SEMINARI DI ANALISI PURA E APPLICATA

Fenomeni di trasporto: Introduzione alla modellazione matematica e simulazione numerica

Numerical methods for partial differential equations

Cristian Marchioli

Udine, April 2019

Content

Part 1

Introduction to PDE

Introduction to PDE

Focus on transport phenomena:

- Mass transport
- Momentum transport
- Heat transport

Transported variable (unknown): $\Gamma(\mathbf{x}(t),t)$

PDEs describe the space/time evolution of the transported variable (via partial derivatives)

Introduction to PDE

PDEs provide a local description of the dependence of Γ on **x**(t) and t.

If Γ depends on a single variable, then the evolution equation is named ODE.

In some cases, one PDE can be de-composed into a system of ODEs: More equations, but easier to solve.

Introduction to PDE

Ingredients:

Physical problem: transport phenomena

Mathematical model: PDE (or ODE)

Numerical method: PDE/ODE solver

Introduction to PDE

General form of a PDE:

$$A\frac{\partial^2\phi}{\partial\xi^2}+B\frac{\partial^2\phi}{\partial\xi\partial\eta}+C\frac{\partial^2\phi}{\partial\eta^2}+D\frac{\partial\phi}{\partial\xi}+E\frac{\partial\phi}{\partial\eta}+F\phi+G=0$$

where:

$\phi=\phi(\xi,\eta)$ dependent variable

ξ,η independent variables

Introduction to PDE

Classification of a PDE:

B^2-4AC [1]

- < 0 → Elliptic PDE (diffusive)
- $= 0$ → Parabolic PDE (damped wave)
- > 0 → Hyperbolic PDE (convective)

Introduction to PDE

This classification works only for $\phi = \phi(\xi, \eta)$!

If $\phi = \phi(\xi, \eta, \gamma)$ then one must write the PDE for either $\phi = \phi(\xi, \eta)$ or $\phi = \phi(\xi, \gamma)$ or $\phi = \phi(\eta, \gamma)$ to classify it according to [1].

Knowledge of the PDE "type" is crucial to select the proper BCs and the most suitable numerical solution method.

Introduction to PDE

Examples:

1. Laplace eq. for steady 2D heat conduction

$$\lambda \frac{\partial^2 T}{\partial x^2} + \lambda \frac{\partial^2 T}{\partial y^2} = 0$$

$A=C=\lambda$, $B=0$ -> $B^2-4AC = -4\lambda < 0$

Elliptic PDE!

Introduction to PDE

Examples:

2. Unsteady 1D heat conduction

$$\rho c_p \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}$$

$A=\lambda$, $B=C=0$ -> $B^2-4AC = 0$

Parabolic PDE!

Introduction to PDE

Examples:

3. Wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

A=1, B=0, C=-c^2 -> B^2-4AC = 4c^2 > 0

Hyperbolic PDE!

Introduction to PDE

Examples:

4. Unsteady 2D heat conduction

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

In space: $A=C=\lambda$, $B=0$ -> $B^2-4AC < 0$ -> Elliptic PDE!

In time: $A=\rho c_p$, $B=C=0$ -> $B^2-4AC = 0$ -> Parab. PDE!

Introduction to PDE

Examples:

5. Unsteady 2D energy equation

$$\rho c_p \left(\underbrace{\frac{\partial T}{\partial t}}_{\text{Parabolic}} + \underbrace{u_x \frac{\partial T}{\partial x} + u_y \frac{\partial T}{\partial y}}_{\text{Hyperbolic}} \right) = \underbrace{\lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)}_{\text{Elliptic}}$$

Introduction to PDE

Examples:

6. Unsteady 2D NS equation

$$\rho\left(\underbrace{\frac{\partial u_i}{\partial t}}_{\text{Parabolic}}+\underbrace{u_j\frac{\partial u_i}{\partial x_j}}_{\text{Hyperbolic}}\right)=-\frac{\partial P}{\partial x_i}+\underbrace{\mu\frac{\partial^2 u_i}{\partial x_j^2}}_{\text{Elliptic}}$$

Introduction to PDE

Boundary conditions: Ref: energy eqn.

1. Dirichlet $\phi_B = f_1(\xi_B)$ → $t_B = t(\xi_B)$

2. Neumann $\left.\frac{\partial \phi}{\partial n}\right|_B = f_2(\xi_B)$ → $-\lambda \left.\frac{\partial t}{\partial n}\right|_B = q_B$

3. Cauchy $a(\xi_B)\cdot\phi + b(\xi_B)\cdot\left.\frac{\partial \phi}{\partial n}\right|_B = f_3(\xi_B)$

→ $-\lambda \left.\frac{\partial t}{\partial n}\right|_B = \alpha\left(t - t_{amb}\right)$

Introduction to PDE

Initial conditions:

1st-order PDE $\phi(\xi,0) = F(\xi)$

2nd-order PDE $\phi(\xi,0) = F(\xi)$ + $\dfrac{\partial \phi}{\partial t}(\xi,0) = G(\xi)$

Introduction to PDE

To specify BCs and ICs one must ensure that the problem is mathematically well-posed:

1. solution exists
2. solution is unique
3. solution is a continuous function of initial and boundary values

Numerically, one must ensure that the scheme is stable

Introduction to PDE

Another possible classification is based on **characteristics**

Characteristic are curves along which the PDE becomes an ODE.

Mathematically, a characteristic curve is called **hypersurface**: manifold or an algebraic variety of dimension $n-1$, embedded in an ambient space (e.g Euclidean space, or affine space or projective space) of dimension n.

–> See notes!

Part 2

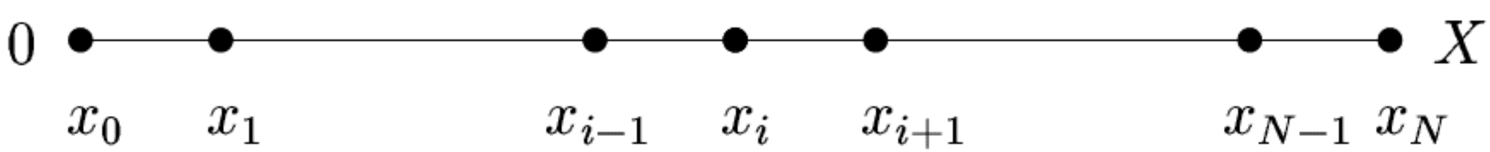
Finite Difference (FD) Discretization

Finite Difference (FD) Discretization

Principle: derivatives in the PDE are approximated by linear combinations of function values at grid points

1D: $\Omega = (0, X)$, $u_i \approx u(x_i)$, $i = 0, 1, \ldots, N$

grid points $x_i = i\Delta x$ mesh size $\Delta x = \frac{X}{N}$



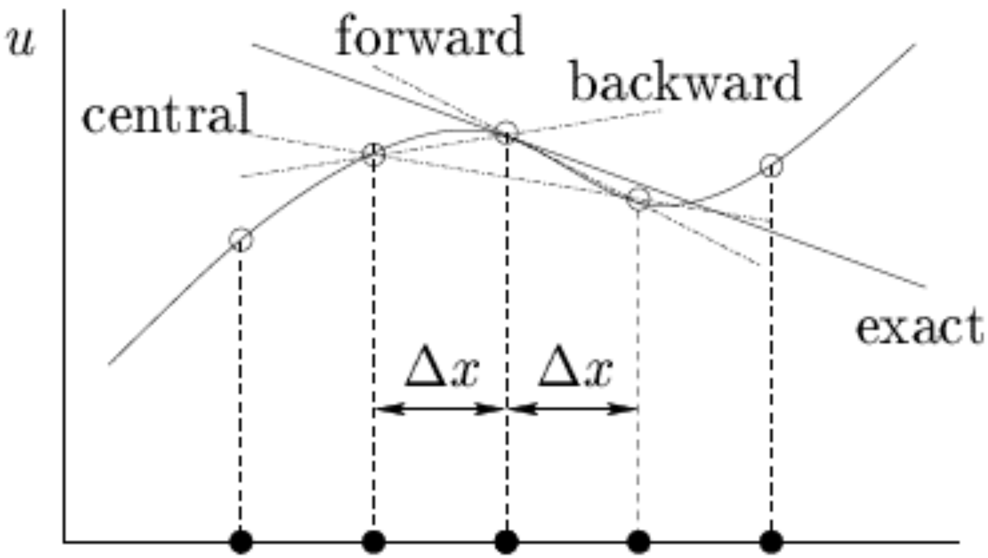
First-order derivatives

$$\frac{\partial u}{\partial x}(\bar{x}) = \lim_{\Delta x \to 0} \frac{u(\bar{x} + \Delta x) - u(\bar{x})}{\Delta x} = \lim_{\Delta x \to 0} \frac{u(\bar{x}) - u(\bar{x} - \Delta x)}{\Delta x}$$

$$= \lim_{\Delta x \to 0} \frac{u(\bar{x} + \Delta x) - u(\bar{x} - \Delta x)}{2\Delta x} \qquad \text{(by definition)}$$

Finite Difference (FD) Discretization

Geometrical interpretation:



Forward:

$$\left[\frac{\partial u}{\partial x}\right]_j^n \cong \frac{u_{j+1}^n - u_j^n}{\Delta x}$$

Backward:

$$\left[\frac{\partial u}{\partial x}\right]_j^n \cong \frac{u_j^n - u_{j-1}^n}{\Delta x}$$

Central:

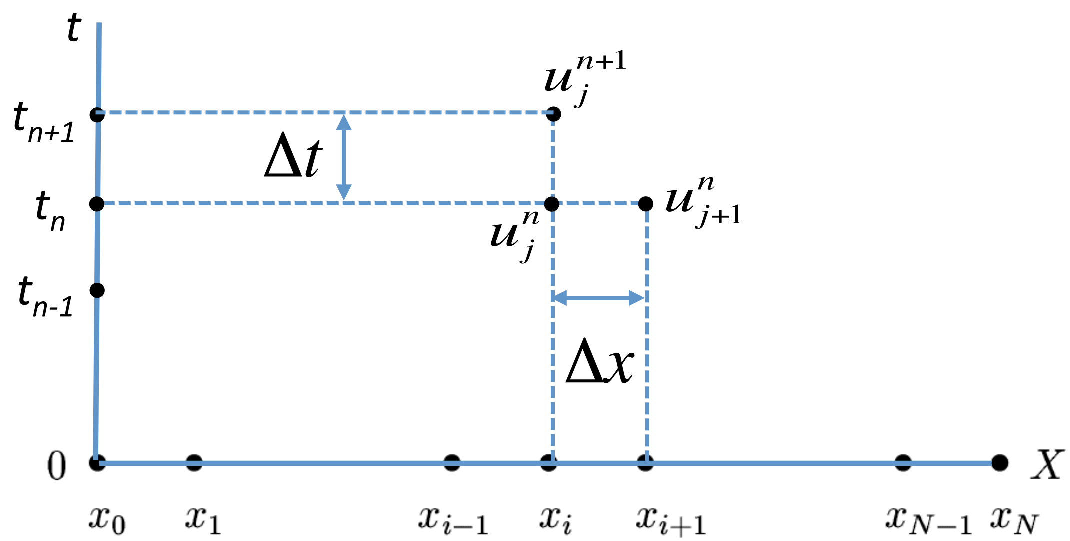
$$\left[\frac{\partial u}{\partial x}\right]_j^n \cong \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}$$

Finite Difference (FD) Discretization

Recall Taylor series expansion:

$$u_{j+1}^{n} = \sum_{m=0}^{\infty} \frac{\Delta x^m}{m!} \left[\frac{\partial^m u}{\partial x^m} \right]_j^n \qquad u_j^{n+1} = \sum_{m=0}^{\infty} \frac{\Delta t^m}{m!} \left[\frac{\partial^m u}{\partial t^m} \right]_j^n$$

in space in time



Finite Difference (FD) Discretization

Apply to unknown:

$$u_{j+1}^{n} = u_{j}^{n} + \Delta x \left[\frac{\partial u}{\partial x} \right]_{j}^{n} + \frac{\Delta x^{2}}{2} \left[\frac{\partial^{2} u}{\partial x^{2}} \right]_{j}^{n} + \frac{\Delta x^{3}}{6} \left[\frac{\partial^{3} u}{\partial x^{3}} \right]_{j}^{n} + \ldots \qquad \text{[T1]}$$

$$u_{j-1}^{n} = u_{j}^{n} - \Delta x \left[\frac{\partial u}{\partial x} \right]_{j}^{n} + \frac{\Delta x^{2}}{2} \left[\frac{\partial^{2} u}{\partial x^{2}} \right]_{j}^{n} - \frac{\Delta x^{3}}{6} \left[\frac{\partial^{3} u}{\partial x^{3}} \right]_{j}^{n} + \ldots \qquad \text{[T2]}$$

with $\text{h.o.t.} \propto O(\Delta x^{3})$

Finite Difference (FD) Discretization

Forward FD (from T1):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} - \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Backward FD (from T2):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} + \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Central FD (from T1-T2):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Finite Difference (FD) Discretization

Forward FD (from T1):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} - \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Backward FD (from T2):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} + \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Central FD (from T1-T2):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Finite Difference (FD) Discretization

Forward FD (from T1):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} - \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Backward FD (from T2):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_j^n}{\Delta x} + \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Central FD (from T1-T2):

$$\left[\frac{\partial u}{\partial x}\right]_j^n = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots$$

Finite Difference (FD) Discretization

Forward FD:

$$TE = \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots \quad \propto O(\Delta x)$$

Backward FD:

$$TE = \frac{\Delta x}{2}\left[\frac{\partial^2 u}{\partial x^2}\right]_j^n - \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots \quad \propto O(\Delta x)$$

Central FD:

$$TE = \frac{(\Delta x)^2}{6}\left[\frac{\partial^3 u}{\partial x^3}\right]_j^n + \ldots \quad \propto O(\Delta x)^2$$

Finite Difference (FD) Discretization

Example: $$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

-> Exact sol. $u(x,t) = \sin(\pi x) \cdot \exp(-\pi^2 t)$

-> Num. sol. $U_j^{n+1} = U_j^n + \Delta t \cdot \frac{\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right)}{\Delta x^2}$

Leading TE is $O(\Delta x)^2$ in space (with central FD) and is $O(\Delta t)$ in time (with forward FD)

Finite Difference (FD) Discretization

Note:

1. If the leading TE is $O(\Delta x)^m$ or $O(\Delta t)^m$ then the discretization method has order *m*
2. All terms in the PDE should be discretized with the same TE (not always possible…)
3. Sometimes TE depends on $\Delta x / \Delta t$ rather than just Δx or Δt

More schemes on the notes…

Part 3

Properties of a numerical method

Properties of a numerical method

1. Consistency
2. Stability
3. Convergence
4. Conservation
5. Boundedness
6. Realizability
7. Accuracy

Properties of a numerical method

1. Consistency:

$$\lim_{\substack{\Delta x \to 0 \\ (\Delta t \to 0)}} (\text{PDE}-\text{DDE}) = \lim_{\substack{\Delta x \to 0 \\ (\Delta t \to 0)}} \text{TE} \to 0$$

where DDE = Discretized Diff. Eq.

TE = Truncation Error

Δx = Grid spacing

Δt = Time step

In other words: Discretization of the PDE should become exact as $\Delta x \to 0$ or $\Delta t \to 0$ (truncation error should vanish)

Properties of a numerical method

2. Stability:

Numerical errors (typically round-off and truncation) generated during the solution of discretized PDE should not be magnified

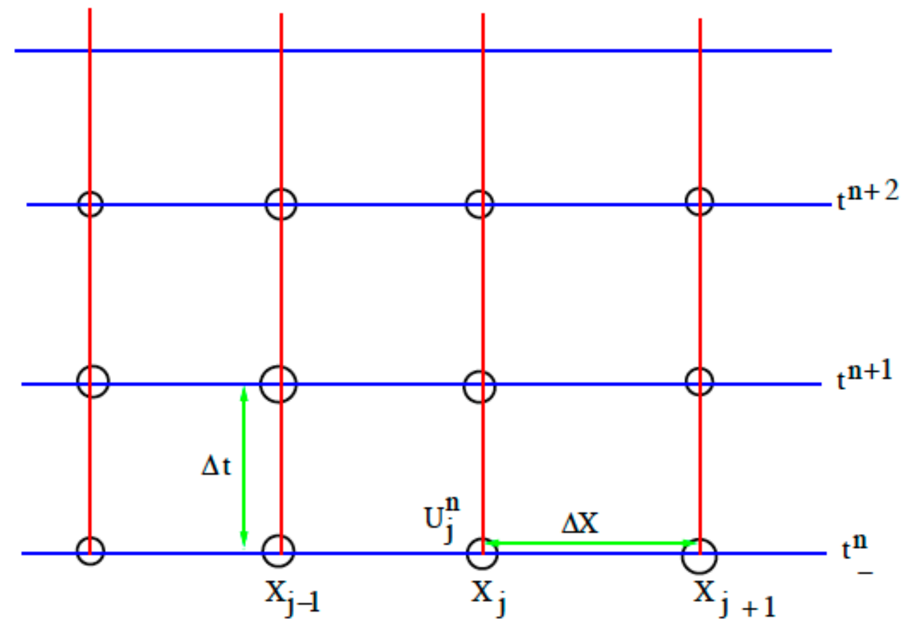
In other words: a method is stable if the total variation of the numerical solution at a fixed time remains bounded as $\Delta t \to 0$.

Stability is sometimes achieved by adding **numerical diffusion** (fictitious damping of the numerical solution as compared to the exact solution)

Properties of a numerical method

2. Stability:

Example of numerical diffusion:



PDE (linear advection eqn)

$$U_t + aU_x = 0$$

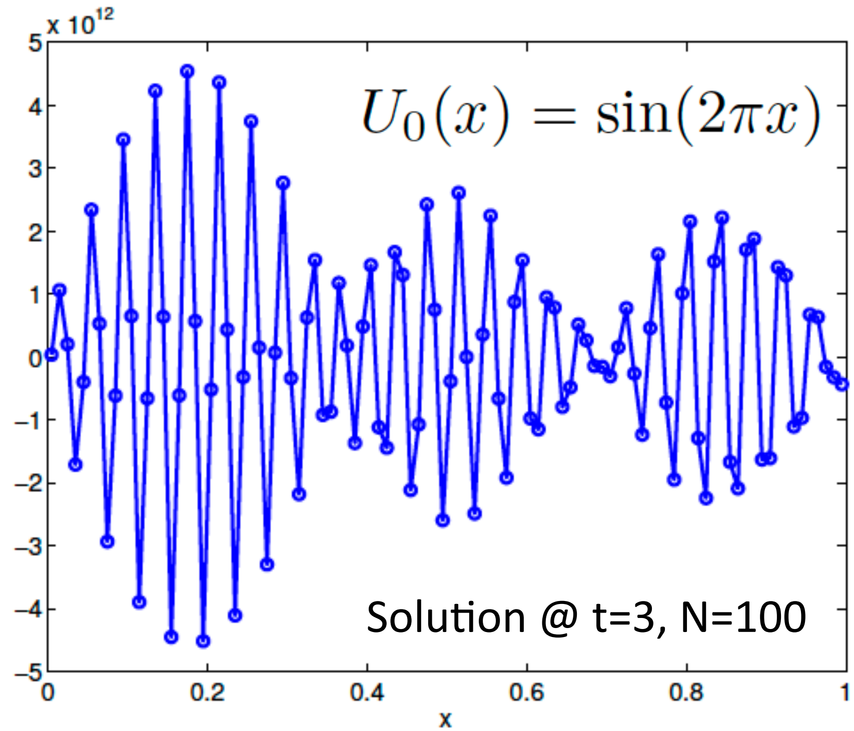
DDE (FTCS)

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_{j+1}^n - U_{j-1}^n)}{2\Delta x} = 0$$

Properties of a numerical method

2. Stability:

Example of numerical diffusion:



PDE (linear advection eqn)

$$U_t + aU_x = 0$$

DDE (FTCS)

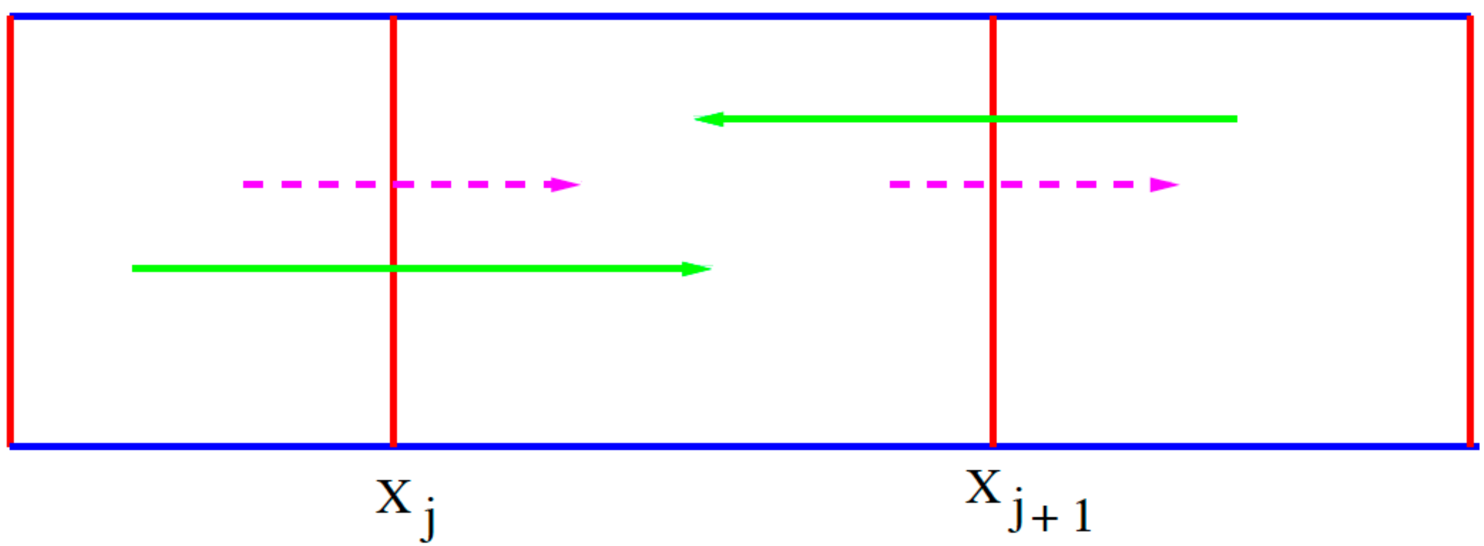
$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_{j+1}^n - U_{j-1}^n)}{2\Delta x} = 0$$

Properties of a numerical method

2. Stability:

Why is the scheme unstable albeit consistent?

Because the exact solution moves to the right for a>0, but the FTCS scheme takes information from both left and right!



Properties of a numerical method

2. Stability:

Example of numerical diffusion:

PDE $$U_t + aU_x = 0$$

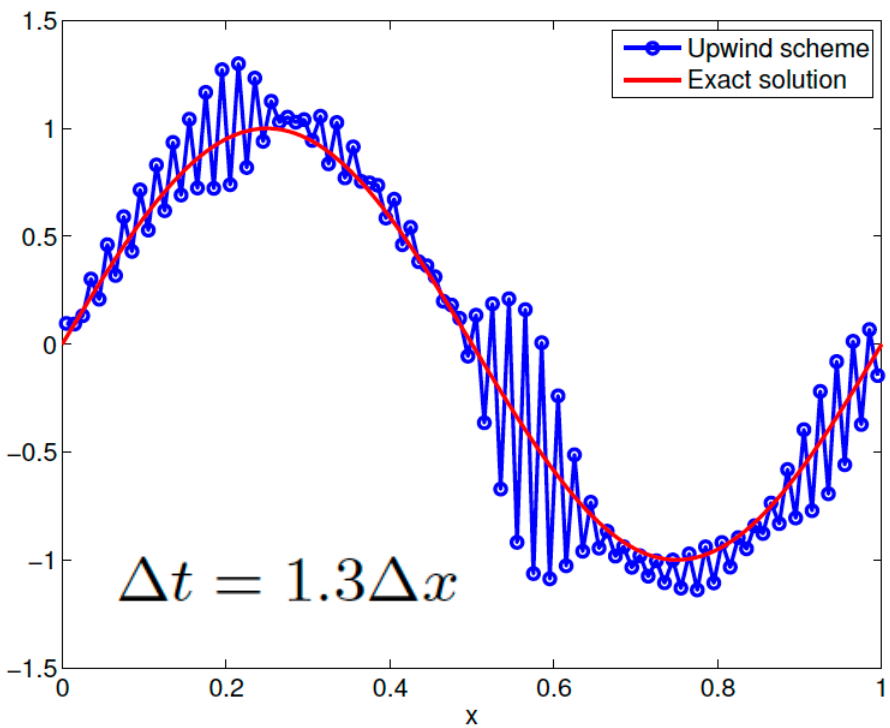
DDE (FTCS) $$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_{j+1}^n - U_{j-1}^n)}{2\Delta x} = 0$$

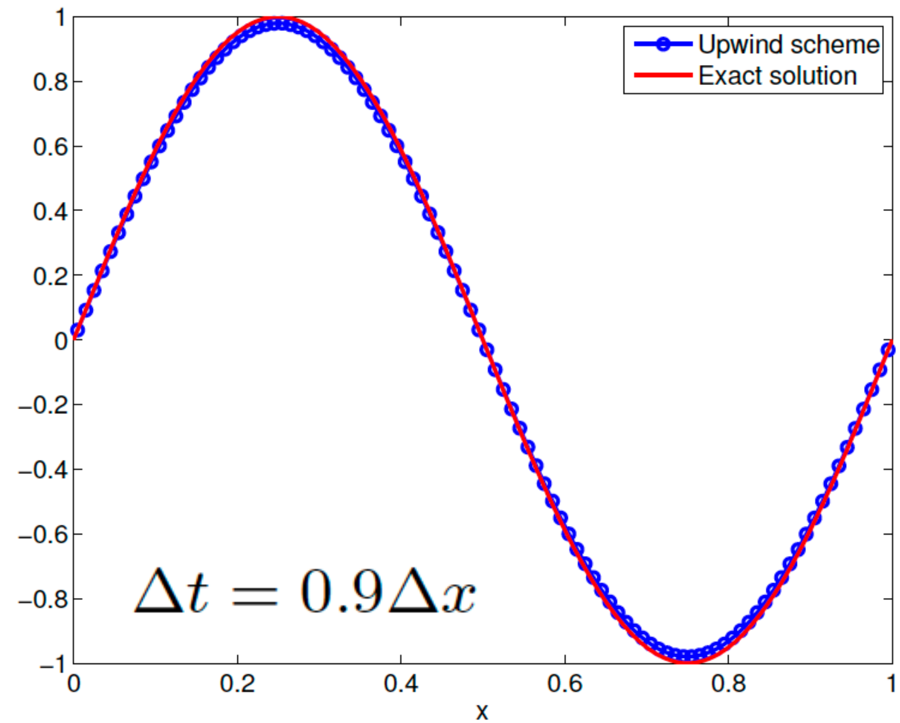
DDE (Upwind scheme)

numerical diffusion (or viscosity)

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

Properties of a numerical method





Note: *Δt/Δx* is an important parameter for stability!

$$\boxed{|a|\frac{\Delta t}{\Delta x} \leqslant 1}$$

CFL condition

Properties of a numerical method

3. Convergence:

The numerical solution should approach the exact PDE solution and converge to it as $\Delta x \to 0$ (and/or $\Delta t \to 0$)

Lax equivalence theorem: **stability + consistency = convergence** (only valid for a consistent FD method and a well-posed linear Initial Value Problem - IVP).

Note: consistency is straightforward to verify, and stability is typically much easier to show than convergence!

What about non-linear IVP? Let's get empirical (find space/time-independent solution through successive refinements…)

Properties of a numerical method

4. Conservation:

Underlying conservation laws should be respected at the discrete level (avoid artificial sources/sinks)

5. Realizability:

The method must ensure physically realizable solution

6. Boundedness:

Quantities like density, viscosity, temperature, concentration, & TKE should remain non-negative and free of spurious wiggles

7. Accuracy:

Affected by modelling + discretization + convergence errors

Part 4

Finite Volumes (FV)

FV method

This method solves for conservation eqns in integral form

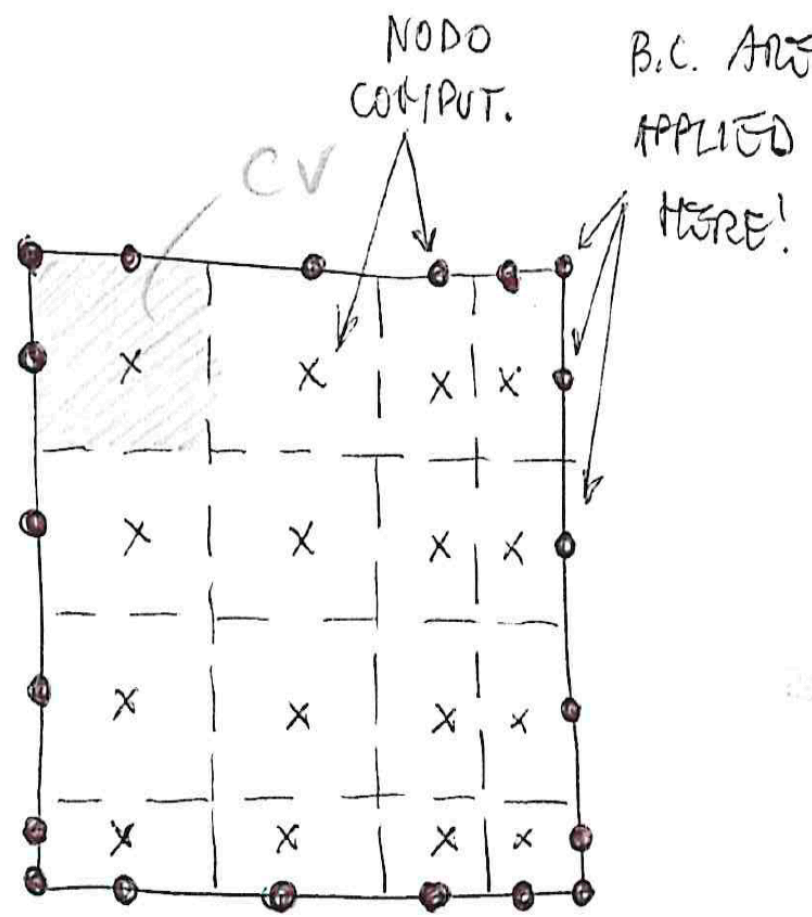
$$\frac{d}{dt}\int_{CV} \phi \, dV + \int_{CS} \vec{m}\cdot\vec{F}_{\phi} \, dS = \int_{CV} P_{\phi} \, dV$$

Rate of change	Flux through CV face	Production within CV

Equations are discretized through Control Volumes

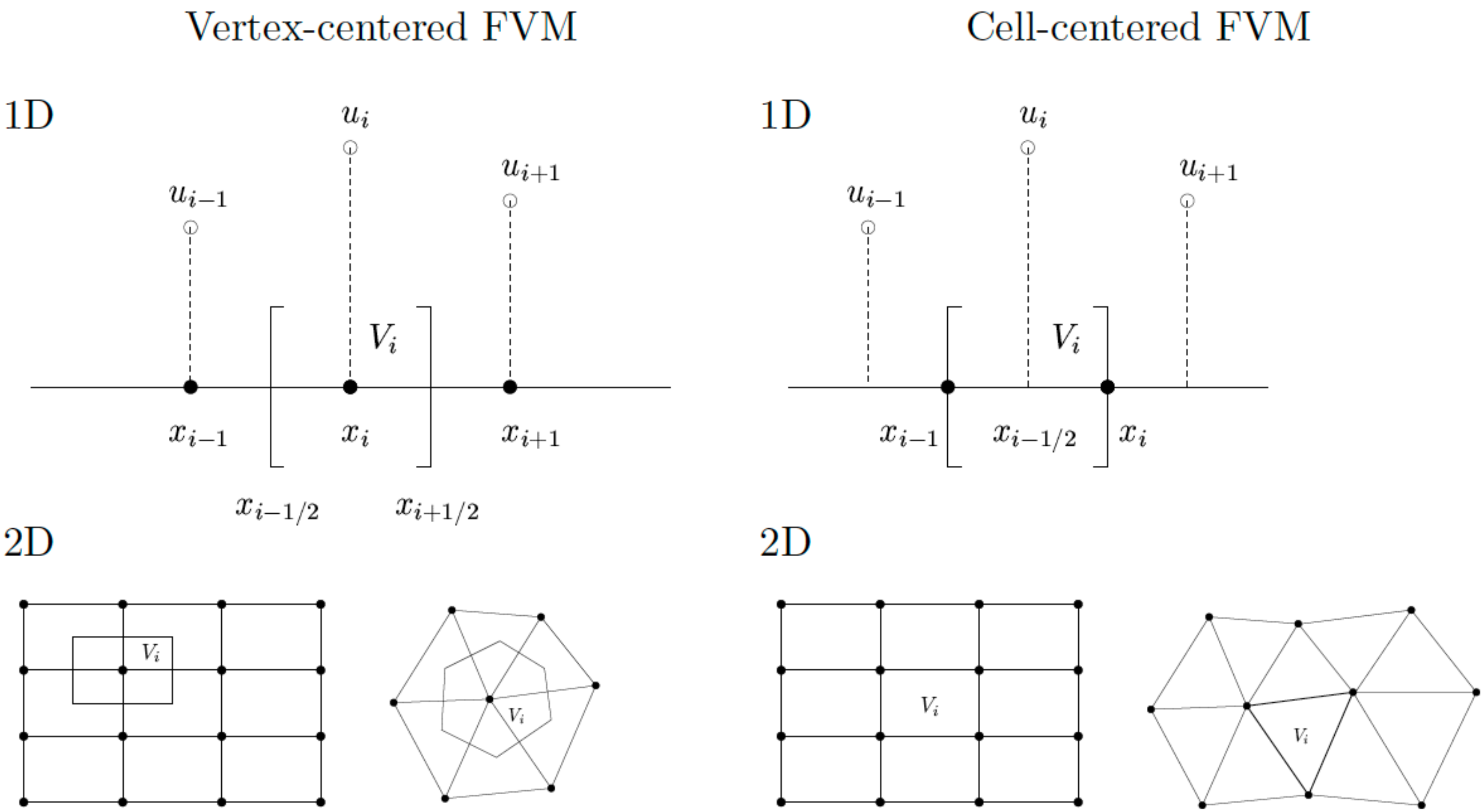
Each CV (and the entire domain) must satisfy conservation properties

To obtain a linear system, integrals must be expressed in terms of mean values



FV method

Definition of CVs:



Different grids / control volumes can be used for different variables $(\mathbf{v}, p, \ldots)$

FV method

$$\int_{CS} \vec{F} \cdot \vec{dS} = \sum_{k=1}^{N} \left(\int_{S_k} \vec{F} \cdot \vec{dS} \right)$$

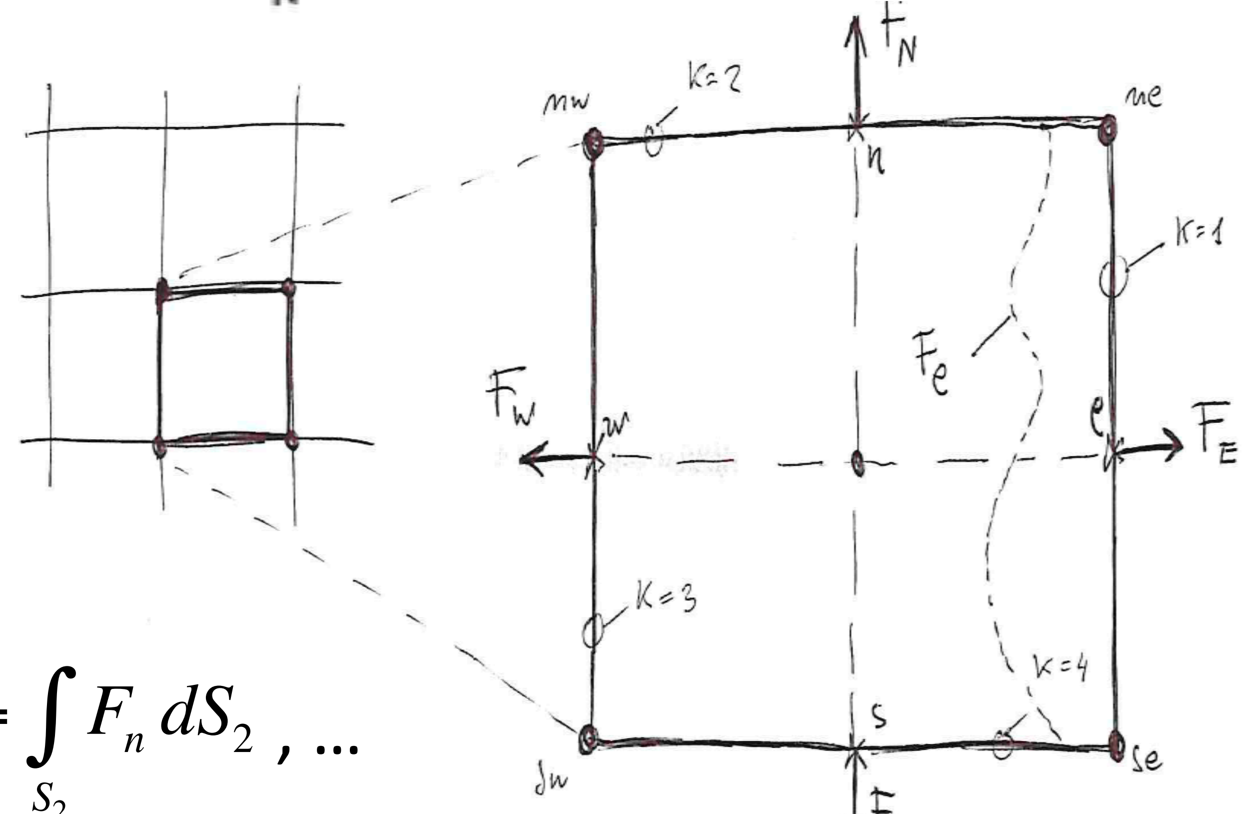
Approximation of surface integral

S_k=k-th CV face

N=number of CV faces

In 2D, N=4

In 3D, N=6



$$F_E = \int_{S_1} F_e \, dS_1 \ , \ F_N = \int_{S_2} F_n \, dS_2 \ , \ldots$$

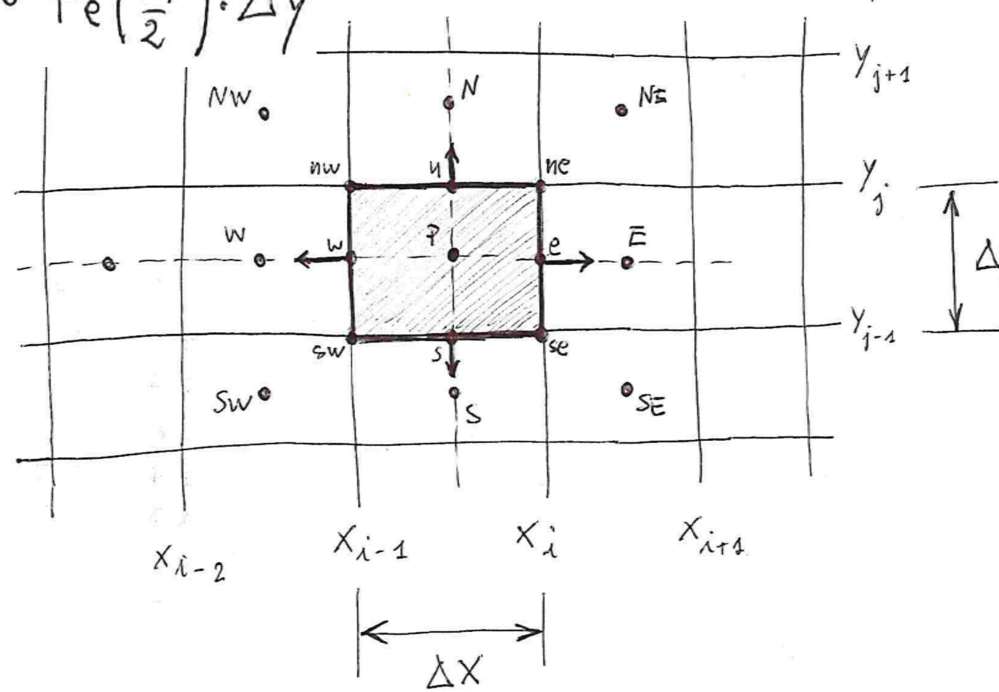
FV method

Quadrature rules provide possible surface/volume integral approximations (see notes for details):

- Midpoint rule (in 2D: 2nd-order accurate in space)

$$\int_{S_e} F_e \, dS_e \equiv \int_0^{\Delta y} F_e \, dy \overset{(*)}{\approx} F_e\left(\frac{\Delta y}{2}\right)\cdot \Delta y$$

$$(*)\begin{cases} \int_a^b f(x)\,dx \approx \underbrace{(b-a)\cdot f(c)}_{M} \\ c = \frac{a+b}{2} \end{cases}$$



In 2D: $S_e = \Delta y$

In 3D: $S_e = \Delta y \Delta z$

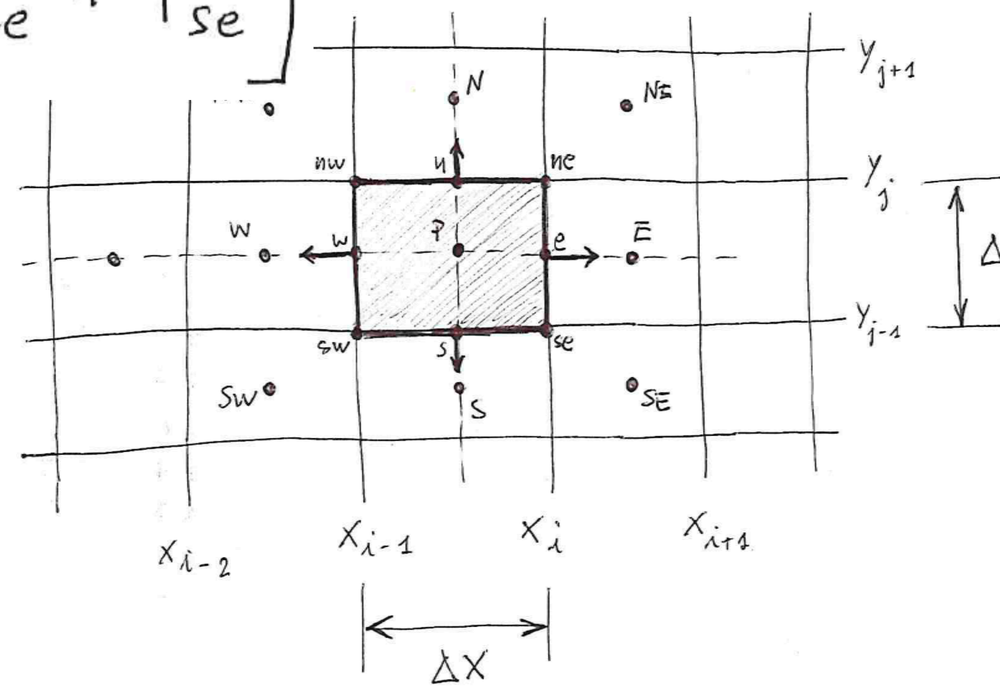
FV method

Quadrature rules provide possible surface/volume integral approximations (see notes for details):

- Trapezoid rule (in 2D: 2nd-order accurate in space)

$$\int_{S_e} F_e \, dS_e \overset{(*)}{\cong} \frac{S_e}{2} \cdot \left[F_{ne} + F_{se} \right]$$

$$(*) \quad \underbrace{\int_a^b f(x)\,dx = \left(\frac{b-a}{2}\right) \cdot \left(f(a) + f(b)\right)}_{T}$$



In 2D: $S_e = \Delta y$

In 3D: $S_e = \Delta y \Delta z$

FV method

Quadrature rules provide possible surface/volume integral approximations (see notes for details):

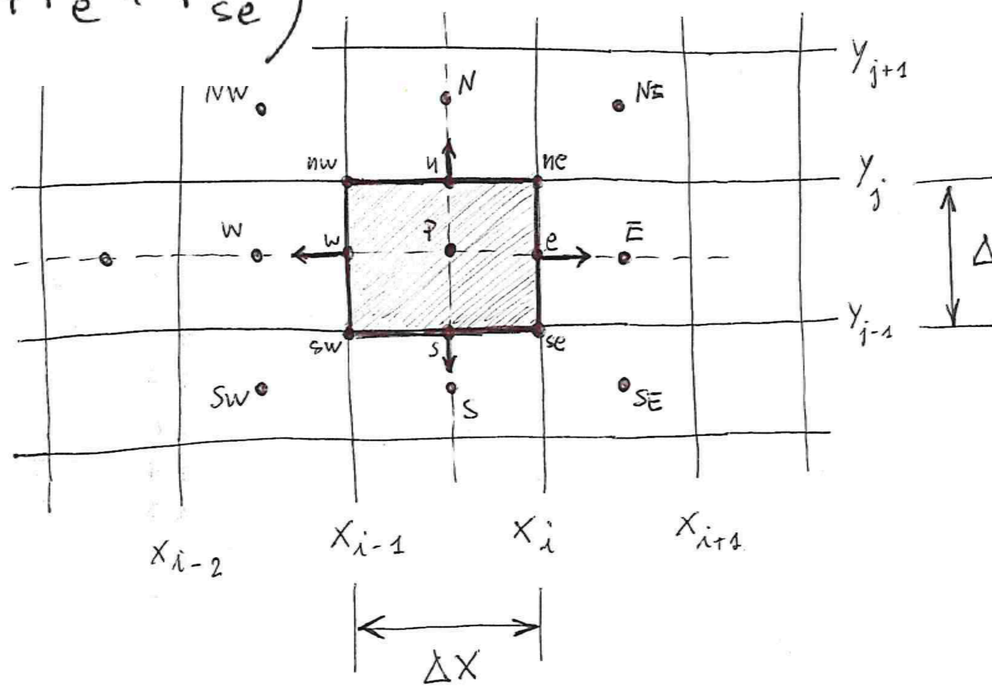
- Simpson's rule (in 2D: 4th-order accurate in space)

$$\int_{S_e} F_e \, dS_e \cong \frac{S_e}{6}\left(F_{ne} + 4F_e + F_{se}\right)$$

$$(*) \quad \int_a^b f(x)\,dx = \frac{2}{3}M + \frac{1}{3}T$$

In 2D: $S_e = \Delta y$

In 3D: $S_e = \Delta y \Delta z$



FV method

Problem: solution only available at CV centers, BUT
function values are needed at quadrature points!

Disadvantages:

- Requires integration but also interpolation
- Difficult to implement high-order (3^{rd} or higher) FV schemes for evaluation of fluxes/integrals in 3D problems

Advantages:

- Conservative by construction
- Physically grounded
- Can be used with complex grids
- Easy to implement

Part 5

Finite Elements (FE)

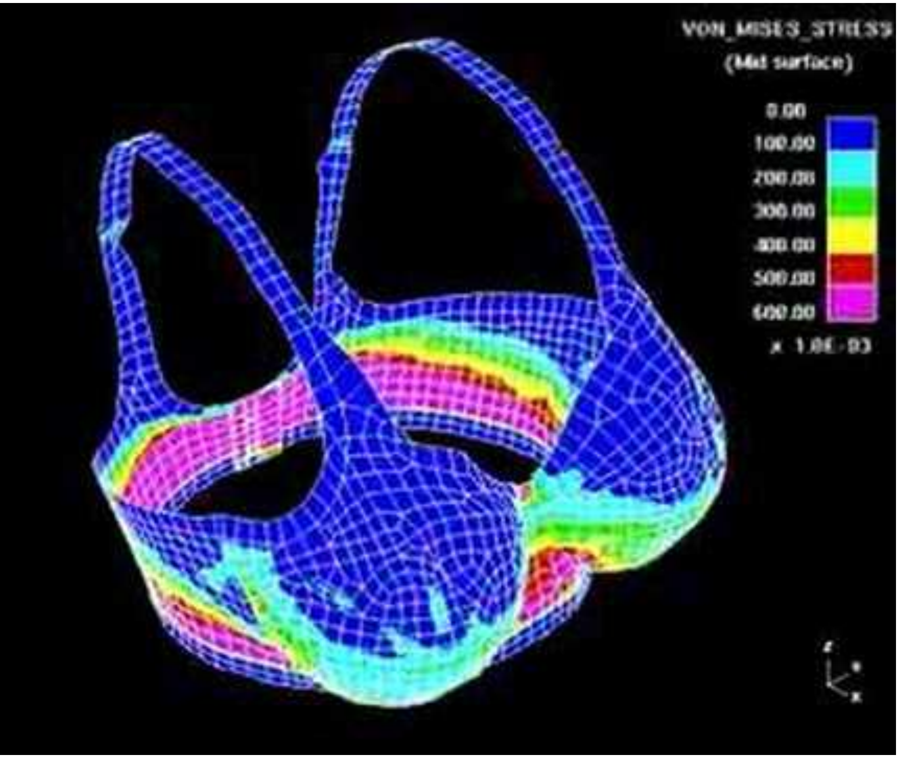
FE method

This method is very popular in structural analysis but is also used in heat transfer, fluid flow, mass transport, and electromagnetic potential BVPs for PDEs.

The FEM formulation:

- subdivides the targeted problem into smaller, simpler parts called finite elements
- for each FE, typically results in a system of ODEs (for unsteady problems) or of algebraic equations (for stationary problems) that yield approximate values of the unknowns at discrete points over the domain

FE method



The FEM shares similarities with the FVM (CVs replaced by FEs)

Main difference: conservation eqns solved for each FE are multiplied by **weight function** before integration

Pros: applicable to complex geometries (unstruct. grids)

Cons: typically requires solution of large sparse systems of linear algebraic eqns (computationally expensive!)

FE method

Galerkin formulation (aka Weighted-Residual Method)

Consider a PDE: $$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$

(*) $\phi_j(x) = x^{j-1}$

$\phi_j(x) = \sin(j\pi x)$

Assume the PDE admits an approximate solution:

$$T(x,y,z,t) = T_0(x,y,z,t) + \sum_{1}^{N} a_j(t)\phi_j(x,y,z)$$

where:

- T_0 = part of the solution that satisfies BCs/Ics
- $a_j(t)$ = unknown coefficients (to be determined!)
- $\phi_j(x,y,z,t)$ = trial functions (known analytical functions (*) e.g. polynomials or trigonometric functions)

FE method

Galerkin formulation (cntd)

Define the eqns' residual, *R*:

$$R(\bar{T}) = \frac{\partial \bar{T}}{\partial t} - \alpha \frac{\partial^2 \bar{T}}{\partial x^2} = 0$$

if $\bar{T}$ = exact PDE solution

$$R(T) = \frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} \neq 0$$

if T= exact PDE solution

Coeffs. $a_j(t)$ can be determined imposing:

$$\iiint W_m(x,y,z)\, R \, dx\, dy\, dz = 0$$

$m = 1, \ldots, M$

FE method

Galerkin formulation (cntd)

Define the eqns' residual, *R*:

$$R(\bar{T}) = \frac{\partial \bar{T}}{\partial t} - \alpha \frac{\partial^2 \bar{T}}{\partial x^2} = 0$$

if $\bar{T}$ = exact PDE solution

$$R(T) = \frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} \neq 0$$

if T= exact PDE solution

Coeffs. $a_j(t)$ can be determined imposing:

$$\iiint W_m(x, y, z)\, R\, dx\, dy\, dz = 0$$

weight

FE method

Galerkin formulation (cntd)

Selection of weights:

$$W_m(x, y, z) = \phi_m(x, y, z)$$

Other formulations use different expressions for W_m…

See notes for an application of the Galerkin-based FE

Part 6

Time-marching methods for ODEs

Time-marching methods for ODEs

Consider the following system:

$$\begin{cases} \dfrac{d\phi(t)}{dt} = f(t, \phi(t)) & \text{ODE} \\ \phi(t=0) = \phi^0 & \text{IC} \end{cases}$$

We want to compute:

$$\phi^{n+1} = \phi^n + \int_{t_n}^{t_{n+1}} f(t, \phi(t))\, dt$$

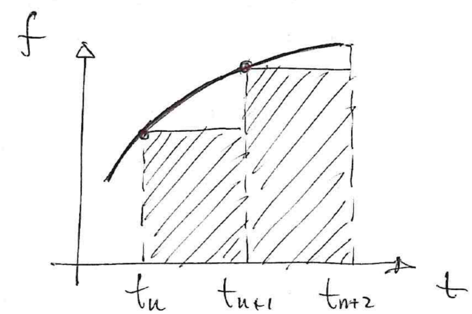
A wealth of methods is available!

Time-marching methods for ODEs

1. Euler methods

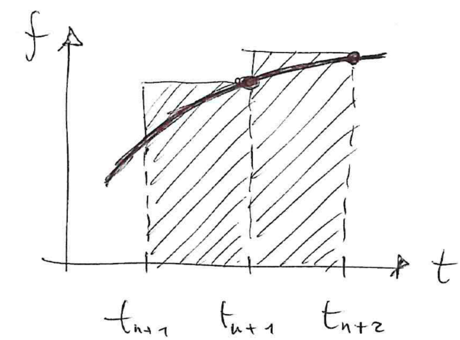
1.1 Forward ≈ O(Δt)

$$\phi^{n+1} = \phi^{n} + f(t_n, \phi^{n})\Delta t$$



1.2 Backward ≈ O(Δt)

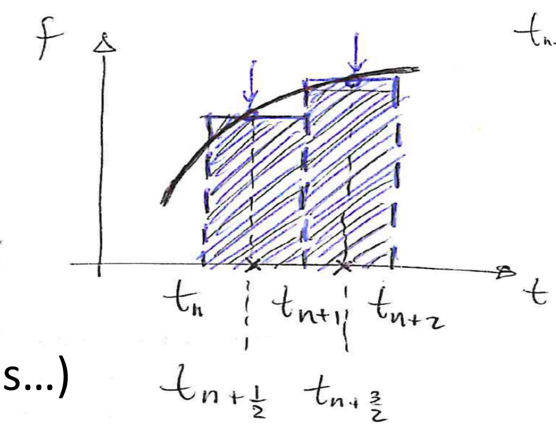
$$\phi^{n+1} = \phi^{n} + f(t_{n+1}, \phi^{n+1})\Delta t$$



2. **Midpoint rule** ≈ O(Δt)²

$$\phi^{n+1} = \phi^{n} + f(t_{n+\frac{1}{2}}, \phi^{n+\frac{1}{2}})\Delta t$$

-> base for Leapfrog method (see last slides…)

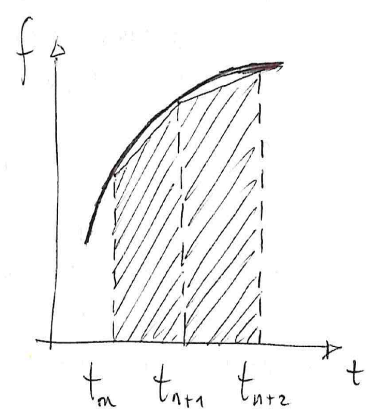


Time-marching methods for ODEs

3. **Trapezoidal rule** ≈ O(Δt)[2]

$$\phi^{n+1} = \phi^{n} + \frac{1}{2}\left[f(t_n, \phi^{n}) + f(t_{n+1}, \phi^{n+1}) \right] \Delta t$$

-> base for Crank-Nicholson and Adams-Moulton



These are examples of two-level (n & n+1 or n-1), one-step methods, which typically work well for small *Δt*.

Note: Small *Δt* -> Stiffness -> Stability -> Boundedness…

Stiff problems are characterized by a range of length and time scales (typical examples: turbulence, atmospheric chemistry)

Time-marching methods for ODEs

If τ_L and τ_S are the longest and the shortest time scale over which the solution of the PDE varies, then:

Stiffness $S = \tau_L / \tau_S$

DDE solvers generally require $\Delta t \cong \tau_S$ but solution must be integrated over time periods $\Delta T \cong \tau_{L.}$

Hence: $N_{t.s.} \cong S$

Ex.: atmospheric chemistry problem

solution for long-lived species over $\Delta T \cong$ 1 year $\cong$ 3E+7 sec

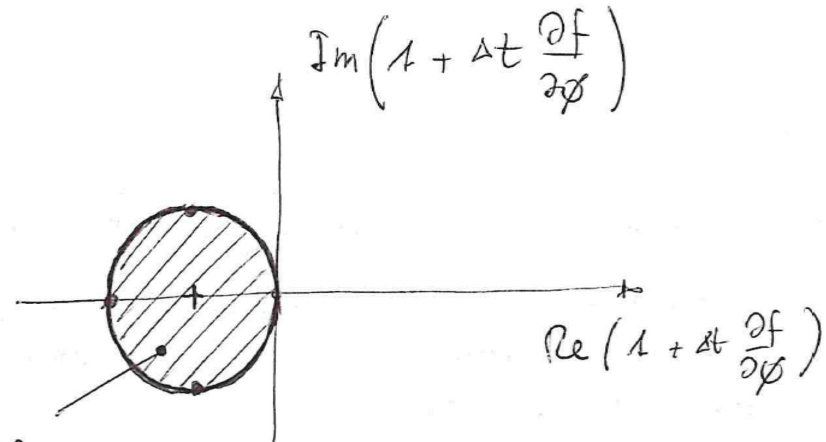
species lifetime $\Delta t \cong$ 1 sec

Time-marching methods for ODEs

Stability of Euler methods:

1. Explicit

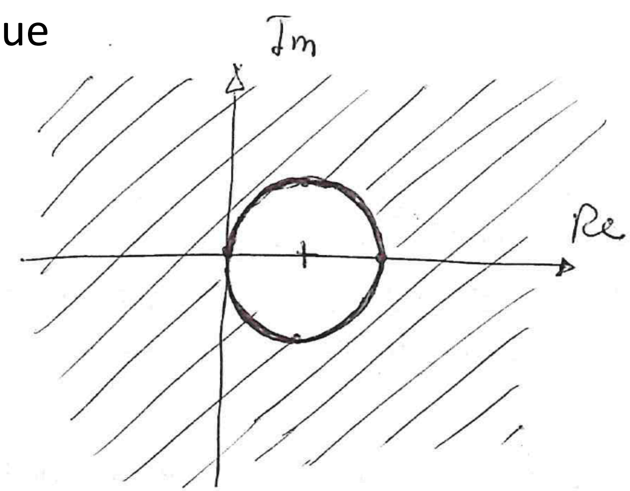
$$\left| 1 + \Delta t \frac{\partial f(t,\phi)}{\partial \phi} \right| < 1$$



$$\left| 1 + \Delta t \underbrace{\frac{\partial f}{\partial \phi}}_{\text{eigenvalue}} \right| \leq 1$$

2. Implicit

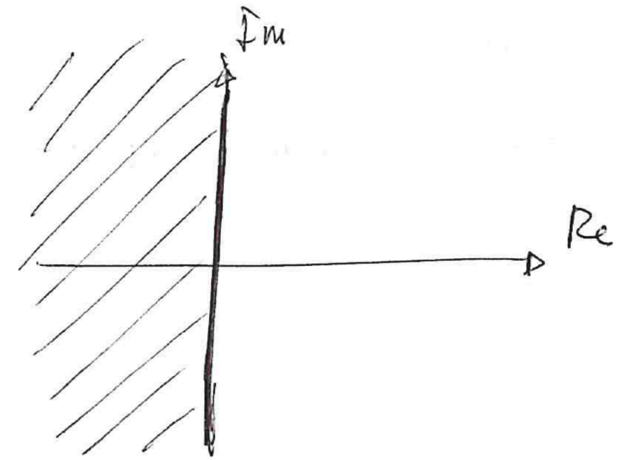
$$\left| \frac{1}{1 - \Delta t \frac{\partial f}{\partial \phi}} \right| \leq 1$$



Time-marching methods for ODEs

Stability of Trapezoidal rule:

$$\left| \frac{1 + \frac{1}{2}\Delta t \frac{\partial f}{\partial \phi}}{1 - \frac{1}{2}\Delta t \frac{\partial f}{\partial \phi}} \right| \leq 1$$



Note:

1. Trapezoidal rule produces bounded solution regardless of Δt if $\frac{\partial f}{\partial \phi} < 0$ -> UNCONDITIONALLY STABLE!
2. Euler methods are always CONDITIONALLY STABLE because boundeness of solution depends on Δt and $\frac{\partial f}{\partial \phi}$

Time-marching methods for ODEs

Explicit methods are:

- easy to code
- computationally cheap (in terms of required memory and computational time)
- unstable for large Δt

Implicit methods are:

- less easy to code
- computationally expensive (wrt explicit methods)
- require iterations to compute the solution
- much more stable

Time-marching methods for ODEs

Predictor-corrector methods $\approx O(\Delta t)^2$

$$\begin{cases} \phi^*_{n+1} = \phi^n + f(t_n, \phi^n)\cdot \Delta t & \text{Explicit Euler} \\ \phi^{n+1} = \phi^n + \frac{1}{2}\left[f(t_n, \phi^n) + f(t_{n+1}, \phi^*_{n+1}) \right] & \text{Trapezoid rule} \end{cases}$$

Multi-point methods (aka Adams methods):

- At least, 2-nd order accuracy
- At least, 2 time instants
- based on polynomial fitting (e.g. Lagrange polynomials)

Time-marching methods for ODEs

Adams-Bashforth methods:

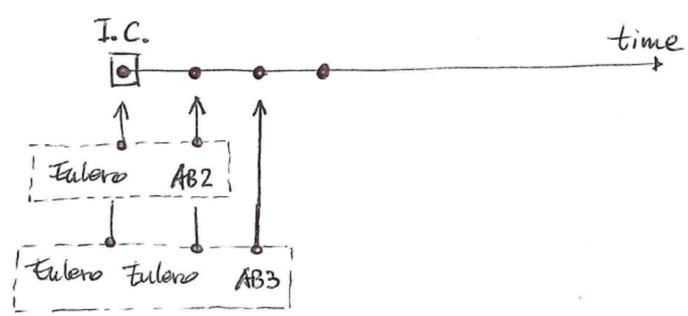
AB1 $\phi^{n+1} = \phi^{n} + \Delta t \cdot f(t_n, \phi^{n})$ (same as explicit Euler)

AB2 $\phi^{n+1} = \phi^{n} + \frac{\Delta t}{2}\left[3 f(t_n, \phi^{n}) - f(t_{n-1}, \phi^{n-1})\right]$

AB3 $\phi^{n+1} = \phi^{n} + \frac{\Delta t}{12}\left[23 f(t_n, \phi^{n}) - 16 f(t_{n-1}, \phi^{n-1}) + 5 f(t_{n-2}, \phi^{n-2})\right]$

$\text{T.E} \propto O(\Delta t^{4})$

AB-R -> $\text{T.E} \propto O(\Delta t^{R+1})$



Time-marching methods for ODEs

Adams-Moulton methods:

AM1 $\phi^{n+1} = \phi^{n} + f(t_{n+1}, \phi^{n+1}) \cdot \Delta t$ (same as implicit Euler)

AM2 $\phi^{n+1} = \phi^{n} + \frac{1}{2}\Delta t \left[f(t_n, \phi^n) + f(t_{n+1}, \phi^{n+1}) \right]$ (same as trapezoid)

AM3 $\phi^{n+1} = \phi^{n} + \frac{\Delta t}{12} \left[5 f(t_{n+1}, \phi^{n+1}) + 8 f(t_n, \phi^n) - f(t_{n-1}, \phi^{n-1}) \right]$

↓

$\mathrm{T.E} \propto O(\Delta t^5)$

AB-*R* -> $\mathrm{T.E} \propto O(\Delta t^{R+2})$

Time-marching methods for ODEs

Runge-Kutta methods:

RK2 (expl.)

$$\phi^{*}_{n+\frac{1}{2}} = \phi^{n} + \frac{\Delta t}{2} \cdot f(t_n, \phi^{n})$$

half-step predictor (explicit Euler)

$$\phi^{n+1} = \phi^{n} + \Delta t \cdot f(t_{n+\frac{1}{2}}, \phi^{*}_{n+\frac{1}{2}})$$

half-step corrector (midpoint rule)

Time-marching methods for ODEs

Runge-Kutta methods:

RK3 (expl.)

$$\phi^{*}_{n+1/3} = \phi^{n} + \frac{\Delta t}{3} \cdot f(t_n, \phi^{n})$$

predictor (explicit Euler)

$$\phi^{**}_{n+2/3} = \phi^{n} + \frac{2\Delta t}{3} \cdot f(t_{n+1/3}, \phi^{*}_{n+1/3})$$

predictor

$$\phi^{n+1} = \phi^{n} + \frac{3\Delta t}{4} \cdot f(t_{n+2/3}, \phi^{**}_{n+2/3}) + \frac{\Delta t}{4} \cdot f(t_{n+1/2}, \phi^{*}_{n+1/3})$$

corrector

Time-marching methods for ODEs

Runge-Kutta methods:

RK4 (expl.)

$$\phi^{*}_{n+1/2} = \phi^{n} + \frac{\Delta t}{2} \cdot \overbrace{f(t_n, \phi^{n})}^{f^{n}}$$ predictor

$$\phi^{**}_{n+1/2} = \phi^{n} + \frac{\Delta t}{2} \cdot \overbrace{f(t_{n+1/2}, \phi^{*}_{n+1/2})}^{f^{*}}$$ predictor

$$\phi^{***}_{n+1} = \phi^{n} + \Delta t \cdot \overbrace{f(t_{n+1/2}, \phi^{**}_{n+1/2})}^{f^{**}}$$ predictor

$$\phi^{n+1} = \phi^{n} + \frac{\Delta t}{6} \cdot \left[f^{n} + 2 f^{*} + 2 f^{**} + f(t_{n+1}, \phi^{***}_{n+1}) \right]$$ corrector

RK5 ...

Time-marching methods for ODEs

Other methods:

1. **Leapfrog** $\phi^{n+1} = \phi^{n-1} + f(t_n, \phi^n) \cdot 2\Delta t$

- Midpoint rule applied to $2\Delta t$ instead of Δt
- Good for hyperbolic PDEs, unstable for parabolic PDEs
- Used in meteorology/oceanography
- Unconditionally unstable for unsteady problems, yet instability is weak for small Δt and can be controlled by imposing:

$$\phi^{n} \simeq \frac{1}{2}\left(\phi^{n-1} + \phi^{n+1}\right)$$

Time-marching methods for ODEs

Other methods:

2. **Lax-Wendroff**: 2^{nd}-order accurate (in space and time) two-step method based on FD and developed for hyperbolic PDEs -> see notes for details

3. **Du Fort – Frankel**: modification of the unstable Leap-frog scheme. This method is explicit and unconditionally stable for the wave equation -> see notes for details

Suggested Readings

Schiesser W.E. & Silebi C. A., Computational Transport Phenomena: Numerical Methods for the Solution of Transport Problems, Cambridge University Press

Mazumder S., Numerical Methods for Partial Differential Equations: Finite Difference and Finite Volume Methods, Elsevier

Farmer R.C., Pike R.W., Cheng G.C., Chen Y.-S., Computational Transport Phenomena for Engineering Analyses, CRC Press

Ferziger J.H. & Peric M., Computational Methods for Fluid Dynamics, Springer-Verlag

Fletcher C. A. J., Computational Techniques for Fluid Dynamics. Vol. I: Fundamental and General Techniques, Springer

Zinkiewicz O. C., Taylor R. L., Nithiarasu P., The finite element method for fluid dynamics, Elsevier