Time stepping methods

ATHENS course: Introduction into Finite Elements

Delft Institute of Applied Mathematics, TU Delft

Matthias Möller (m.moller@tudelft.nl)

18 March 2015

TUDelft

Objectives of this lecture

In this lecture you will learn about

- single- and multi-step time integration methods
- general concepts for time-stepping
  - explicit and implicit schemes
  - predictor-corrector approach
  - higher-order methods
- the modified equation approach for stability analysis
- advanced time-stepping methods with stabilizing properties
- adaptive control of the time-step size

Slides and handouts are available at my website:
`http://ta.twi.tudelft.nl/nw/users/matthias/teaching.html`

Part I

Space-Time Discretization

Space-Time discretization

Initial-boundary value problem (IBVP)

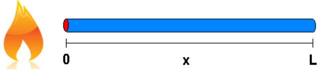
$$\begin{cases} \frac{\partial u(\mathbf{x},t)}{\partial t} + \mathcal{L}u(\mathbf{x},t) = f & \text{in } \Omega \times (0,T) & \text{time-dependent PDE} \\ u(\mathbf{x},t) = g_0 & \text{on } \Gamma_0 \times (0,T) & \\ \mathbf{n}\cdot\nabla u(\mathbf{x},t) = g_1 & \text{on } \Gamma_1 \times (0,T) & \text{boundary conditions} \\ \mathbf{n}\cdot\nabla u(\mathbf{x},t) + \alpha u(\mathbf{x}) = g_2 & \text{on } \Gamma_2 \times (0,T) & \\ u(\mathbf{x},0) = u_0 & \text{in } \Omega \text{ at } t=0 & \text{initial condition} \end{cases}$$

Observation: transient term $\frac{\partial u}{\partial t}$ represents **linear convection** with unit velocity along the t-direction (*"forward in time"*)

- Interpret n-dimensional IBVP as $n+1$-dimensional space-time BVP
- Use FEM-space discretization technique to approximate the $n+1$-dimensional BVP by a set of algebraic equations

Example: Transient heat equation

Experiment: rod with initial constant temperature is heated from the left

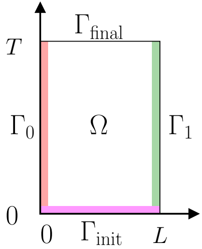


Transient heat equation in 1D

$\frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2} = 0$	in $(0, L) \times (0, T)$	time-dependent PDE
$u(0, t) = 1$	at $x = 0$, in $(0, T)$	Dirichlet boundary conditions
$\frac{\partial u(1,t)}{\partial x} = 0$	at $x = L$, in $(0, T)$	Neumann boundary conditions
$u(x, 0) = u_0$	in $(0, L)$, at $t = 0$	initial condition

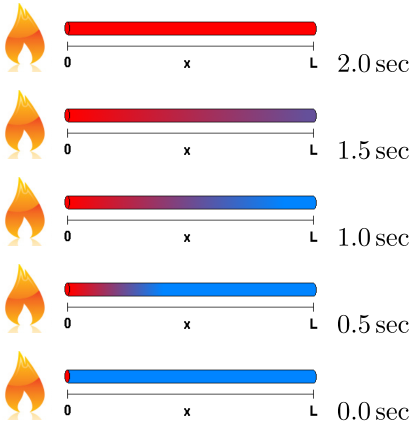
Space-time heat equation $\mathcal{L} := \frac{\partial}{\partial t} - \alpha \frac{\partial^2}{\partial x^2}$

$\mathcal{L}u = 0$	in Ω	stationary PDE
$u = 1$	on Γ_0	Dirichlet boundary conditions
$\frac{\partial u}{\partial x} = 0$	on Γ_1	Neumann boundary conditions
$u = u_0$	on Γ_{init}	Dirichlet boundary conditions

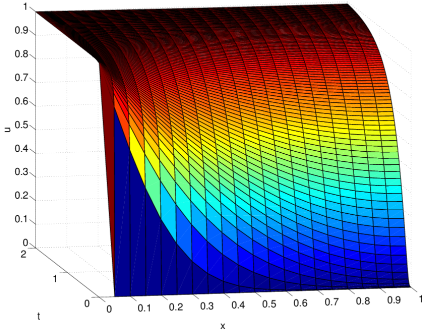


Example: Transient heat equation

Transient heat equation



Space-time heat equation



Grid: $\Delta x = 0.05, \quad \Delta t = 0.01$

Attains steady-state limit $u \equiv 1$, that is, $\frac{\partial u}{\partial t} \to 0$ for time $t \to \infty$

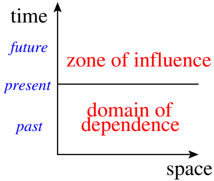
Take home lessons #1

- Time-dependent IBVPs in nD can be interpreted as stationary BVPs in $(n+1)$D subject to additional boundary conditions at Γ_{init}
- Solution values for all time levels can be computed simultaneously
- Space-time discretization can become computationally expensive (large memory requirements and computing times especially in 3D)
- Final time T must be known a priori and cannot be changed
- Mesh generation in 4D is non-trivial (need for special elements)
- Solutions to boundary value problems can be computed by marching the solution of the associated transient problem to the steady state

Part II

Single-Step Methods

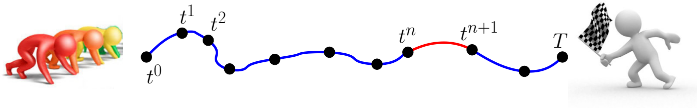
Time-stepping techniques



<u>Idea:</u> Unsteady flows are **parabolic in time**

⇒ use *"time-stepping"* methods to advance transient solutions step-by-step in time

<u>Time discretization:</u> $0 = t^0 < t^1 < t^2 \cdots < t^M = T$ and solve IBVP on short time intervals (t^n, t^{n+1}), where $t^{n+1} = t^n + \Delta t$



- Initialize solution values by the initial condition $u^0 \approx u_0$ in Ω
- Given $u^n \approx u(t^n)$ use it as initial condition to compute $u^{n+1} \approx u(t^{n+1})$

TU Delft

Time-stepping techniques

Time-dependent PDE $$\frac{\partial u(\mathbf{x},t)}{\partial t} + \mathcal{L}u(\mathbf{x},t) = f \quad \text{in } \Omega \times (0,T)$$

<u>Observation:</u> The space and time variables $\mathbf{x}$ and t are essentially decoupled and can be discretized independently

Space discretization by the FEM yields time-dependent unknowns $u_h(t)$

Method of lines (MOL): $\mathcal{L} \to \mathcal{L}_h$ yields ODE system for $u_i(t)$

$$\frac{\mathrm{d}u_h(t)}{\mathrm{d}t} + \mathcal{L}_h u_h(t) = f_h \quad \text{on } (t^n, t^{n+1}) \quad \text{semi-discretized equations}$$

<u>Remark:</u> It is also possible to perform the time discretization *before* the discretization in space (later in this lecture)

Method of lines (MOL)

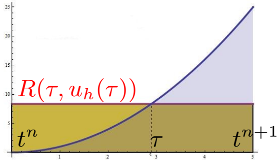
MOL $\rightarrow$ $\frac{\mathrm{d}u_h(t)}{\mathrm{d}t} + \mathcal{L}_h u_h(t) = f_h, \qquad R(t, u_h(t)) = f_h - \mathcal{L}_h u_h(t)$

Initial value problem $\begin{cases} \frac{\mathrm{d}u_h(t)}{\mathrm{d}t} = R(t, u_h(t)) \\ u_h(t^n) = u_h^n \end{cases}$ on (t^n, t^{n+1})

Exact integration $\int_{t^n}^{t^{n+1}} \frac{\mathrm{d}u_h}{\mathrm{d}t}\,\mathrm{d}t = u_h^{n+1} - u_h^n = \int_{t^n}^{t^{n+1}} R(t, u_h(t))\,\mathrm{d}t$

Mean value theorem: $\exists\, \tau \in (t^n, t^{n+1})$

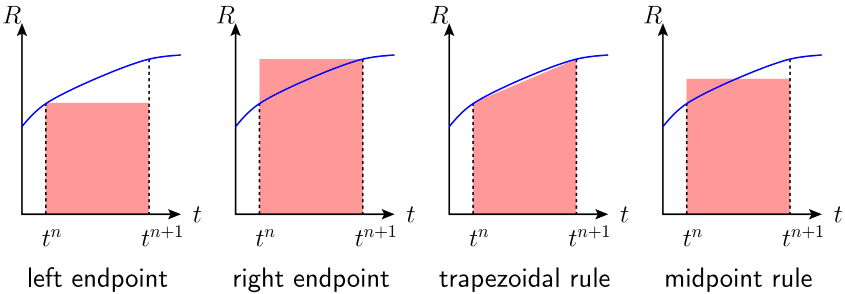
$u_h^{n+1} = u_h^n + R(\tau, u_h(\tau))\Delta t, \quad \Delta t = t^{n+1} - t^n$



Idea: evaluate the integral **numerically** using a suitable quadrature rule

Two-level time-stepping schemes

Numerical integration on the time interval (t^n, t^{n+1})



Forward Euler	$u_h^{n+1} = u_h^n + R(t^n, u_h^n)\Delta t + \mathcal{O}(\Delta t^2)$
Backward Euler	$u_h^{n+1} = u_h^n + R(t^{n+1}, u_h^{n+1})\Delta t + \mathcal{O}(\Delta t^2)$
Crank-Nicolson	$u_h^{n+1} = u_h^n + \frac{1}{2}\left[R(t^n, u_h^n) + R(t^{n+1}, u_h^{n+1})\right]\Delta t + \mathcal{O}(\Delta t^3)$
Leapfrog method	$u_h^{n+1} = u_h^n + R(t^{n+1/2}, u_h^{n+1/2})\Delta t + \mathcal{O}(\Delta t^3)$

TUDelft

Two-level θ-time-stepping schemes

Standard finite difference discretization of the time derivative

$$\frac{u_h^{n+1} - u_h^n}{\Delta t} = \theta R(t^{n+1}, u_h^{n+1}) + (1-\theta) R(t^n, u_h^n), \qquad 0 \leq \theta \leq 1$$

where $0 \leq \theta \leq 1$ is the implicitness parameter

$\theta = 0$	forward Euler	$\mathcal{O}(\Delta t)$	explicit	*update of* u_h^{n+1}
$\theta = 1/2$	Crank-Nicolson	$\mathcal{O}(\Delta t^2)$	implicit	*solution of (linear) algebraic system*
$\theta = 1$	backward Euler	$\mathcal{O}(\Delta t)$	implicit	*solution of (linear) algebraic system*

For $\theta \geq \frac{1}{2}$ the θ-scheme is A-stable ($\rightarrow$ numerical solution is bounded)

So, what it the best time-stepping method then? It depends

Properties of the θ-scheme

The optimal choice of the time-stepping scheme depends on its purpose:

- to march the numerical solution to a *steady state limit* (intermediate results are immaterial) use *large* Δt to reach convergence quickly

or

- to obtain a *time-accurate discretization* of a highly dynamic flow problem (evolution details are essential and must be captured) use *sufficiently small time steps* Δt to reduce temporal errors

Time discretization $\quad t^0 \to t^1 \to t^2 \to \dots t^n = n\Delta t \dots \to T = M\Delta t$

Accumulation of truncation errors $\quad n = 0, 1, \dots, M-1, \quad M = \frac{T}{\Delta t}$

$$\epsilon_\tau^{\text{loc}} = \mathcal{O}(\Delta t^p) \qquad \Rightarrow \qquad \epsilon_\tau^{\text{glob}} = M\epsilon_\tau^{\text{loc}} = \mathcal{O}(\Delta t^{p-1})$$

The order of a time-stepping method (i.e., the asymptotic rate at which the error is reduced as $\Delta t \to 0$) is not the sole indicator of accuracy

Explicit vs. implicit time-stepping schemes

Explicit time-stepping schemes

- easy to implement and parallelize, low cost per time step
- a good starting point for the development of CFD software
- small time steps are required for stability reasons, especially if the velocity and/or mesh size are varying strongly
- extremely inefficient for solution of stationary problems unless *local time-stepping* i.e. $\Delta t = \Delta t(\mathbf{x})$ is employed

Remark: Use the forward Euler method ($\theta = 0$) with care since it may be unconditionally unstable (e.g., for standard Galerkin in space)

Explicit vs. implicit time-stepping schemes

Implicit time-stepping schemes

- 👍 stable over a wide range of time steps, sometimes unconditionally
- 👍 constitute excellent iterative solvers for steady-state problems
- 👎 difficult to implement and parallelize, high cost per time step
- 👎 insufficiently accurate for truly transient problems at large Δt
- 👎 convergence of linear solvers deteriorates/fails as Δt increases

Galerkin method of lines

Continuous problem: find $u \in V$ such that

$$\int_\Omega v \left[\frac{\partial u}{\partial t} + \mathcal{L}u - f\right] \mathrm{d}\mathbf{x} = 0 \qquad \forall v \in V, \quad \forall t \in (t^n, t^{n+1})$$

FEM approximation $\quad u_h(\mathbf{x}, t) = \sum_{j=1}^{N} u_j(t)\varphi_j(\mathbf{x}), \quad u_i^n \approx u(\mathbf{x}_i, t^n)$

Semi-discrete problem: find $u_h \in V_h$ such that

$$\sum_{j=1}^{N} \underbrace{\left[\int_\Omega \varphi_i \varphi_j \,\mathrm{d}\mathbf{x}\right]}_{m_{ij}} \frac{\mathrm{d}u_j}{\mathrm{d}t} + \sum_{j=1}^{N} \underbrace{\left[\int_\Omega \varphi_i \mathcal{L}\varphi_j \,\mathrm{d}\mathbf{x}\right]}_{a_{ij}} u_j = \underbrace{\int_\Omega \varphi_i f \,\mathrm{d}\mathbf{x}}_{b_i}, \quad \begin{array}{l}\forall i = 1, \ldots, N, \\ \forall t \in (t^n, t^{n+1})\end{array}$$

Differential-Algebraic Equation (DAE) $\quad M_C \frac{\mathrm{d}u}{\mathrm{d}t} + Au = b$

where $M_C = \{m_{ij}\}$ is the *mass matrix* and $u(t) = [u_1(t), \ldots, u_N(t)]^T$

Galerkin method of lines

Galerkin MOL $\rightarrow$ $M_C \frac{\mathrm{d}u}{\mathrm{d}t} + Au = b, \qquad R(t,u) = M_C^{-1}[b - Au]$

$$[M_C - \theta \Delta t A]u^{n+1} = [M_C + (1-\theta)\Delta t A]u^n + \Delta t b^{n+\theta}$$

where $b^{n+\theta} = \theta b^{n+1} + (1-\theta)b^n, \quad 0 \leq \theta \leq 1, \quad n = 0, 1, \ldots, M-1$

Linear algebraic system needs to be solved even for explicit scheme ($\theta = 0$)

Idea: Replace M_C by the lumped mass matrix $M_L = \mathrm{diag}\{m_i\}$ where

$$m_i = \sum_j m_{ij} = \int_\Omega \varphi_i, \qquad \sum_j \varphi_j \, \mathrm{d}\mathbf{x} = \int_\Omega \varphi_i \, \mathrm{d}\mathbf{x} \quad \text{since} \quad \sum_j \varphi \equiv 1$$

for most finite elements. 1D case *FDM=FVM=FEM+mass lumping*

Lumped-Galerkin MOL $\rightarrow$ $M_L \frac{\mathrm{d}u}{\mathrm{d}t} + Au = b, \qquad R(t,u) = M_L^{-1}[b - Au]$

Take home lessons #2

- Due to the parabolic nature of unsteady flows, time-stepping schemes can be used to advance transient solutions step-by-step
- Space discretizations techniques yield a first-order ODE system which is integrated forward in time numerically (method of lines)
- Galerkin method of lines leads to differential-algebraic equations unless the consistent mass matrix is replaced by the lumped mass matrix
- The optimal choice of the time-stepping scheme depends on its purpose (steady-state vs. dynamic flows, explicit vs. implicit)

Part III

Multi-Step and other Advanced Methods

Fractional-step θ-scheme

Given parameters $\theta \in (0,1)$, $\gamma = 1-2\theta$, and $\alpha \in [0,1]$ subdivide time interval (t^n, t^{n+1}) into three substeps and update the solution as follows

$$\text{Step 1.} \quad u^{n+\theta} = u^n + [\alpha R(t^{n+\theta}, u^{n+\theta}) + (1-\alpha) R(t^n, u^n)]\theta\Delta t$$

$$\text{Step 2.} \quad u^{n+1-\theta} = u^{n+\theta} + [(1-\alpha) R(t^{n+1-\theta}, u^{n+1-\theta}) + \alpha R(t^{n+\theta}, u^{n+\theta})]\gamma\Delta$$

$$\text{Step 3.} \quad u^{n+1} = u^{n+1-\theta} + [\alpha R(t^{n+1}, u^{n+1}) + (1-\alpha) R(t^{n+1-\theta}, u^{n+1-\theta})]\theta\Delta t$$

Properties of this time-stepping method

- second-order accurate in the special case $\theta = 1 - \frac{\sqrt{2}}{2}$
- coefficient matrices are the same for all substeps if $\alpha = \frac{1-2\theta}{1-\theta}$
- combine the advantages of Crank-Nicolson and backward Euler

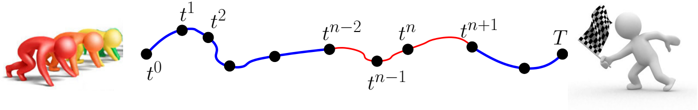
Predictor-corrector and multipoint methods

Objective: to combine the simplicity of explicit schemes and robustness of implicit ones in the framework of a fractional-step algorithm, e.g.,

1. Predictor	$\tilde{u}^{n+1} = u^n + R(t^n, u^n)\Delta t$	forward Euler
2. Corrector	$u^{n+1} = u^n + \frac{1}{2}[R(t^n, u^n) + R(t^{n+1}, \tilde{u}^{n+1})]\Delta t$	Crank-Nicolson
or	$u^{n+1} = u^n + R(t^{n+1}, \tilde{u}^{n+1})\Delta t$	backward Euler

Stability still leaves a lot to be desired, additional correction steps usually do not pay off since iterations may diverge if Δt is too large

Order barrier: two-level methods are at most second-order accurate, so extra points are needed to construct higher-order integration schemes

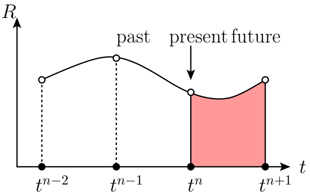


Adams methods

<u>Derivation:</u> *polynomial fitting*

Truncation error: $\epsilon_\tau^{\text{glob}} = \mathcal{O}(\Delta t^p)$

for polynomials of degree $p-1$ which interpolate function values at p points



Adams-Bashforth methods (explicit)

$p=1$ $\quad u^{n+1} = u^n + \Delta t R(t^n, u^n)$ forward Euler

$p=2$ $\quad u^{n+1} = u^n + \frac{\Delta t}{2}[3R(t^n, u^n) - R(t^{n-1}, u^{n-1})]$

$p=3$ $\quad u^{n+1} = u^n + \frac{\Delta t}{12}[23R(t^n, u^n) - 16R(t^{n-1}, u^{n-1}) + 5R(t^{n-2}, u^{n-2})]$

Adams-Moulton methods (implicit)

$p=1$ $\quad u^{n+1} = u^n + \Delta t R(t^{n+1}, u^{n+1})$ backward Euler

$p=2$ $\quad u^{n+1} = u^n + \frac{\Delta t}{2}[R(t^{n+1}, u^{n+1}) + R(t^n, u^n)]$ Crank-Nicolson

$p=3$ $\quad u^{n+1} = u^n + \frac{\Delta t}{12}[5R(t^{n+1}, u^{n+1}) + 8R(t^n, u^n) - R(t^{n-1}, u^{n-1})]$

Adams methods

Predictor-corrector algorithm

1. Compute $\tilde{u}^{n+1}$ using an Adams-Bashforth method of order $p-1$
2. Compute u^{n+1} using an Adams-Moulton method of order p with predicted value $R(t^{n+1}, \tilde{u}^{n+1})$ instead of $R(t^{n+1}, u^{n+1})$

- 👍 methods of any order are easy to derive and implement
- 👍 only one function evaluation per time step is performed
- 👍 error estimators for ODEs can be used to adapt the order
- 👎 other methods are needed to start/restart the calculation
- 👎 time step is difficult to change (coefficients are different)
- 👎 tend to be unstable and produce nonphysical oscillations

Runge-Kutta methods

Multipredictor-multicorrector algorithms of order p

$p = 2$ $\quad u^{(1)} = u^n + \frac{\Delta t}{2} R(t^n, u^n)$ forward Euler / predictor

$u^{n+1} = u^n + \Delta t R(t^n + \frac{1}{2}\Delta t, u^{(1)})$ midpoint rule / corrector

$p = 4$ $\quad u^{(1)} = u^n + \frac{\Delta t}{2} R(t^n, u^n)$ forward Euler / predictor

$u^{(2)} = u^n + \frac{\Delta t}{2} R(t^n + \frac{1}{2}\Delta t, u^{(1)})$ backward Euler / corrector

$u^{(3)} = u^n + \Delta t R(t^n + \frac{1}{2}\Delta t, u^{(2)})$ midpoint rule / predictor

$$u^{n+1} = u^n + \frac{\Delta t}{6}[R(t^n, u^n) + 2R(t^n + \tfrac{1}{2}\Delta t, u^{(1)})$$ Simpson rule

$$+ 2R(t^n + \tfrac{1}{2}\Delta t, u^{(2)}) + R(t^{n+1}, u^{(3)})]$$ corrector

Runge-Kutta methods

Another view on Runge-Kutta methods ($p = 4$)

$$u^{(1)} = u^n + \frac{\Delta t}{2} R(t^n, u^n), \qquad k^{(1)} = R(t^n, u^{(0)})$$

$$u^{(2)} = u^n + \frac{\Delta t}{2} R(t^n + 0.5\Delta t, u^{(1)}) \qquad k^{(2)} = R(t^n + 0.5\Delta t, u^{(1)})$$

$$u^{(3)} = u^n + \Delta t R(t^n + 0.5\Delta t, u^{(2)}) \qquad k^{(3)} = R(t^n + 0.5\Delta t, u^{(2)})$$

$$u^{n+1} = u^n + \frac{\Delta t}{6} [R(t^n, u^n) + 2R(t^n + 0.5\Delta t, u^{(1)}) + 2R(t^n + 0.5\Delta t, u^{(2)}) + R(t^n + \Delta t, u^{(3)})]$$

TUDelft

Runge-Kutta methods

Another view on Runge-Kutta methods ($p = 4$)

$$
\begin{aligned}
k^{(1)} &= R(t^n, u^n) \\
k^{(2)} &= R(t^n + 0.5\Delta t, u^n + \frac{\Delta t}{2} k^{(1)}) \\
k^{(3)} &= R(t^n + 0.5\Delta t, u^n + \frac{\Delta t}{2} k^{(2)}) \\
k^{(4)} &= R(t^n + 1.0\Delta t, u^n + \Delta t k^{(3)})
\end{aligned}
$$

$$
u^{n+1} = u^n + \frac{\Delta t}{6}[k^{(1)} + 2k^{(2)} + 2k^{(3)} + k^{(4)}]
$$

Runge-Kutta methods

Generic s-step explicit Runge-Kutta method

$$
\begin{aligned}
k^{(1)} &= R(t^n, u^n) \\
k^{(2)} &= R(t^n + c_2\Delta t, u^n + \Delta t(a_{21}k^{(1)})) \\
k^{(3)} &= R(t^n + c_3\Delta t, u^n + \Delta t(a_{31}k^{(1)} + a_{32}k^{(2)})) \\
&\vdots \\
k^{(s)} &= R(t^n + c_s\Delta t, u^n + \Delta t(a_{s1}k^{(1)} + a_{s2}k^{(2)} + \cdots + a_{s,s-1}k^{(s-1)}))
\end{aligned}
$$

$$
u^{n+1} = u^n + \Delta t[b_1 k^{(1)} + b_2 k^{(2)} + \cdots + b_s k^{(s)}]
$$

Runge-Kutta methods

Butcher tableau of generic s-step explicit Runge-Kutta method

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_2 & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \dots & b_{s-1} & b_s \\
 & b_1^* & b_2^* & \dots & b_{s-1}^* & b_s^*
\end{array}
$$

'Embedded' Runge-Kutta methods compute solutions of order p and $p-1$

$$
\begin{aligned}
u^{n+1} &= u^n + \Delta t[b_1 k^{(1)} + b_2 k^{(2)} + \cdots + b_s k^{(s)}] \\
u^* &= u^n + \Delta t[b_1^* k^{(1)} + b_2^* k^{(2)} + \cdots + b_s^* k^{(s)}]
\end{aligned}
$$

and estimate the error of the solution as follows

$$
e^{n+1} = u^{n+1} - u^* = (b_1 - b_1^*)k^{(1)} + (b_2 - b_2^*)k^{(2)} + \cdots + (b_s - b_s^*)k^{(s)}
$$

Runge-Kutta methods

Runge-Kutta methods

- 👍 self-starting, easy to operate with variable time steps
- 👍 more stable and accurate than Adams methods of the same order
- 👍 universal implementation via Butcher tableaux
- 👎 high order approximations are rather difficult to derive;
  p function evaluations per time step are required
- 👎 more expensive than Adams methods of comparable order

Use of variable time step sizes makes it possible to achieve
the desired accuracy at a relatively low computational cost

Explicit method: use the largest time step satisfying the stability condition

Implicit method: estimate the error and adjust the time step if necessary

Take home lessons #3

- Advantages of Crank-Nicolson and backward Euler method can be combined by using the fractional-step θ-scheme
- Second-order barrier of two-level methods can be avoided by adding extra points from previous time steps (Adams methods)
- Simple time-stepping schemes can be combined in a predictor-corrector algorithm (Runge-Kutta methods)

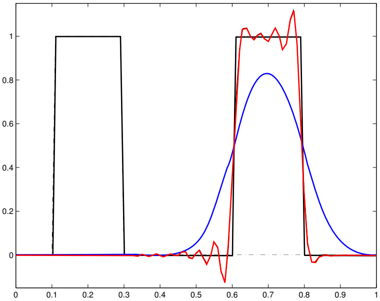
Part IV

Analysis of numerical dissipation and dispersion

Properties of numerical methods

Example: convection-dominated / hyperbolic PDEs $Pe \gg 1,\ Re \gg 1$

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0, \quad v > 0$$



High-order numerical method

- wiggles *(phase errors)*

Low-order numerical method

- smearing *(amplitude errors)*

Objective: to analyze the creation of wiggles and the smearing of solutions; design accurate methods which do not lead to spurious oscillations

Modified equation method

Motivation: PDEs are difficult or impossible to solve analytically but their *qualitative behavior* is easier to predict than that of discretized equations

Observation: exact solution (no roundoff errors) of the discretized equations satisfies a PDE which is generally different from the one to be solved

Original PDE

$$\frac{\partial u}{\partial t} + \mathcal{L}u = 0$$

Numerical method solves modified equation

$$\frac{\partial u}{\partial t} + \mathcal{L}u = \sum_{p=1}^{\infty} \alpha_{2p} \frac{\partial^{2p} u}{\partial x^{2p}} + \sum_{p=1}^{\infty} \alpha_{2p+1} \frac{\partial^{2p+1} u}{\partial x^{2p+1}}$$

1. Expand all nodal values in the difference scheme in a double Taylor series about a single point (x_i, t^n) of the space-time mesh → PDE
2. Express *high-order time derivatives* and *mixed derivatives* in this PDE in terms of **space derivatives** only → modified equation

TUDelft

Derivation of the modified equation

Example: pure convection equation $\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0,\ v > 0$ discretized by

- BDS in space
- FE in time

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + v\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \quad \text{(upwind)}$$

Taylor series expansions about the point (x_i, t^n)

$$u_i^{n+1} = u_i^n + \Delta t \left(\frac{\partial u}{\partial t}\right)_i^n + \frac{(\Delta t)^2}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_i^n + \frac{(\Delta t)^3}{6}\left(\frac{\partial^3 u}{\partial t^3}\right)_i^n + \dots$$

$$u_{i-1}^n = u_i^n - \Delta x \left(\frac{\partial u}{\partial x}\right)_i^n + \frac{(\Delta x)^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_i^n - \frac{(\Delta x)^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_i^n + \dots$$

Substitution into the difference scheme yields

$$(*) \quad \left(\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x}\right)_i^n = -\frac{\Delta t}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_i^n - \frac{(\Delta t)^2}{6}\left(\frac{\partial^3 u}{\partial t^3}\right)_i^n + \frac{v\Delta x}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_i^n - \frac{v(\Delta x)^2}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_i^n + \dots$$

Next step: Replace high-order time derivatives by space derivatives

TU Delft

Derivation of the modified equation

Differentiate $(*)$ with respect to t to obtain

$$(1)\qquad \frac{\partial^2 u}{\partial t^2} + v\frac{\partial^2 u}{\partial x \partial t} = -\frac{\Delta t}{2}\frac{\partial^3 u}{\partial t^3} - \frac{(\Delta t)^2}{6}\frac{\partial^4 u}{\partial t^4} + \frac{v\Delta x}{2}\frac{\partial^3 u}{\partial x^2 \partial t} - \frac{v(\Delta x)^2}{6}\frac{\partial^4 u}{\partial x^3 \partial t} + \dots$$

Differentiate $(*)$ with respect to x and multiply by v to obtain

$$(2)\qquad v\frac{\partial^2 u}{\partial t \partial x} + v^2\frac{\partial^2 u}{\partial x^2} = -\frac{v\Delta t}{2}\frac{\partial^3 u}{\partial t^2 \partial x} - \frac{v(\Delta t)^2}{6}\frac{\partial^4 u}{\partial t^3 \partial x} + \frac{v^2\Delta x}{2}\frac{\partial^3 u}{\partial x^3} - \frac{v^2(\Delta x)^2}{6}\frac{\partial^4 u}{\partial x^4} + \dots$$

Subtract (2) from (1) and drop high-order terms to obtain

$$(3)\qquad \frac{\partial^2 u}{\partial t^2} = v^2\frac{\partial^2 u}{\partial x^2} + \frac{\Delta t}{2}\left[-\frac{\partial^3 u}{\partial t^3} + v\frac{\partial^3 u}{\partial t^2 \partial x} + \mathcal{O}(\Delta t)\right] + \frac{\Delta x}{2}\left[v\frac{\partial^3 u}{\partial x^2 \partial t} - v^2\frac{\partial^3 u}{\partial x^3} + \mathcal{O}(\Delta x)\right]$$

Replace 3rd order time derivative and mixed derivatives by space derivatives

Derivation of the modified equation

Differentiate (2) with respect to x and divide by v to obtain

(4) $$\frac{\partial^3 u}{\partial x^2 \partial t} = -v \frac{\partial^3 u}{\partial x^3} + \mathcal{O}[\Delta t, \Delta x]$$

Differentiate (3) with respect to t and x to obtain respectively

(5) $$\frac{\partial^3 u}{\partial t^3} = v^2 \frac{\partial^3 u}{\partial x^2 \partial t} + \mathcal{O}[\Delta t, \Delta x]$$ and (6) $$\frac{\partial^3 u}{\partial t^2 \partial x} = v^2 \frac{\partial^3 u}{\partial x^3} + \mathcal{O}[\Delta t, \Delta x]$$

Equations (5) and (4) imply (7) $$\frac{\partial^3 u}{\partial t^3} = -v^3 \frac{\partial^3 u}{\partial x^3} + \mathcal{O}[\Delta t, \Delta x]$$

Plugging equations (4), (6), and (7) into (3) yields

(8) $$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2} + v^2 (v\Delta t - \Delta x) \frac{\partial^3 u}{\partial x^3} + \mathcal{O}[\Delta t, \Delta x]$$

Derivation of the modified equation

Substitute (7) and (8) into (∗) to obtain the modified equation

$$\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial x}=-\frac{v^2\Delta t}{2}\left[\frac{\partial^2 u}{\partial x^2}+(v\Delta t-\Delta x)\frac{\partial^3 u}{\partial x^3}\right]+\frac{v^3(\Delta t)^2}{6}\frac{\partial^3 u}{\partial x^3}+\frac{v\Delta x}{2}\frac{\partial^2 u}{\partial x^2}-\frac{v(\Delta x)^2}{6}\frac{\partial^3 u}{\partial x^3}+\dots$$

Reformulation in terms of the Courant number $\nu = v\frac{\Delta t}{\Delta x}$ yields

Modified equation for FE/BDS discretization of $\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial x}=0,\ v>0$

$$\underbrace{\frac{\partial u}{\partial t}+v\frac{\partial u}{\partial x}}_{\text{original PDE}}=\underbrace{\frac{v\Delta x}{2}(1-\nu)\frac{\partial^2 u}{\partial x^2}}_{\text{numerical dissipation}}+\underbrace{\frac{v(\Delta x)^2}{6}(3\nu-2\nu^2-1)\frac{\partial^3 u}{\partial x^3}}_{\text{numerical dispersion}}+\dots$$

Remark: the CFL stability condition $\nu \leq 1$ must be satisfied for the discrete problem to be well-posed. In the case $\nu > 1$, the numerical diffusion coefficient $\frac{v\Delta x}{2}(1-\nu)$ is negative, which corresponds to a *backward heat equation.*

Significance of terms in the modified equation

Exact solution of the discretized equation satisfies the modified equation

$$\frac{\partial u}{\partial t} + \mathcal{L}u = \sum_{p=1}^{\infty} \alpha_{2p} \frac{\partial^{2p} u}{\partial x^{2p}} + \sum_{p=1}^{\infty} \alpha_{2p+1} \frac{\partial^{2p+1} u}{\partial x^{2p+1}}, \qquad \text{with} \quad \mathcal{L} = v\frac{\partial}{\partial x}$$

- Even-order derivatives $\frac{\partial^{2p} u}{\partial x^{2p}}$

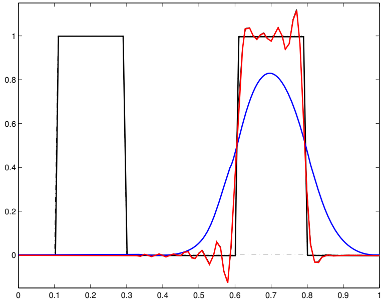
  cause numerical dissipation

  smearing (amplitude errors)

- Odd-order derivatives $\frac{\partial^{2p+1} u}{\partial x^{2p+1}}$

  cause numerical dispersion

  wiggles (phase errors)



Qualitative analysis: the numerical behavior of the discretization scheme largely depends on the relative importance of dispersive and dissipative effects

TUDelft

Forward Euler central difference scheme

Stability condition (necessary but not sufficient)

Coefficients of the even-order derivatives in the modified equation must have alternating signs, the one for the second-order term being positive

- Modified equation for the FE/CDS scheme

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = -\frac{v\Delta x}{2}\nu\frac{\partial^2 u}{\partial x^2} - \frac{v(\Delta x)^2}{6}(1+2\nu^2)\frac{\partial^3 u}{\partial x^3} + \dots$$

  where $\nu = v\frac{\Delta t}{\Delta x}$ is the Courant number

- FE/CDS scheme is unconditionally unstable since the coefficient

$$-\frac{v\Delta x}{2}\nu = -\frac{v^2\Delta t}{2} < 0 \quad \Leftrightarrow \quad v \neq 0$$

  of the second-order term is negative for nonzero velocity

TUDelft

Forward Euler finite element method

Stability condition (necessary but not sufficient)

Coefficients of the even-order derivatives in the modified equation must have alternating signs, the one for the second-order term being positive

- Modified equation for the FE/FEM scheme with $\nu = v\frac{\Delta t}{\Delta x}$

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = -\frac{v\Delta x}{2}\nu\frac{\partial^2 u}{\partial x^2} - \frac{v(\Delta x)^2}{3}\nu^2\frac{\partial^3 u}{\partial x^3} + \dots$$

- Elimination of leading dispersion error due to space discretization
- FE/FEM scheme is unconditionally unstable since the coefficient

$$-\frac{v\Delta x}{2}\nu = -\frac{v^2\Delta t}{2} < 0 \quad \Leftrightarrow \quad v \neq 0$$

of the second-order term is negative for nonzero velocity

Stabilization by means of artificial diffusion

Stabilization strategy: if the necessary stability condition is violated, it can be enforced by adding artificial diffusion to the numerical scheme:

- Stabilized methods

  $+\delta(\mathbf{v}\cdot\nabla)^2 u$ *streamline diffusion*

- Nonoscillatory methods

  $+\delta(\mathbf{v}\cdot\nabla)^2 u + \epsilon(u)\Delta u$ *shock-capturing viscosity*

Remark: in the one-dimensional case both terms are proportional to $\frac{\partial^2 u}{\partial x^2}$

Free parameters depending on the mesh size h and the residual $R(u)$

$$\delta = \frac{c_\delta h}{1+|\mathbf{v}|}, \qquad \epsilon(u) = c_\epsilon h^2 R(u)$$

Problem: how to determine proper values of the constants c_δ and c_ϵ **???**

Take home lessons #4

- The exact solution of the discretized equations does not satisfy the original PDE but the modified equation
- Even-order derivatives in the modified equation cause numerical dissipation which affects to amplitude errors ($\rightarrow$ *smearing*)
- Odd-order derivatives in the modified equation cause numerical dispersion which affects the phase errors ($\rightarrow$ *wiggles*)
- For stability, the coefficients of the even-order derivatives must have alternating sign, the one for the second-order term being positive
- Numerical schemes can be stabilized by adding artificial diffusion

Part V

Lax-Wendroff Time-Stepping Method

Lax-Wendroff method

Objective: to use a high-order time-stepping method for stabilization

Consider a time-dependent PDE $\frac{\partial u}{\partial t} + \mathcal{L}u = 0$ in $\Omega \times (0, T)$

1. Discretize it in time by means of the Taylor series expansion

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

2. Transform time derivatives into space derivatives using the PDE

$$\frac{\partial u}{\partial t} = -\mathcal{L}u, \qquad \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial t}\left(\frac{\partial u}{\partial t}\right) = \frac{\partial}{\partial t}\left(-\mathcal{L}u\right) = -\mathcal{L}\frac{\partial u}{\partial t} = \mathcal{L}^2 u$$

3. Substitute the resulting expressions into the Taylor series

$$u^{n+1} = u^n - \Delta t \mathcal{L} u^n + \frac{(\Delta t)^2}{2} \mathcal{L}^2 u^n + \mathcal{O}(\Delta t^3)$$

4. Perform space discretization using the FDM / FVM / FEM

Lax-Wendroff central difference scheme

Example: pure convection equation $\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0$

Time derivatives $\mathcal{L} = v\frac{\partial}{\partial x} \quad \Rightarrow \quad \frac{\partial u}{\partial t} = -v\frac{\partial u}{\partial x}, \qquad \frac{\partial^2 u}{\partial t^2} = v^2\frac{\partial^2 u}{\partial x^2}$

Semi-discrete scheme $u^{n+1} = u^n - v\Delta t\left(\frac{\partial u}{\partial x}\right)^n + \frac{v^2(\Delta t)^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)^n + \mathcal{O}(\Delta t^3)$

Central difference approximation (CDS) in space

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1}-u_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2), \qquad \left(\frac{\partial^2 u}{\partial x^2}\right)_i = \frac{u_{i+1}-2u_i+u_{i-1}}{(\Delta x)^2} + \mathcal{O}(\Delta x^2)$$

Fully discrete LW/CDS scheme

$$\underbrace{\frac{u_i^{n+1}-u_i^n}{\Delta t} + v\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}}_{\text{FE/CDS scheme}} = \underbrace{\frac{v^2\Delta t}{2}\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}}_{\text{numerical dissipation}} + \mathcal{O}[\Delta t^2, \Delta x^2]$$

$\Leftrightarrow$ FE/CDS scheme stabilized by artificial diffusion (no adjustable parameter)

TU Delft

Lax-Wendroff central difference scheme

Modified equation for the LW/CDS scheme

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = -\frac{v(\Delta x)^2}{6}(1-\nu^2)\frac{\partial^3 u}{\partial x^3} - \frac{v(\Delta x)^3}{8}\nu(1-\nu^2)\frac{\partial^4 u}{\partial x^4}$$
$$-\frac{v(\Delta x)^4}{120}(1+5\nu^2-6\nu^4)\frac{\partial^5 u}{\partial x^5} + \dots, \quad \nu = v\frac{\Delta t}{\Delta x}$$

- conditionally stable for $\nu^2 \leq 1$ in 1D, $\nu^2 \leq \frac{1}{8}$ in 2D, $\nu^2 \leq \frac{1}{27}$ in 3D
- the second-order derivative (leading dissipation error) has been eliminated
- the leading truncation error vanishes for $\nu^2 = 1$ (unit CFL property)
- the negative dispersion coefficient corresponds to a lagging phase error i. e. harmonics travel too slow, spurious oscillations occur *behind* steep fronts

Lax-Wendroff finite element method

Modified equation for the LW/FEM scheme

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = \frac{v(\Delta x)^2}{6}\nu^2\frac{\partial^3 u}{\partial x^3} - \frac{v(\Delta x)^3}{24}\nu(1-3\nu^2)\frac{\partial^4 u}{\partial x^4} + \frac{v(\Delta x)^4}{180}(1-\frac{15}{2}\nu^2+9\nu^4)\frac{\partial^5 u}{\partial x^5} + \dots, \qquad \nu = v\frac{\Delta t}{\Delta x}$$

- conditionally stable for $\nu^2 \leq \frac{1}{3}$ in 1D, $\nu^2 \leq \frac{1}{24}$ in 2D, $\nu^2 \leq \frac{1}{81}$ in 3D
- the second-order derivative (leading dissipation error) has been eliminated
- the truncation error does not vanish for $\nu^2 = 1$ (no unit CFL property)
- the positive dispersion coefficient corresponds to a leading phase error i. e. harmonics travel too fast, spurious oscillations occur *ahead* of steep fronts

Part VI

Taylor-Galerkin Time-Stepping Methods

Taylor-Galerkin methods

TG methods: family of high-order time-stepping schemes which stabilize the convective terms by means of intrinsic streamline diffusion (Donea '84)

Generic procedure (to be adapted for a particular scheme)

1. Taylor series expansion(s) in time up to the third/fourth order

   $u^{n+1} = u^n + \dots, \qquad u^n = u^{n+1} - \dots$ or $u^{n-1} = u^n - \dots$

2. Transform time derivatives into space derivatives avoiding third-order space derivatives. As example, let $\frac{\partial u}{\partial t} + \mathcal{L}u = 0$ then

   $$\frac{\partial u}{\partial t} = -\mathcal{L}u, \quad \frac{\partial^2 u}{\partial t^2} = \mathcal{L}^2 u, \quad \frac{\partial^3 u}{\partial t^3} = \mathcal{L}^2 \frac{\partial u}{\partial t} = \mathcal{L}^2 \frac{u^{n+1} - u^n}{\Delta t} + \mathcal{O}(\Delta t)$$

3. Substitute the resulting expressions into the Taylor series and perform space discretization using the FEM / FDM / FVM

Euler Taylor-Galerkin scheme

Consider convection-dominated PDE $\frac{\partial u}{\partial t} + \mathcal{L}u = 0$ in $\Omega \times (0, T)$

- Taylor series expansion up to the third order

$$u^{n+1} = u^n + \Delta t \left(\frac{\partial u}{\partial t}\right)^n + \frac{(\Delta t)^2}{2} \left(\frac{\partial^2 u}{\partial t^2}\right)^n + \frac{(\Delta t)^3}{6} \left(\frac{\partial^3 u}{\partial t^3}\right)^n + \mathcal{O}(\Delta t^3)$$

- Substitution of $\frac{\partial u}{\partial t} = -\mathcal{L}u$, $\frac{\partial^2 u}{\partial t^2} = \mathcal{L}^2 u$, $\frac{\partial^3 u}{\partial t^3} = \mathcal{L}^2 \frac{u^{n+1} - u^n}{\Delta t} + \mathcal{O}(\Delta t)$

  yields the semi-discrete FE/TG scheme

$$u^{n+1} = u^n - \Delta t \mathcal{L} u^n + \frac{(\Delta t)^2}{2} \mathcal{L}^2 u^n + \frac{(\Delta t)^2}{6} \mathcal{L}^2 (u^{n+1} - u^n) + \mathcal{O}(\Delta t^4)$$

The Lax-Wendroff scheme is recovered for $u^{n+1} = u^n$ (steady state)

Equivalent form

$$\left[\mathcal{I} - \frac{(\Delta t)^2}{6} \mathcal{L}^2\right] \frac{u^{n+1} - u^n}{\Delta t} = -\mathcal{L} u^n + \frac{\Delta t}{2} \mathcal{L}^2 u^n$$

Euler Taylor-Galerkin scheme

Example: pure convection equation $\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = 0, \quad \mathcal{L} = v\frac{\partial}{\partial x}$

discretized in space by the Galerkin FEM with linear elements

Modified equation for the FE/TG scheme

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} = -\frac{v(\Delta x)^3}{24}\nu(1-\nu^2)\frac{\partial^4 u}{\partial x^4} + \frac{v(\Delta x)^4}{180}(1-5\nu^2+4\nu^4)\frac{\partial^5 u}{\partial x^5} + \dots \qquad \nu = v\frac{\Delta t}{\Delta x}$$

- conditionally stable for $\nu^2 \le 1$ in 1D, $\nu^2 \le \frac{1}{8}$ in 2D, $\nu^2 \le \frac{1}{27}$ in 3D
- the leading truncation error vanishes for $\nu^2 = 1$ (unit CFL property)
- the leading dispersion error is of higher order than that for LW/FEM

Take home lessons #5

- Numerical methods can be stabilized using high-order time-stepping schemes which introduce artificial diffusion (no free parameter)
- Such methods can be constructed by the three-step approach:
  - discretize the PDE in time by a Taylor series expansion
  - replace time derivatives by space derivatives using the PDE
  - perform discretization in space by FDM / FVM / FEM
- Lax-Wendroff methods involve space derivative which may be significantly higher than in the original PDE (e.g. $\frac{\partial^3 u}{\partial t^3} = -\mathcal{L}^3 u$)
- Taylor-Galerkin methods avoid higher-order space derivatives but their derivation needs to be adapted for each particular scheme

Part VII

Adaptive Time-Step Control

Automatic time step control

Objective: to make sure that $\|u - u_{\Delta t}\| \approx TOL$ (prescribed tolerance)

Assume that the error at $t = t^n$ is equal to zero (no accumulation)

Local truncation error

- $u_{\Delta t} = u + \Delta t^2 e(u) + \mathcal{O}(\Delta t^4)$
- $u_{m\Delta t} = u + m^2 \Delta t^2 e(u) + \mathcal{O}(\Delta t^4)$

Heuristic error analysis

$$e(u) \approx \frac{u_{m\Delta t} - u_{\Delta t}}{\Delta t^2 (m^2 - 1)}$$

Estimate of the relative error

$$\|u - u_{\Delta t_*}\| \approx \left(\frac{\Delta t_*}{\Delta t}\right)^2 \frac{\|u_{m\Delta t} - u_{\Delta t}\|}{m^2 - 1} = TOL$$

Adaptive time stepping

$$\Delta t_*^2 = TOL \frac{\Delta t^2 (m^2 - 1)}{\|u_{\Delta t} - u_{m\Delta t}\|}$$

Richardson extrapolation

$$u = \frac{m^2 u_{\Delta t} - u_{m\Delta t}}{m^2 - 1} + \mathcal{O}(\Delta t^4)$$

Practical implementation

Given the old solution u^n do:

1. Make one large time step of size $m\Delta t$ to compute $u_{m\Delta t}$
2. Make m small time steps each of size Δt to compute $u_{\Delta t}$
3. Evaluate the relative solution changes $\|u_{\Delta t} - u_{m\Delta t}\|$
4. Calculate the 'optimal' value Δt_* for the next time step
5. If $\Delta t_* \ll \Delta t$, reset the solution and go back to the beginning
6. Set $u^{n+1} = u_{\Delta t}$ or perform Richardson extrapolation

- 👍 enhance robustness and overall efficiency of the code
- 👍 simulation results are more credible due to estimated error
- 👎 cost per time step increases substantially ($u_{m\Delta t}$ may be as expensive to obtain as $u_{\Delta t}$ due to slow convergence at large time steps)

Take home lessons #6

- Adaptive time-stepping enhances the robustness of the code, the overall efficiency and the credibility of simulation results
- Automatic time step control is based on local truncation error analysis