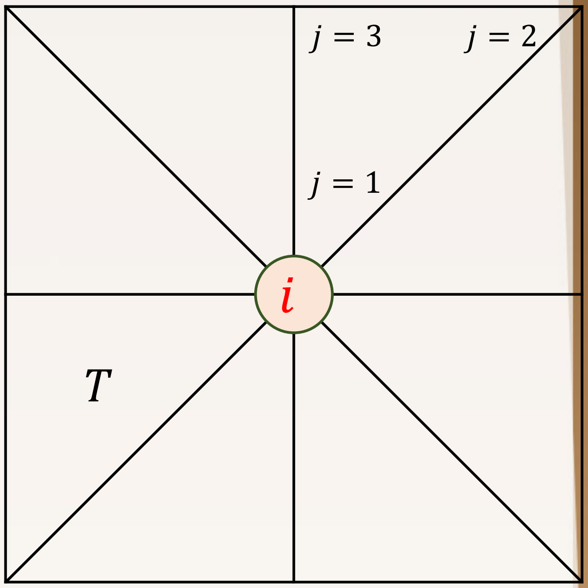
where

$$\Delta u_i^n = u_i^{n+1} - u_i^n \qquad (3)$$

The mass matrix for Implicit Solver

$$\sum_{T \in \cup \Delta_i} \left\{ m_{ij}^T \left(\frac{\partial u}{\partial t} \right)_i + \phi_i^T \right\} = 0 \qquad (4)$$

$$\begin{bmatrix} m_{11}^T & m_{12}^T & m_{13}^T \\ m_{21}^T & m_{22}^T & m_{23}^T \\ m_{31}^T & m_{32}^T & m_{33}^T \end{bmatrix} \quad \begin{bmatrix} \left(\frac{\partial u}{\partial t}\right)_1 \\ \left(\frac{\partial u}{\partial t}\right)_2 \\ \left(\frac{\partial u}{\partial t}\right)_3 \end{bmatrix} \quad \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \phi_3^T \end{bmatrix}$$



$$\frac{\partial u}{\partial t} + \nabla \cdot \vec{F} = 0 \quad \Longrightarrow \quad \frac{\partial u}{\partial t} + \vec{\lambda} \cdot \nabla u = 0$$

The semi-discrete model reads,

Two-time level

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \left(\vec{\lambda} \cdot \nabla u\right)^{n+\frac{1}{2}} = 0$$

Crank-Nicolson (5a)

Three-time level

$$\frac{3u_i^{n+1} - 4u_i^n + u_i^{n-1}}{2\Delta t} + \left(\vec{\lambda} \cdot \nabla u\right)^{n+1} = 0 \qquad (5b)$$

Both the time discretisation schemes are second order accurate

$$\left(\frac{\partial u}{\partial t}\right)_i^{n+\frac{1}{2}} = \frac{u_i^{n+1} - u_i^n}{\Delta t} + \mathcal{O}(\Delta t^2) \qquad (6a)$$

$$\left(\frac{\partial u}{\partial t}\right)_i^{n+1} = \frac{3u_i^{n+1} - 4u_i^n + u_i^{n-1}}{2\Delta t} + \mathcal{O}(\Delta t^2) \qquad (6b)$$

The notation of $\mathcal{O}(\Delta t^2)$ in equations (6a) and (6b) denote that the truncation errors is a series starting from $\Delta t^2,\ \Delta t^3, \cdots, \Delta t^n$.

To obtain equations (6a) and (6b), one should expand the Taylor series for the term $\left(\frac{\partial u}{\partial t}\right)_i$ of equations (5a) and (5b) at

$n + \frac{1}{2}$ for two-level time discretisation, and

$n + 1$ for three-level time discretisation.

The Taylor series of u at t_n reads

$$u_i(t + \Delta t) = \sum_{n=0}^{\infty} \frac{(\Delta t)^n}{n!} \frac{d^n u_i(t_n)}{dx^n} \qquad (7)$$

Two-time level

$$u_i(t_{n+1}) = u_i(t_{n+\frac{1}{2}}) + \left(\frac{\Delta t}{2}\right)\frac{du_i(t_{n+\frac{1}{2}})}{dt} + \frac{(\Delta t/2)^2}{2}\frac{d^2u_i(t_{n+\frac{1}{2}})}{dt^2} + \mathcal{O}(\Delta t^3)$$

$$- \quad u_i(t_n) = u_i(t_{n+\frac{1}{2}}) + \left(-\frac{\Delta t}{2}\right)\frac{du_i(t_{n+\frac{1}{2}})}{dt} + \frac{(-\Delta t/2)^2}{2}\frac{d^2u_i(t_{n+\frac{1}{2}})}{dt^2} + \mathcal{O}(\Delta t^3)$$

$$u_i^{n+1} - u_i^n = \Delta t\left(\frac{\partial u}{\partial t}\right)_i^{n+\frac{1}{2}} + \mathcal{O}(\Delta t^3)$$

Three-time level

$$\times 4 \quad \left\{ u_i(t_n) = u_i(t_{n+1}) + (-\Delta t)\frac{du_i(t_{n+1})}{dt} + \frac{(-\Delta t)^2}{2}\frac{d^2u_i(t_{n+1})}{dt^2} + \mathcal{O}(\Delta t^3) \right.$$

$$- \quad u_i(t_{n-1}) = u_i(t_{n+1}) + (-2\Delta t)\frac{du_i(t_{n+1})}{dt} + \frac{(-2\Delta t)^2}{2}\frac{d^2u_i(t_{n+1})}{dt^2} + \mathcal{O}(\Delta t^3)$$

$$3u_i^{n+1} - 4u_i^n + u_i^{n-1} = 2\Delta t\left(\frac{\partial u}{\partial t}\right)_i^{n+1} + \mathcal{O}(\Delta t^3)$$

$$\left(\phi_i^T\right)^{n+\frac{1}{2}} = \frac{1}{2}\left[\left(\phi_i^T\right)^{n+1} + \left(\phi_i^T\right)^{n}\right] \quad (8c)$$

Two-time level

$$\sum_{T\in\cup\Delta_i}\left\{\sum_{j\in T} m_{ij}^T\left(\frac{u_i^{n+1}-u_i^n}{\Delta t}\right) + \left(\phi_i^T\right)^{n+\frac{1}{2}}\right\} = 0 \quad (8a)$$

Three-time level

$$\sum_{T\in\cup\Delta_i}\left\{\sum_{j\in T} m_{ij}^T\left(\frac{3u_i^{n+1}-4u_i^n+u_i^{n-1}}{2\Delta t}\right) + \left(\phi_i^T\right)^{n+1}\right\} = 0 \quad (8b)$$

where

$$\left(\phi_i^T\right)^{n} = \sum_{j\in T} k_j^T u_j^{T,n} \quad (9a)$$

$$\left(\phi_i^T\right)^{n+1} = \sum_{j\in T} k_j^T u_j^{T,n+1} \quad (9b)$$

Two-time level

$$\sum_{T\in\cup\Delta_i}\left\{\sum_{j\in T} m_{ij}^T\left(\frac{u_i^{n+1}-u_i^n}{\Delta t}\right)+\frac{1}{2}\sum_{j\in T}\left(k_j^T u_j^{T,n}+k_j^T u_j^{T,n+1}\right)\right\}=0 \quad (10a)$$

Three-time level

$$\sum_{T\in\cup\Delta_i}\left\{\sum_{j\in T} m_{ij}^T\left(\frac{3u_i^{n+1}-4u_i^n+u_i^{n-1}}{2\Delta t}\right)+\sum_{j\in T} k_j^T u_j^{T,n+1}\right\}=0 \quad (10b)$$

Two-time level

$$\sum_{T\in\cup\Delta_i}\sum_{j\in T}\tilde{m}_{ij}^{T}u_i^{n+1}=\sum_{T\in\cup\Delta_i}\sum_{j\in T}m_{ij}^{T}u_i^{n}+\frac{\Delta t}{2}\sum_{T\in\cup\Delta_i}\beta_i^{T}(\phi^{T})^{n} \quad (11a)$$

Three-time level

$$\sum_{T\in\cup\Delta_i}\sum_{j\in T}\tilde{m}_{ij}^{T}u_i^{n+1}=\sum_{T\in\cup\Delta_i}\sum_{j\in T}m_{ij}^{T}(4u_i^{n}-u_i^{n-1}) \quad (11b)$$

where

$$m_{ij}^{T}=\frac{S_T}{3}\begin{bmatrix}\beta_1^T(2-\beta_1^T) & \beta_1^T(1-\beta_2^T) & \beta_1^T(1-\beta_3^T)\\ \beta_2^T(1-\beta_1^T) & \beta_2^T(2-\beta_2^T) & \beta_2^T(1-\beta_3^T)\\ \beta_3^T(1-\beta_1^T) & \beta_3^T(1-\beta_2^T) & \beta_3^T(2-\beta_3^T)\end{bmatrix},$$

$$\tilde{m}_{ij}^{T}=m_{ij}^{T}+\sigma\begin{bmatrix}\beta_1^T k_1^T & \beta_1^T k_2^T & \beta_1^T k_3^T\\ \beta_2^T k_1^T & \beta_2^T k_2^T & \beta_2^T k_3^T\\ \beta_3^T k_1^T & \beta_3^T k_2^T & \beta_3^T k_3^T\end{bmatrix},$$

$$\sigma=\begin{cases}\dfrac{\Delta t}{2}, & \text{two level}\\ 2\Delta t, & \text{three level}\end{cases} \quad (11c)$$