

CMEA II

GianAndrea Müller / Pascal Müller

January 15, 2018

CONTENTS

| | | | | | |
|--|----------|---|-----------|--|--|
| 1 ODE | 2 | | | | |
| 1.1 Definitions | 2 | | | | |
| 1.2 IVP | 2 | | | | |
| 1.2.1 Single Body Dynamics | 2 | | | | |
| 1.2.2 Trick: Vectorization | 2 | | | | |
| 1.2.3 N-Body Dynamics | 2 | | | | |
| 1.3 Matrix exponential technique | 2 | | | | |
| 2 Numerical Methods for ODE | 2 | | | | |
| 2.0.1 Time discretization | 2 | | | | |
| 2.1 Forward Euler Method (Explicit) | 2 | | | | |
| 2.2 Backward Euler Method (Implicit) | 2 | | | | |
| 2.3 Trapezoidal Method | 2 | | | | |
| 2.4 Mid-point Rule | 2 | | | | |
| 2.5 Truncuation error | 3 | | | | |
| 2.5.1 Example: Forward Euler method | 3 | | | | |
| 2.5.2 One-Step Error | 3 | | | | |
| 2.5.3 Global Error | 3 | | | | |
| 2.6 Newtons Method | 3 | | | | |
| 3 Higher-Order Methods for ODEs | 3 | | | | |
| 3.1 Runge-Kutta-2 (RK-2) | 3 | | | | |
| 3.1.1 Order of Accuracy | 3 | | | | |
| 3.2 Runge-Kutta-4 (RK-4) | 3 | | | | |
| | | 3.3 Runge-Kutta-N (RK-N) | 3 | | |
| | | 3.3.1 Explicit RK | 3 | | |
| | | 3.3.2 Diagonally Implicit RK (DIRK) | 4 | | |
| | | 4 Multi-Step Methods | 4 | | |
| | | 4.1 Adam's Method | 4 | | |
| | | 4.2 Adams-Bashforth Method | 4 | | |
| | | 4.2.1 Approach to find β_j | 4 | | |
| | | 4.3 Adams-Moulton Methods | 4 | | |
| | | 4.4 Truncation Error | 4 | | |
| | | 4.5 Starting Values | 4 | | |
| | | 4.6 Discussion of MS Methods | 4 | | |
| | | 5 Stability of Numerical Methods for ODEs | 5 | | |
| | | 5.1 Convergence of Forward Euler for Linear ODEs | 5 | | |
| | | 5.2 Convergence of Forward Euler for Non-Linear ODEs | 5 | | |
| | | 5.3 Convergence of Consistent One-Step Methods | 5 | | |
| | | 5.4 Absolute Stability | 5 | | |
| | | 5.4.1 Absolute Stability of BE | 5 | | |
| | | 5.4.2 Absolute Stability of the Trapezoidal Rule | 5 | | |
| | | 5.5 Stiff Problems | 5 | | |
| | | 6 The Poisson Equation | 5 | | |
| | | 6.0.1 Derivation of the Poissonequation | 5 | | |
| | | 6.1 Poisson Equation in 1D | 6 | | |
| | | 6.2 Finite Difference Method | 6 | | |
| | | 6.2.1 Discretization | 6 | | |
| | | 6.2.2 Finite difference scheme | 6 | | |
| | | 6.2.3 Numerical Results | 6 | | |
| | | 6.3 FDS for 2-D Poisson Equation | 6 | | |
| | | 6.3.1 Discretization | 6 | | |
| | | 6.3.2 Finite Difference Scheme | 6 | | |
| | | 7 FEM for 1D-Poisson | 7 | | |
| | | 7.1 Variational Principle | 7 | | |
| | | 7.2 Variational Formulation | 7 | | |
| | | 7.3 FEM Formulation | 7 | | |
| | | 7.4 Concrete Realisation of FEM | 8 | | |
| | | 7.5 Convergence Analysis | 8 | | |
| | | 8 FEM for 2D-Poisson | 8 | | |
| | | 8.1 Variational Formulation | 8 | | |
| | | 8.2 FEM Formulation | 8 | | |
| | | 8.3 Concrete Realisation of FEM | 8 | | |
| | | 9 Implementation of the FEM | 9 | | |
| | | 9.1 Step 1: Triangulation | 9 | | |
| | | 9.2 Step 2: Element Stiffness Matrices and Element Load Vectors | 9 | | |
| | | 9.3 Step 3: Assembly | 9 | | |
| | | 10 Parabolic PDE | 9 | | |
| | | 10.1 Seperation of Variables | 10 | | |
| | | 10.1.1 Application of Gauss-Green | 10 | | |
| | | 10.2 Energy Estimate | 10 | | |
| | | 10.3 Uniqueness of a solution | 10 | | |
| | | 10.4 Maximum Principles | 10 | | |
| | | 10.5 Explicit Finite Difference Schemes for the Heat Equation | 10 | | |
| | | 10.5.1 Simplified notation | 10 | | |
| | | 10.5.2 Truncation Error | 11 | | |
| | | 10.6 An Implicit Finite Difference Scheme | 11 | | |
| | | 10.7 Crank-Nicloson Scheme | 11 | | |
| | | 10.7.1 Truncation error | 11 | | |
| | | 10.8 Convergence Studies | 11 | | |
| | | 11 Linear Transport Equations (Hyperbolic PDEs) | 11 | | |
| | | 11.1 Method of Characteristics | 11 | | |
| | | 11.1.1 Gronwall's inequality | 11 | | |
| | | 11.2 Finite difference schemes | 12 | | |
| | | 11.2.1 Discretization | 12 | | |
| | | 11.2.2 Upwind schemes | 12 | | |
| | | 11.2.3 Stability for the upwind scheme | 12 | | |

1 ODE

Differential equation: $\mathbb{F}(t, u, u_t, u_{tt}, \dots) = 0$

1.1 DEFINITIONS

| | |
|------------------------|-----------------------------|
| Autonomous $F(U(t))$ | Non-Autonomous $F(U(t), t)$ |
| F does not depend on t | $u'(t) = u(t) + t$ |
| Linear | Non-Linear |
| F is linear in $u(t)$ | $u'(t) = \sin(u(t))$ |

1.2 IVP

To find a specific solution to an ODE of degree k we need $(k-1)$ initial values.

$$\mathbb{F}(t, u, u', u'', u^{(3)}, \dots, u^{(k)}) = 0 \quad \text{with} \quad u^{(i)}(0) = u_i \quad \forall i$$

1.2.1 SINGLE BODY DYNAMICS

$$\begin{aligned} x'(t) &= v(t), & x(0) &= x_0 \\ v'(t) &= f, & v(0) &= v_0 \end{aligned}$$

$$\begin{aligned} U &= [x, v], & F &= [v, f] \\ U' &= F(t, U), & U(0) &= [x_0, v_0] \end{aligned}$$

1.2.2 TRICK: VECTORIZATION

begin with $\theta''(t) = -\frac{g}{L} \sin(\theta(t))$

$$\vec{u}(t) = \begin{pmatrix} \theta(t) \\ \theta'(t) \end{pmatrix} \quad \vec{u}'(t) = \begin{pmatrix} \theta'(t) \\ -\sin(\theta(t)) \end{pmatrix} = \vec{F}(t, \vec{u}(t))$$

1.2.3 N-BODY DYNAMICS

$$\begin{aligned} x'_i &= v_i, & i &= 1, 2, \dots, N, \\ m_i v'_i &= f_i, & i &= 1, 2, \dots, N, \\ f_i &= \sum_{j=1, j \neq i}^N \frac{G m_i m_j (x_i - x_j)}{|x_i - x_j|^3} & \text{G - Gravitational constant} \end{aligned}$$

$$U = [x_1, \dots, x_N, v_1, \dots, v_N]$$

$$F = [v_1, \dots, v_N, f_1, \dots, f_N]$$

$$U'(t) = F(U(t)), \quad U(0) = [x_1^0, \dots, x_N^0, v_1^0, \dots, v_N^0]$$

1.3 MATRIX EXPONENTIAL TECHNIQUE

$$u'(t) = Au(t)$$

$$u(0) = u_0$$

Diagonalise $A = R\Lambda R^{-1}$

Then $(R^{-1}u(t))' = R^{-1}(R\Lambda R^{-1})u(t) = \Lambda R^{-1}u(t)$

Now let $w = R^{-1}u$

$$w'(t) = \Lambda w(t)$$

$$w(0) = R^{-1}u_0$$

Thus $w_i(t) = w_i(0)e^{\lambda_i t}$ and $w(t) = e^{\Lambda t}w(0) = e^{\Lambda t}R^{-1}u_0$.

$$\begin{aligned} R^{-1}u(t) &= e^{\Lambda t}R^{-1}u_0 \\ u(t) &= Re^{\Lambda t}R^{-1}u_0 = e^{At}u_0 \end{aligned}$$

$$e^{At} = I + At + \frac{1}{2}A^2t^2 + \dots + \frac{1}{k!}A^k t^k + \dots$$

2 NUMERICAL METHODS FOR ODE

$$u'(t) = F(t, u(t)),$$

$$u(0) = u_0$$

$u_0 \in \mathbb{R}^m$ is a constant, $u : [0, T] \rightarrow \mathbb{R}^m$ and $F : [0, T] \times \mathbb{R}^m \rightarrow \mathbb{R}^m$

2.0.1 TIME DISCRETIZATION

$[0, T] = \cup_{n=0}^{N-1} [t^n, t^{n+1})$ where $t^n = n\Delta t$ and $\Delta t = \frac{T}{N}$

Approximation of the solution: $\{U_n\}_{n=0}^N \quad U_n \approx u(t^n)$

2.1 FORWARD EULER METHOD (EXPLICIT)

$$u'(t^n) \approx \frac{u(t^{n+1}) - u(t^n)}{\Delta t} \approx \frac{U_{n+1} - U_n}{\Delta t}$$

Good approximation if Δt is small

$$u'(t) = F(t, u(t)) \implies \frac{U_{n+1} - U_n}{\Delta t} = F(t^n, U_n)$$

Solving for U^{n+1} yields:

$$U_{n+1} = U_n + \Delta t F(t^n, U_n) \quad U_0 = u_0$$

Algorithm:

$$U_0 = u_0$$

for $(n = 0; n < N; n++)$ **do**

$$U_{n+1} = U_n + \Delta t F(t^n, U_n)$$

end for

2.2 BACKWARD EULER METHOD (IMPLICIT)

$$u'(t^{n+1}) \approx \frac{u(t^{n+1}) - u(t^n)}{\Delta t} \approx \frac{U_{n+1} - U_n}{\Delta t}$$

$$U_{n+1} - \Delta t F(t^{n+1}, U_{n+1}) = U_n \quad U_0 = u_0$$

Solving for U_{n+1} we notice that the result still depends on $F(t^{n+1}, U_{n+1})!$

Advantage to Forward Euler: Stability. **Disadvantage:** More computation necessary.

2.3 TRAPEZOIDAL METHOD

$$u(t^{n+1}) - u(t^n) = \int_{t^n}^{t^{n+1}} u'(s) ds = \int_{t^n}^{t^{n+1}} F(s, u(s)) ds$$

Fundamental Theorem of Calculus

$$\int_a^b f(x) dx \approx (b-a) \left(\frac{f(a)+f(b)}{2} \right) \quad \text{Trapezoidal rule}$$

$$\frac{U_{n+1} - U_n}{\Delta t} = \frac{1}{2} (F(t^n, U_n) + F(t^{n+1}, U_{n+1}))$$

$$U_{n+1} = u_n + \frac{\Delta t}{2} F(t^n, U_n) + \frac{\Delta t}{2} F(t^{n+1}, U_{n+1})$$

2.4 MID-POINT RULE

$$u'(t^n) \approx \frac{u(t^{n+1}) - u(t^{n-1})}{2\Delta t} \approx \frac{U_{n+1} - U_{n-1}}{2\Delta t}$$

$$\frac{U_{n+1} - U_{n-1}}{2\Delta t} = F(t^n, U_n)$$

$$U_{n+1} = U_{n-1} + 2\Delta t F(t^n, U_n) \quad U_0 = u_0$$

In the beginning we only have one initial value, but need two. Possible solution: $U_1 = u_0 + \Delta F(0, u_0)$ (Forward Euler method)

2.5 TRUNCTUATION ERROR

The truncation error is the error made when the exact solution is inserted in the consisten form of numerical methods.

2.5.1 EXAMPLE: FORWARD EULER METHOD

$$T_n = \frac{u(t^{n+1}) - u(t^n)}{\Delta t} - F(t^n, u(t^n))$$

Taylor:

$$\frac{u(t^n) - u(t^n) + \Delta t u'(t^n) + \frac{(\Delta t)^2}{2} u''(t^n) + \mathcal{O}((\Delta t)^3)}{\Delta t} - F(t^n, u(t^n))$$

$$\underbrace{u'(t^n) - F(t^n, u(t^n))}_{=0 \text{ using IVP}} + \frac{\Delta t}{2} u''(t^n) + \mathcal{O}((\Delta t)^2) = \mathcal{O}(\Delta t)$$

Therefore the truncation error is of $\mathcal{O}(\Delta t)$.

$$T_n^{Midpoint} = \mathcal{O}(\Delta t^2) = T_n^{trapezoidal}$$

$$T_n^{BackwardEuler} = \mathcal{O}(\Delta t)$$

2.5.2 ONE-STEP ERROR

The one step error, is the error discovered when inserting the exact solution into the update for a given method.

2.5.3 GLOBAL ERROR

$$E_N := u(t^N) - U_N \leq \sum_{j=0}^{N-1} |L_j|$$

$$E_N \approx \mathcal{O}((\Delta t)^{q+1} \cdot N = \mathcal{O}((\Delta t)^{q+1} \cdot \frac{T}{\Delta t} \approx \mathcal{O}((\Delta t)^q))$$

2.6 NEWTONS METHOD

$$\begin{aligned} \vec{G}(x) &= 0 \\ D\vec{G}(x_k)\Delta x_k &= -\vec{G}(x_k) \\ x_{k+1} &= x_k + \Delta x_k \end{aligned}$$

3 HIGHER-ORDER METHODS FOR ODES

3.1 RUNGA-KUTTA-2 (RK-2)

RK-„2“ → 2-stage method

$$u'(t) = F(t, u(t))$$

$$u(0) = u_0$$

Use the mid-point rule, approximating $u(t^n + \frac{\Delta t}{2})$ with Forward-Euler.

$$\begin{aligned} Y_1 &= U_n \\ Y_2 &= U_n + \frac{\Delta t}{2} F(t^n, Y_1) \\ U_{n+1} &= U_n + \Delta t F(t^n + \frac{\Delta t}{2}, Y_2) \\ U_0 &= u_0 \end{aligned}$$

in the update-form:

$$\begin{aligned} U_{n+1} &= U_n + \Delta t F(t^n + \frac{\Delta t}{2}, U_n + \frac{\Delta t}{2} F(t^n, U_n)) \\ U_0 &= u_0 \end{aligned}$$

3.1.1 ORDER OF ACCURACY

1. missing

UNFINISHED - SEE SCRIPT

3.2 RUNGE-KUTTA-4 (RK-4)

$$\begin{aligned} Y_1 &= U_n \\ Y_2 &= U_n + \frac{\Delta t}{2} F(t^n, Y_1) \\ Y_3 &= U_n + \frac{\Delta t}{2} F(t^n + \frac{\Delta t}{2}, Y_2) \\ Y_4 &= U_n + \frac{\Delta t}{2} F(t^n + \frac{\Delta t}{2}, Y_3) \\ U_{n+1} &= U_n + \frac{\Delta t}{6} (F(t^n, Y_1) + 2F(t^n + \frac{\Delta t}{2}, Y_2) \\ &\quad + 2F(t^n + \frac{\Delta t}{2}, Y_3) + F(t^n + \Delta t, Y_4)) \\ U_0 &= u_0 \end{aligned}$$

3.3 RUNGE-KUTTA-N (RK-N)

$$\begin{aligned} Y_1 &= U_n + \Delta t \sum_{j=1}^s a_{1j} F(t^n + c_j \Delta t, Y_j) \\ Y_2 &= U_n + \Delta t \sum_{j=1}^s a_{2j} F(t^n + c_j \Delta t, Y_j) \\ &\vdots \\ Y_s &= U_n + \Delta t \sum_{j=1}^s a_{sj} F(t^n + c_j \Delta t, Y_j) \\ U_{n+1} &= U_n + \Delta t \sum_{j=1}^s b_j F(t^n + c_j \Delta t, Y_j) \\ U_0 &= u_0 \end{aligned}$$

The Runge-Kutta Method can be represented by the Butcher Tableau.

| | | | | |
|----------|----------|----------|----------|----------|
| c_1 | a_{11} | a_{12} | \cdots | a_{1s} |
| c_2 | a_{21} | a_{22} | \cdots | a_{2s} |
| \vdots | \vdots | \vdots | | \vdots |
| c_s | a_{s1} | a_{s2} | \cdots | a_{ss} |
| | b_1 | b_2 | \cdots | b_s |

$$\sum_{j=1}^s a_{ij} = c_i \quad \sum_{j=1}^s b_j = 1$$

If the conditions above hold, RK is consistent.

$$\sum_{j=1}^s b_j c_j = \frac{1}{2}$$

If the condition above holds, RK is second order accurate.

$$\begin{aligned} \sum_{j=1}^s b_j c_j^2 &= \frac{1}{3} \\ \sum_{j=1}^s \sum_{i=1}^s b_i a_{ij} c_j &= \frac{1}{6} \end{aligned}$$

If the conditions above hold, RK is third order accurate.

3.3.1 EXPLICIT RK

$$a_{ij} = 0 \text{ if } j \geq i \quad \text{A strictly lower triangular.}$$

If the condition above holds, RK is an explicit method.

For explicit methods only function evaluations are made. No inverted matrices.

$$U_n = Y_1 \mapsto Y_2 \mapsto \dots \mapsto Y_s \mapsto U_{n+1}$$

$$Y_i = f(Y_1, \dots, Y_{i-1})$$

3.3.2 DIAGONALLY IMPLICIT RK (DIRK)

$$a_{ij} = 0 \text{ if } j > i, \quad a_{ii} \neq 0 \text{ for some } i \quad \text{A lower triangular.}$$

If the condition above holds, RK is diagonally implicit.

Requieres the solution of nonlinear equations.

$$Y_i - \Delta a_{ii} F(t^n + c_i \Delta t, Y_i) = U_n + \Delta t \sum_{j=1, j < i}^s a_{ij} F(t^n + c_i \Delta t, Y_i)$$

$$Y_i = f(Y_1, \dots, Y_{i-1}, Y_i)$$

4 MULTI-STEP METHODS

$$\sum_{j=0}^{\gamma} \alpha_j U_{n+j} = \Delta t \sum_{j=0}^{\gamma} \beta_j F(t^{n+j}, U_{n+j}) \quad \text{linear multi-step method}$$

The method is explicit if $\beta_\gamma = 0$.

4.1 ADAM'S METHOD

$$\begin{aligned} \alpha_\gamma &= 1 \\ \alpha_{\gamma-1} &= -1 \\ \alpha_j &= 0 \end{aligned}$$

$$U_{n+\gamma} = U_{n+\gamma-1} + \Delta t \sum_{j=0}^{\gamma} \beta_j F(t^{n+j}, U_{n+j})$$

If focussing on a autonomous ODE ($F(t, u(t)) = F(u)$) we obtain:

$$U_{n+\gamma} = U_{n+\gamma-1} + \Delta t \sum_{j=0}^{\gamma} \beta_j F(U_{n+j})$$

4.2 ADAMS-BASHFORTH METHOD

$$\beta_\gamma = 0$$

Thus making Adam's Method explicit.

$$U_{n+\gamma} = U_{n+\gamma-1} + \Delta t \sum_{j=0}^{\gamma-1} \beta_j F(U_{n+j})$$

4.2.1 APPROACH TO FIND β_j

Try to solve:

$$u(t^{n+\gamma}) - u(t^{n+\gamma-1}) = \int_{t^{n+\gamma-1}}^{t^{n+\gamma}} u'(s) ds = \int_{t^{n+\gamma-1}}^{t^{n+\gamma}} F(u(s)) ds$$

The integral can then be approximated using numerical quadrature rules:

$$\begin{aligned} (AB1) \quad U_{n+1} &= U_n + \Delta t F(U_n) \\ (AB2) \quad U_{n+2} &= U_{n+1} + \frac{\Delta t}{2} (-F(U_n) + 3F(U_{n+1})) \\ (AB3) \quad U_{n+3} &= U_{n+2} + \frac{\Delta t}{12} (5F(U_n) - 16F(U_{n+1}) + 23F(U_{n+2})) \end{aligned}$$

4.3 ADAMS-MOULTON METHODS

$$\beta_\gamma \neq 0$$

Thus making Adam's Method implicit.

The resulting methods are $(\gamma + 1)$ -order accurate.

$$\begin{aligned} (AM1) : U_{n+1} &= U_n + \frac{\Delta t}{2} (F(U_n) + F(U_{n+1})) \\ (AM2) : U_{n+2} &= U_{n+1} + \frac{\Delta t}{12} (-F(U_n) + 8F(U_{n+1}) + 5F(U_{n+2})) \\ (AM3) : U_{n+3} &= U_{n+2} + \frac{\Delta t}{24} (F(U_n) - 5F(U_{n+1}) + 19F(U_{n+2}) \\ &\quad + 9F(U_{n+3})) \end{aligned}$$

4.4 TRUNCATION ERROR

$$\begin{aligned} T_{n+\gamma} &= \frac{1}{\Delta t} \left(\sum_{j=0}^{\gamma} \alpha_j u(t^{n+j}) - \Delta t \sum_{j=0}^{\gamma} \beta_j F(u(t^{n+j})) \right) \\ &= \frac{1}{\Delta t} \left(\sum_{j=0}^{\gamma} \alpha_j u(t^{n+j}) - \Delta t \sum_{j=0}^{\gamma} \beta_j u'(t^{n+j}) \right) \end{aligned}$$

Substitute with taylor expansions:

$$\begin{aligned} u(t^{n+j}) &= u(t^n) + j\Delta t u'(t^n) + \frac{j^2 \Delta t^2}{2} u''(t^n) + \dots + \frac{j^k \Delta t^k}{k!} u^{(k)}(t^n) + \dots \\ u'(t^{n+j}) &= u'(t^n) + j\Delta t u''(t^n) + \frac{j^2 \Delta t^2}{2} u'''(t^n) + \dots + \frac{j^k \Delta t^k}{k!} u^{(k+1)}(t^n) + \dots \end{aligned}$$

to end up with:

$$\begin{aligned} T_{n+\gamma} &= \frac{1}{\Delta t} \left(\sum_{j=0}^{\gamma} \alpha_j u(t^n) + \left(\sum_{j=0}^{\gamma} j(\alpha_j - \beta_j) \right) u'(t^n) \right. \\ &\quad \left. + \Delta t \left(\sum_{j=0}^{\gamma} \left(\frac{j^2 \alpha_j}{2} - j\beta_j \right) u''(t^n) + \dots \right. \right. \\ &\quad \left. \left. \Delta t^{k-1} \left(\sum_{j=0}^{\gamma} \left(\frac{j^k \alpha_j}{k!} - \frac{j^{k-1} \beta_j}{(k-1)!} \right) \right) u^{(k)}(t^n) \right) \right) \end{aligned}$$

$$\sum_{j=0}^{\gamma} \alpha_j = 0 \quad \sum_{j=0}^{\gamma} j\alpha_j = \sum_{j=0}^{\gamma} \beta_j \quad \text{Consistency}$$

$$\sum_{j=0}^{\gamma} \frac{j^q}{q!} \alpha_j = \sum_{j=0}^{\gamma} \frac{j^{q-1}}{(q-1)!} \beta_j, \quad \forall q \leq k+1 \quad \text{Truncation error of order } \Delta t^k$$

4.5 STARTING VALUES

A γ -step method requires γ starting values $U_0, U_1, \dots, U_{\gamma-1}$.

The initial condition from the IVP is thus used to find the other U_i via a RK method.

| | |
|-------------------------|--------------------------|
| explicit Adams-Bashfort | RK of order $\gamma - 1$ |
| implicit Adams-Moulton | RK of order γ |

In both cases we use an RK with one order of accuracy less than the multi-step method. In the case of AB, RK has a one-step error $\mathcal{O}(\Delta t^\gamma)$ yielding a total error $(\gamma - 1) \cdot \mathcal{O}(\Delta t^\gamma)$ therefore the global error of the multi-step method is still $\mathcal{O}(\Delta t^\gamma)$.

4.6 DISCUSSION OF MS METHODS

- **Advantage:** The right hand side Function F only needs to be evaluated once each time step. In contrast RK methods need multiple evaluations.
- **Disadvantage:** Variable time-steps are difficult to implement. In addition they require several starting values.

5 STABILITY OF NUMERICAL METHODS FOR ODES

$$u'(t) = F(t, u(t))$$

$$u(0) = u_0$$

$$\lim_{\substack{\Delta t \rightarrow 0 \\ N \cdot \Delta t = T}} U_N = u(T) \quad \text{Convergence condition}$$

For a one-step method, the starting value coincides with the initial condition. For multi-step methods we need the condition:

$$\lim_{\Delta t \rightarrow 0} U_j(\Delta t) = u_0 \quad \text{Convergence of starting values}$$

5.1 CONVERGENCE OF FORWARD EULER FOR LINEAR ODES

$$u'(t) = \lambda u(t) + g(t)$$

$$u(0) = u_0$$

$$U_{n+1} = U_n + \Delta t(\lambda U_n + g(t^n))$$

$$= (1 + \lambda \Delta t)U_n + \Delta t g(t^n)$$

$$U_0 = u_0$$

With g as a disturbance.

We want to calculate the error: $E_N = u(t^N) - U_N \quad T = N\Delta t$

$$|E_N| \leq T e^{|\lambda|T} \mathcal{O}(\Delta t) \Rightarrow \lim_{\Delta t \rightarrow 0} E_N \rightarrow 0$$

Thus FE is convergent and first order accurate for linear ODEs.

5.2 CONVERGENCE OF FORWARD EULER FOR NON-LINEAR ODES

$$u'(t) = F(u(t))$$

$$u(0) = u_0$$

$$U_{n+1} = U_n + \Delta t F(U_n)$$

$$U_0 = u_0$$

Assuming that the ODE is autonomous and the function F is Lipschitz continuous.

$$||E_N|| \leq T e^{LT} \cdot \mathcal{O}(\Delta t) \Rightarrow \lim_{\Delta t \rightarrow 0} E_N = 0$$

Thus FE is convergent for the general IVP.

5.3 CONVERGENCE OF CONSISTENT ONE-STEP METHODS

$$U_{n+1} = U_n + \Delta t \Phi(U_n, t^n, \Delta t)$$

$$U_0 = u_0$$

Truncation of a consistent one-step methods:

$$T_n := \frac{u(t^{n+1}) - u(t^n)}{\Delta t} - \Phi(u(t^n), t^n, \Delta t)$$

$$||E_N|| \leq T e^{LT} \cdot \mathcal{O}(\Delta t) \Rightarrow \lim_{\Delta t \rightarrow 0} E_N = 0$$

Thus general explicit one-step methods are convergent if they are consistent.

PLACEHOLDER 5.4 WHY CONVERGENCE IS NOT ENOUGH

5.4 ABSOLUTE STABILITY

$$u'(t) = \lambda u(t)$$

$$u(0) = u_0$$

$$U_{n+1} = (1 + \lambda \Delta t)U_n$$

$$|1 + \lambda \Delta t| \leq 1 \quad |U_{n+1}| \leq |U_n| \quad \text{Absolute stability}$$

This makes only sense if $\lambda \leq 0$.

5.4.1 ABSOLUTE STABILITY OF BE

$$\frac{U_{n+1} - U_n}{\Delta t} \Rightarrow U_{n+1} = \frac{1}{1 - \lambda \Delta t} U_n$$

$$\left| \frac{1}{1 - \lambda \Delta t} \right| \leq 1 \Rightarrow \lambda \Delta t \in (-\infty, 0] \cup [2, \infty) \quad \text{Absolute stability}$$

BE remains stable even for large time steps.

5.4.2 ABSOLUTE STABILITY OF THE TRAPEZOIDAL RULE

$$\frac{U_{n+1} - U_n}{\Delta t} = \frac{1}{2}(\lambda U_n + \lambda U_{n+1}) \rightarrow U_{n+1} = \frac{1 + \frac{\lambda \Delta t}{2}}{1 - \frac{\lambda \Delta t}{2}} U_n$$

$$\left| \frac{1 + \frac{\lambda \Delta t}{2}}{1 - \frac{\lambda \Delta t}{2}} \right| \leq 1 \quad \text{Absolute stability}$$

5.5 STIFF PROBLEMS

PDE IN GENERAL

$$\mathbb{F}(x, u, \nabla u, D^2 u, D^k u, \dots) = 0$$

$$u_{x_i} = \frac{\partial u}{\partial x_i} \quad u_{x_i, x_j} = \frac{\partial^2 u}{\partial x_i \partial x_j} \quad \text{Partial derivative}$$

$$\nabla u = (u_{x_1}, u_{x_2}, \dots, u_{x_n}) \quad \text{Gradient}$$

$$D^2 u = \begin{pmatrix} u_{x_1, x_1} & \dots & u_{x_1, x_n} \\ \vdots & & \vdots \\ u_{x_n, x_1} & \dots & u_{x_n, x_n} \end{pmatrix}$$

$$\Delta u = \sum_{i=1}^n u_{x_i, x_i} \quad \text{Laplace operator}$$

6 THE POISSON EQUATION

$$-\Delta u = f \quad \text{Poisson Equation}$$

$$-\Delta u = -u''(x) = f(x), \quad \forall x \in (0, 1) \quad u(0) = u(1) = 0 \quad 1D$$

6.0.1 DERIVATION OF THE POISSON EQUATION

1. Elastic body defined on domain Ω
2. Find configuration $u(x)$ s.t. the deformation energy is minimal.
3. Dirichlet BC: $u|_{\partial\Omega} \equiv 0$
4. Total elastic energy: $J(u) = \frac{1}{2} \int_{\Omega} |\nabla u|^2 dx - \int_{\Omega} u f dx$
f: load function .
5. Euler-Lagrange equation:
 $J'(u, v) = \lim_{\tau \rightarrow 0} \frac{J(u+\tau v) - J(u)}{\tau} = 0$
v is direction in which we calculate the derivative.

6. $|\nabla w|^2 = \langle \nabla w, \nabla w \rangle$ and taking the limit thus losing higher order terms yields:

$$\boxed{\int_{\Omega} \langle \nabla u, \nabla v \rangle dx - \int_{\Omega} f v dx = 0} \quad \text{Euler-Lagrange Equation}$$

7. Integration by parts yields: $-\int_{\Omega} v \Delta u dx - \int_{\Omega} v f dx + \int_{\partial\Omega} v \frac{\partial u}{\partial \nu} dx(x) = 0$
 $\frac{\partial u}{\partial \nu} = \nabla u \cdot \nu$: unit outward vector, normal to boundary, $v \equiv 0$ on $\partial\Omega$

$$\Rightarrow \int_{\Omega} (-\Delta u - f) v dx = 0 \quad \forall v$$

8. $\boxed{-\Delta u = f \quad u|_{\partial\Omega} \equiv 0}$ Poisson equation

6.1 POISSON EQUATION IN 1D

$$-u''(x) = f(x), \quad \forall x \in (0, 1)$$

$$u(0) = u(1) = 0$$

- $u'(y) = C_2 + \int_0^y u''(z) dz = C_2 - \int_0^y f(z) dz$
- $u(x) = C_1 + \int_0^x u'(y) dy = C_1 + C_2 x - \int_0^x \int_0^y f(z) dz dy$
- $F(y) = \int_0^y f(z) dz \Rightarrow F'(y) = f(y)$
- IBP: $\int_0^x F(y) dy = \int_0^x y' F(y) dy = x F(x) - \int_0^x y F'(y) dy = \int_0^x (x - y) f(y) dy$
- $u(x) = C_1 + C_2 x - \int_0^x (x - y) f(y) dy$
- $u(x) = \int_0^1 x(1 - y) f(y) dy - \int_0^x (x - y) f(y) dy$
 $C_1 = u(0) = 0 \quad C_2 = u(1) = \int_0^1 (1 - y) f(y) dy$

7. $\boxed{G(x, y) = \begin{cases} y(1 - x) & 0 \leq y \leq x \\ x(1 - y) & x \leq y \leq 1 \end{cases}}$ Greens function

8. $u(x) = \int_0^1 G(x, y) f(y) dy$

- The integral to find $u(x)$ is not always possible to evaluate exactly \rightarrow numerical quadrature rule.
- Slight perturbations in the poisson equation invalidate our solution. Thus a **general form of the PE** is used often:
 $-(a(x)u'(x))' + b(x)u'(x) + c(x)u(x) = f(x), \quad \forall x \in (0, 1)$
 $u(0) = u(1) = 0$
- Greens function representation are only available in 1D.

6.2 FINITE DIFFERENCE METHOD

6.2.1 DISCRETIZATION

Discretise domain $[0, 1]$ into $N + 2$ points ($N = \frac{1}{\Delta x} - 1$) by setting:

$$x_0 = 0, \quad x_{N+1} = 1, \quad x_j = j\Delta x, \quad j = 1, \dots, N$$

$$u_j \approx u(x_j)$$

$$f_j = f(x_j)$$

$$u'(x_j) \approx \frac{u_{j+1} - u_j}{\Delta x}$$

$$u''(x_j) \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2}$$

6.2.2 FINITE DIFFERENCE SCHEME

$$-u_{j+1} + 2u_j - u_{j-1} = \Delta x^2 f_j, \quad \forall j = 1, \dots, N$$

Yields a system of equations.

$$2u_1 - u_2 = \Delta x^2 f_1 \quad \text{BC: } u_0 = u(0) = 0, \quad j = 1$$

$$-u_{N-1} + 2u_N = \Delta x^2 f_N \quad \text{BC: } u_{N+1} = u(1) = 0, \quad j = N$$

$$U = [u_1, u_2, \dots, u_N]^T \quad F = \Delta x^2 [f_1, f_2, \dots, f_N]^T$$

$$AU = F$$

where $A = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix} \in \mathbb{R}^{N \times N}$

6.2.3 NUMERICAL RESULTS

For any Δx it holds that:

$$\|u^{\Delta x}\|_{\infty} \leq \frac{1}{8} \|f^{\Delta x}\|_{\infty}$$

where $u^{\Delta x} = [u_1, u_2, \dots, u_N]$, $f^{\Delta x} = [f_1, f_2, \dots, f_N]$

$$\|E^{\Delta x}\|_{\infty} \leq \frac{\Delta x^2}{96} \max_{0 \leq x \leq 1} |f''(x)|$$

where $E^{\Delta x} = [E_1, \dots, E_N]$

6.3 FDS FOR 2-D POISSON EQUATION

$$\boxed{\begin{aligned} -(u_{xx}(x) + u_{yy}(x)) &= f(x) & \text{for } x \in \Omega = (0, 1)^2 \\ u(x) &= 0 & \text{for } x \in \partial\Omega \end{aligned}}$$

6.3.1 DISCRETIZATION

$$\begin{array}{cc|cc} x_i = i\Delta x & \forall 1 \leq i \leq N & y_j = j\Delta y & \forall 1 \leq j \leq M \\ x_0 = 0 & x_{N+1} = 1 & y_0 = 0 & y_{M+1} = 1 \end{array}$$

$$u_{ij} \approx u(x_i, y_j) \quad f_{ij} = f(x_i, y_j)$$

$$\boxed{u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \quad u_{yy} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2}}$$

6.3.2 FINITE DIFFERENCE SCHEME

$$-\left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} \right) = f_{ij}$$

BC: $u_{0,j} = u_{N+1,j} = 0 \quad \forall j = 0, 1, \dots, M + 1$

BC: $u_{i,0} = u_{i,M+1} = 0, \quad \forall i = 0, 1, \dots, N + 1$

FOR $N=M \Leftrightarrow \Delta x = \Delta y$

$$U = [u_{1,1}, u_{2,1}, \dots, u_{N,1}, u_{1,2}, \dots, u_{N,2}, u_{1,N}, \dots, u_{N,N}]^T$$

$$F = \Delta x^2 [f_{1,1}, f_{2,1}, \dots, u_{N,1}, u_{1,2}, \dots, u_{N,2}, u_{1,N}, \dots, u_{N,N}]^T$$

$$AU = F$$

$$\text{where } A = \begin{bmatrix} B & -I & 0 & \cdots & 0 \\ -I & B & -I & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -I & B & -I \\ 0 & \cdots & 0 & -I & B \end{bmatrix} \in \mathbb{R}^{N^2 \times N^2} \text{ and}$$

$$B = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 4 & -1 \\ 0 & \cdots & 0 & -1 & 4 \end{bmatrix} \in \mathbb{R}^{N \times N}$$

7 FEM FOR 1D-POISSON

$$\begin{aligned} -u''(x) &= f(x) \quad \forall x \in (0, 1) \\ u(0) &= u(1) = 0 \end{aligned}$$

7.1 VARIATIONAL PRINCIPLE

STARTING POINT

The poisson equation represents the Euler-Lagrange equations corresponding to the solutions of the variational problem:

$$\min_u J(u) \quad J(u) = \frac{1}{2} \int_0^1 |u'(x)|^2 dx - \int_0^1 u(x) f(x) dx$$

REQUIEREMENTS FOR U

Dirichlet energy should be well defined thus:

$$\int_0^1 |u'(x)|^2 dx < \infty \quad \left| \int_0^1 u(x) f(x) dx \right| < \infty$$

$$H_0^1([0, 1]) := \left\{ u : [0, 1] \rightarrow \mathbb{R} : u(0) = u(1) = 0 \text{ and } \int_0^1 |u'(x)|^2 dx < \infty \right\}$$

$H_0^1([0, 1])$ is the set of all functions that vanish at the boundary and have the property that the integral of the square of their derivative is bounded. *Sobolev space*

$$\text{Do } H_0^1 \text{ FULFIL } \left| \int_0^1 u(x) f(x) dx \right| < \infty$$

$$\|u\|_{H_0^1([0, 1])} := \left(\int_0^1 |u'(x)|^2 dx \right)^{\frac{1}{2}}$$

Use the Cauchy-Schwarz inequality: $|v \cdot w| \leq \|v\| \|w\|$

$$\begin{aligned} \left| \int_0^1 u(x) f(x) dx \right| &\leq \int_0^1 |u(x)| |f(x)| dx \leq \\ &\left(\int_0^1 |u(x)|^2 dx \right)^{\frac{1}{2}} \cdot \left(\int_0^1 |f(x)|^2 dx \right)^{\frac{1}{2}} \end{aligned}$$

$$L^2([0, 1]) := \left\{ g : [0, 1] \rightarrow \mathbb{R} : \int_0^1 |g(x)|^2 dx < \infty \right\}$$

$$\|g\|_{L^2([0, 1])} = \left(\int_0^1 |g(x)|^2 dx \right)^{\frac{1}{2}}$$

$$u, f \in L^2([0, 1]) \Rightarrow \left| \int_0^1 u(x) f(x) dx \right| < \infty$$

CONSTRAINTS FOR WELL DEFINED DIRICHLET ENERGY

$$\begin{aligned} u &\in H_0^1([0, 1]), \quad u \in L^2([0, 1]) \\ f &\in L^2([0, 1]) \end{aligned}$$

But by the *Poincaré inequality*: $\|u\|_{L^2([0, 1])} \leq \|u\|_{H_0^1([0, 1])}$

Precise problem formulation:

Given $f \in L^2([0, 1])$, find $u \in H_0^1([0, 1])$, such that u minimises the energy functional $J(v)$ for all $v \in H_0^1([0, 1])$.

7.2 VARIATIONAL FORMULATION

$$\text{Find } u \in H_0^1([0, 1]) : J'(u, v) = 0 = \lim_{\tau \rightarrow 0} \frac{J(u+\tau v) - J(u)}{\tau}$$

$$\text{Thus } u \text{ must satisfy: } \int_0^1 u'(x) v'(x) dx = \int_0^1 v(x) f(x) dx \quad (A)$$

- (A) is the variational formulation of the 1D Poisson equation. Also known as principle of virtual work.

- (A) is well defined since

$$\begin{aligned} \left| \int_0^1 u'(x) v'(x) dx \right| &\leq \|u\|_{H_0^1([0, 1])} \|v\|_{H_0^1([0, 1])} < \infty \\ \text{and since } f &\in L^2([0, 1]), \text{ using Cauchy-Schwarz and Poincaré:} \\ \left| \int_0^1 v(x) f(x) dx \right| &\leq \|v\|_{L^2([0, 1])} \|f\|_{L^2([0, 1])} \\ &\leq \|v\|_{H_0^1([0, 1])} \|f\|_{L^2([0, 1])} < \infty \end{aligned}$$

- If we set $u = v$ and use Cauchy-Schwarz and Poincaré:

$$\begin{aligned} \|u\|_{H_0^1([0, 1])}^2 &= \int_0^1 |u'(x)|^2 dx = \int_0^1 u(x) f(x) dx \\ &\leq \|u\|_{L^2([0, 1])} \|f\|_{L^2([0, 1])} \leq \|u\|_{H_0^1([0, 1])} \|f\|_{L^2([0, 1])} \\ &\Rightarrow \|u\|_{H_0^1([0, 1])} \leq \|f\|_{L^2([0, 1])} \end{aligned}$$
 Thus we are provided with a stability estimate on the solution. The deflection of the membrane scales smaller than the applied load.

Derivation of the variational formulation:

- $-u''(x) = f(x)$
- Multiply with test function v : $-u''(x)v(x) = f(x)v(x)$
- Integrate: $-\int_0^1 u''(x)v(x) dx = \int_0^1 f(x)v(x) dx$
- Look at LHS:

$$\int_0^1 -u''(x)v(x) dx \stackrel{IBP}{=} [-u'(x)v(x)] - 0^1 - \int -u'(x)v'(x) dx = \int_0^1 u'(x)v'(x) dx.$$
- Thus: $\int_0^1 u'(x)v'(x) dx = \int_0^1 f(x)v(x) dx \quad \forall v \in V$

7.3 FEM FORMULATION

$$u \in V = H_0^1([0, 1]) : \int_0^1 u'(x) v'(x) dx = \int_0^1 v(x) f(x) dx$$

$$\text{Notation: } (g, h) = \int_0^1 g(x) h(x) dx$$

$$\int_0^1 u'(x) v'(x) dx = \int_0^1 v(x) f(x) dx \longrightarrow (u', v') = (f, v) \quad (A)$$

The function space V is infinite dimensional, FEM replaces V with a suitable finite-dimensional subspace $V^h \subseteq V$ and attempts to find $u \in V^h$ such that (A) holds $\forall v \in V^h$

$$V^h = \{w : [0, 1] \rightarrow \mathbb{R} : w \text{ is continuous, } w(0) = w(1) = 0 \text{ and } w|_{[(j-1)h, jh]} \text{ is linear } \forall j \in \{1, \dots, N+1\}\}$$

DISCRETIZATION

Domain $\Omega = [0, 1]$, $h > 0$, $N = \frac{1}{h} - 1$, $N + 2$ points:

$$x_0 = 0, \quad x_{N+1} = 1, \quad x_j = jh, \quad \forall j \in (1, N)$$

V^h is the set of all **continuous, piecewise linear** functions on $[0, 1]$ with respect to the partition $[0, 1] = \bigcup_{j=1}^{N+1} [(j-1)h, jh]$.

Basis of hat functions: $w(x) = \sum_{j=1}^N w_j \phi_j(x)$

$$\phi_j(x) = \begin{cases} \frac{x - x_{j-1}}{h} & x \in [x_{j-1}, x_j] \\ \frac{x_{j+1} - x}{h} & x \in [x_j, x_{j+1}] \\ 0 & \text{otherwise} \end{cases}$$

FEM 1D POISSON VARIATIONAL FORMULATION

Find $u_h \in V^h$ such that $(u'_h, v') = (f, v) \forall v \in V^h$

7.4 CONCRETE REALISATION OF FEM

$$v = \sum_{j=1}^N v_j \phi_j(x)$$

$$(u'_h, v') = (f, v) \Leftrightarrow \left(u'_h, \left(\sum_{j=1}^N v_j \phi_j(x) \right)' \right) = \left(f, \sum_{j=1}^N v_j \phi_j(x) \right)$$

By linearity: $\sum_{j=1}^N v_j (u'_h, \phi'_j) = \sum_{j=1}^N v_j (f, \phi_j)$

$(u'_h, \phi'_j) = (f, \phi_j)$ must hold for all $j = 1, \dots, N$

And $u_h \in V^h$: $u_h = \sum_{i=1}^N u_i \phi_i(x)$

$$\Rightarrow \sum_{i=1}^N u_i (\phi'_i, \phi'_j) = (f, \phi_j)$$

MATRIX FORMULATION

$$A = \{A_{ij}\}_{i,j=1,\dots,N} \quad A_{ij} = (\phi'_i, \phi'_j)$$

$$U = \{u_j\}_{j=1}^N \quad F = \{F_j\}_{j=1}^N \quad \sum_{i,j} A_{ij} u_j = F_j$$

$$AU = F$$

A is termed the stiffness matrix, F is termed the load vector and U is termed the solution vector.

1. A is symmetric: $A_{ij} = (\phi_i, \phi_j) = (\phi_j, \phi_i)$
2. A is positive definite:
3. **A is invertible and FEM well defined.**

7.5 CONVERGENCE ANALYSIS

MISSING

8 FEM FOR 2D-POISSON

$$-\Delta u = f \text{ in } \Omega \quad u|_{\partial\Omega} \equiv 0$$

DEFINITIONS

$$L^2(\Omega) = \{v : \Omega \rightarrow \mathbb{R} \int_{\Omega} |v|^2 dx < \infty\}$$

$$\|v\|_{L^2(\Omega)} = \left(\int_{\Omega} |v|^2 dx \right)^{1/2}$$

$$(u, v)_{L^2(\Omega)} = \int_{\Omega} u(x) v(x) dx \quad \text{Inner product}$$

$$H_0^1(\Omega) = \{v : \Omega \rightarrow \mathbb{R} : \int_{\Omega} |\nabla v|^2 dx < \infty \text{ and } v = 0 \text{ on } \partial\Omega\}$$

$$\|v\|_{H_0^1(\Omega)} = \left(\int_{\Omega} |\nabla v|^2 dx \right)^{\frac{1}{2}}$$

$$(u, v)_{H_0^1(\Omega)} = \int_{\Omega} \langle \nabla u, \nabla v \rangle dx$$

$\langle \cdot, \cdot \rangle$ - Dot product in 2D

$$\int_{\Omega} |v|^2 dx \leq C \int_{\Omega} |\nabla v|^2 dx \quad \text{Poincaré inequality}$$

8.1 VARIATIONAL FORMULATION

$$J(w) = \frac{1}{2} \int_{\Omega} |\nabla w|^2 dx - \int_{\Omega} w(x) f(x) dx \quad \text{Dirichlet Energy}$$

Find $u \in V = H_0^1(\Omega)$ such that for all $v \in V$ it holds that

$$(u, v)_{H_0^1(\Omega)} = (f, v)_{L^2(\Omega)}$$

or equivalently

$$\int_{\Omega} \langle \nabla u, \nabla v \rangle dx = \int_{\Omega} f(x) v(x) dx$$

8.2 FEM FORMULATION

V has infinite dimensions. In order to discretize it we have to consider a finite-dimensional subspace V^h .

TRIANGULATION

$$\bar{\Omega} = \bigcup_{K \in T_h} \bar{K} = \bar{K}_1 \cup \bar{K}_2 \cup \dots \cup \bar{K}_M$$

K_i are non-overlapping triangles. $k_i \cap k_j = \phi$ = common vertices = common edges.

T_h denotes the whole triangulation.

Mesh width: $h = \max_{K \in T_h} \text{diam}(K)$

$\text{diam}(K)$: length of the longest edge of the triangle K

\mathcal{N}_i : set of nodes (vertices)

$$V^h = \{v : \Omega \rightarrow \mathbb{R} : v \text{ cont.}, v|_K \text{ linear for each } K \in T_h, v = 0 \text{ on } \partial\Omega\}$$

V^h is the space of continuous, piecewise linear functions that vanish on the boundary $\partial\Omega$.

8.3 CONCRETE REALISATION OF FEM

$$\phi_j(x) \in V^h, \quad j = 1, \dots, N \quad \text{Hat functions}$$

$$\text{where } \phi_j(\mathcal{N}_i) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

$$v(x) = \sum_{j=1}^N v_j \phi_j(x) \quad \text{Basis for } V^h$$

where $v_j = v(\mathcal{N}_j) \forall j = 1, \dots, N$

Thus we can formulate the problem as:

$$\begin{aligned} (u_h, \phi_j)_{H_0^1(\Omega)} &= (f, \phi_j)_{L^2(\Omega)} \\ \int_{\Omega} \langle \nabla u_h, \nabla \phi_j \rangle dx &= \int_{\Omega} f(x) \phi_j(x) dx, \quad \forall 1 \leq j \leq N \\ u_h &= \sum_{i=1}^N u_i \phi_i \end{aligned}$$

$$\sum_{i=1}^N u_i \int_{\Omega} \langle \nabla \phi_i, \nabla \phi_j \rangle dx = \int_{\Omega} f(x) \phi_j(x) dx$$

$$\begin{aligned} U &= \{u_i\}_{i=1}^N & F &= \{F_j\}_{j=1}^N \\ A &= \{A_{ij}\}_{i,j=1}^N \\ F_j &= \int_{\Omega} f(x) \phi_j(x) dx & A_{ij} &= \int_{\Omega} \langle \nabla \phi_i, \nabla \phi_j \rangle dx \\ & & AU &= F \end{aligned}$$

$$A = \begin{bmatrix} B & -I & 0 & \cdots & 0 \\ -I & B & -I & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -I & B & -I \\ 0 & \cdots & 0 & -I & B \end{bmatrix}$$

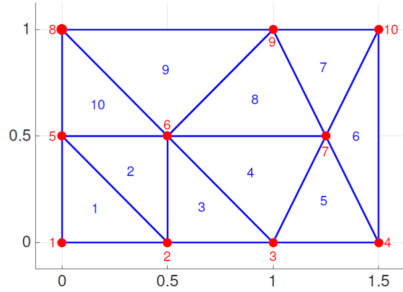
$$B = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 4 & -1 \\ 0 & \cdots & 0 & -1 & 4 \end{bmatrix}$$

- A is symmetric.
- A is positive definite.

9 IMPLEMENTATION OF THE FEM

9.1 STEP 1: TRIANGULATION

Exemplary triangulation T_h



$$Z = \begin{bmatrix} 0 & \frac{1}{2} & 1 & \frac{3}{2} & 0 & \frac{1}{2} & \frac{5}{4} & 0 & 1 & \frac{3}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 4 & 7 & 6 & 6 & 5 \\ 2 & 5 & 3 & 6 & 4 & 7 & 9 & 7 & 8 & 6 \\ 5 & 6 & 6 & 7 & 7 & 10 & 10 & 9 & 9 & 8 \end{bmatrix}$$

$$Z \in \mathbb{R}^{2 \times N}$$

- $Z(\cdot, j)$ refers to node \mathcal{N}_j
- $Z(1, j), Z(2, j)$ refer to x and y of \mathcal{N}_j

$$T \in \mathbb{R}^{3 \times M}$$

- $T(\cdot, j)$ refers to the j^{th} triangle K_j
- $T(i, j), (i = 1, 2, 3)$ represent the indices of the nodes of K_j

Often an additional vector denoting boundary nodes is added, enabling boundary conditions.

9.2 STEP 2: ELEMENT STIFFNESS MATRICES AND ELEMENT LOAD VECTORS

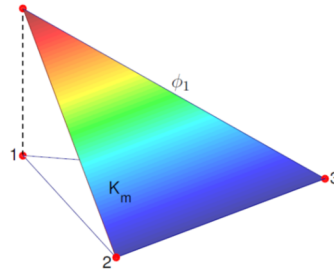
$$A_{ij} = \int_{\Omega} \langle \nabla \phi_i, \nabla \phi_j \rangle dx = \sum_{m=1}^M \int_{K_m} \langle \nabla \phi_i, \nabla \phi_j \rangle dx$$

$\int_{K_m} \langle \nabla \phi_i, \nabla \phi_j \rangle dx \neq 0$ only iff \mathcal{N}_i and \mathcal{N}_j are vertices of the same triangle!

$T(\alpha, m)$, $\alpha = 1, 2, 3$ - Labels of the vertices of K_m

$Z(i, T(\alpha, m))$, $i = 1, 2$, $\alpha = 1, 2, 3$ - Coordinates of each vertex $T(\alpha, m)$

$$\phi_{\alpha}(\mathcal{N}_{T(\beta, m)}) = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta \end{cases} \quad \text{Local shape functions}$$

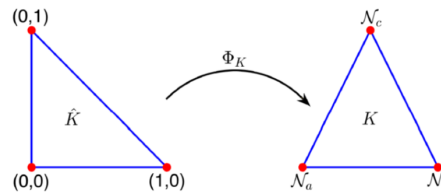


$$A_{\alpha, \beta}^m = \int_{K_m} \langle \nabla \phi_{\alpha}, \nabla \phi_{\beta} \rangle dx \quad \text{Stiffness matrix } A^m \text{ for } K_m$$

- $A^m \in \mathbb{R}^{3 \times 3}$ is symmetric.

$$F_{\alpha}^m = \int_{K_m} f(x) \phi_{\alpha}(x) dx \quad \text{Element load vector } F^m \text{ for } K_m$$

REFERENCE TRIANGLE



$\Phi_K : \hat{K} \rightarrow K$ Mapping

$$\begin{aligned} x = \Phi_K(\hat{x}) &= (\mathcal{N}_b - \mathcal{N}_a \quad \mathcal{N}_c - \mathcal{N}_a) \hat{x} + \mathcal{N}_a \\ &= J_K \hat{x} + \mathcal{N}_a \end{aligned}$$

$$F_{\alpha}^K = \int_K f(x) \phi_{\alpha}(x) dx = \int_{\hat{K}} f(\Phi_K(\hat{x})) \hat{\phi}_{\alpha}(\hat{x}) |\det(J_K)| d\hat{x}$$

$$\begin{aligned} A_{\alpha, \beta}^K &= \int_K \langle \nabla \phi_{\alpha}, \nabla \phi_{\beta} \rangle dx \\ &= \int_{\hat{K}} \langle J_K^{-T} \hat{\nabla} \hat{\phi}_{\alpha}, J_K^{-T} \hat{\nabla} \hat{\phi}_{\beta} \rangle |\det J_K| d\hat{x} \end{aligned}$$

- $\hat{\phi}_{\alpha}$ local shape functions of the reference triangle.
- $\phi_{\alpha}(x) = \hat{\phi}_{\alpha}(\hat{x})$
- $\nabla \phi_{\alpha} = J_K^{-T} \hat{\nabla} \hat{\phi}_{\alpha}(\hat{x})$

9.3 STEP 3: ASSEMBLY

1. $A = \text{zeros}(N, N)$;
2. $f = \text{zeros}(N)$;
3. looping over all triangles:

fetch A^m and F^m

$$A(T(\alpha, m), T(\beta, m)) = A(T(\alpha, m), T(\beta, m)) + A_{\alpha, \beta}^m$$

$$F(T(\alpha, m)) = F(\alpha, m) + F_{\alpha}^m$$

10 PARABOLIC PDE

$$Au_{xx} + 2Bu_{xt} + Cu_{tt} = F(x, t, u, u_x, u_t, \dots)$$

$AC - B^2 < 0 \rightarrow$ hyperbolic

$AC - B^2 = 0 \rightarrow$ parabolic

$AC - B^2 > 0 \rightarrow$ elliptic

$$\begin{aligned} u_t - \Delta u &= 0 && \text{on } \Omega \times (0, T) \\ u(x, 0) &= u_0(x) && \text{on } \Omega \\ u(0, t) &= u(1, t) = 0 && \text{on } (0, T) \end{aligned}$$

Heat Equation

10.1 SEPERATION OF VARIABLES

$$u = u(x, t) = \mathcal{T}(t)\mathcal{X}(x)$$

$$\frac{\mathcal{T}'(t)}{\mathcal{T}(t)} = \frac{\mathcal{X}''(x)}{\mathcal{X}(x)} = -\lambda_k$$

Solve these two ODE using BC yields:

$$\begin{aligned}\mathcal{X}_k(x) &= \sin(k\pi x), \quad \lambda_k = (k\pi)^2 \quad \forall k \in \mathbb{Z} \\ \mathcal{T}(t) &= e^{-(k\pi)^2 t}\end{aligned}$$

$$u(x, t) = \mathcal{X}(x)\mathcal{T}(t) = e^{-(k\pi)^2 t} \sin(k\pi x)$$

It remains to find a particular solution.

$$\int_0^1 \sin(k\pi x) \sin(m\pi x) dx = \begin{cases} 0 & \text{if } k \neq m \\ \frac{1}{2} & \text{if } k = m \end{cases}$$

Based on the theory of the fourier series:

$$u_0(x) = \sum_{k=1}^{\infty} u_k^0 \sin(k\pi x)$$

where $u_k^0 = 2 \int_0^1 u_0(x) \sin(k\pi x) dx$

$$u(x, t) = \sum_{k=1}^{\infty} u_k^0 e^{-(k\pi)^2 t} \sin(k\pi x)$$

This solution fulfils the initial PDE as well as the boundary conditions.

EVALUATION OF SEPERATION OF VARIABLES

1. Expand u_0 using the Fourier series:

$$u_0^N(x) = \sum_{k=1}^N u_k^0 \sin(k\pi x)$$

The error $|u_0 - u_0^N|$ is small for large N and if u_0 is smooth and satisfies $u_0(0) = u_0(1) = 0$

2. Calculate the coefficients $\{u_k^0\}_{k=1}^N$

$$u_k^0 = 2 \int_0^1 u_0(x) \sin(k\pi x) dx$$

3. $u_N(x, t) = \sum_{k=1}^N u_k^0 e^{-(k\pi)^2 t} \sin(k\pi x)$

Two error sources: Error due to the finite-truncation of the Fourier series, i.e. the series is only exact if infinite!

Error due to the use of quadrature rules to approximate integrals.

10.1.1 APPLICATION OF GAUSS-GREEN

$$\frac{d}{dt} \int_{\omega} u(x, t) dx = - \int_{\partial\omega} F \cdot v d\sigma(x)$$

Rate of change of u in ω = Flux over the boundary +
Sources - Sinks

$u(x, t) \rightarrow$ some quantity (velocity, temperature, concentration, pressure,...), $F \rightarrow$ flux, $v \rightarrow$ unit outward normal

$$\int_{\omega} u_t dx = - \int_{\partial\omega} F \cdot v d\sigma(x)$$

Gauss-Green (Integration by parts): $\int_{\partial\omega} F \cdot v d\sigma(x) =$

$$\int_{\omega} \nabla(F) dx$$

$$\int_{\omega} (u_t + \nabla F) dx = 0 \quad \forall \omega \in \Omega \Rightarrow u_t + \nabla(F) = 0$$

where $F = F(t, x, u, \nabla u, \dots)$

Special case: heat conduction: $F \propto -\nabla u \Rightarrow u_t = \Delta u$

10.2 ENERGY ESTIMATE

$$\mathcal{E}(t) := \frac{1}{2} \int_0^1 |u(x, t)|^2 dx \quad \text{Energy estimate}$$

$$\begin{aligned}\frac{d\mathcal{E}}{dt} &= \frac{1}{2} \int_0^1 (u^2)_t dx = \int_0^1 u u_t dx = \int_0^1 u u_{xx} dx = \\ &= - \int_0^1 u_x^2 dx + [u u_x]_0^1 \stackrel{\text{BC}}{=} - \int_0^1 u_x^2 dx\end{aligned}$$

$$\frac{d\mathcal{E}}{dt} = - \int_0^1 u_x^2 dx \leq 0 \Rightarrow \mathcal{E}(t) \leq \mathcal{E}(0)$$

10.3 UNIQUENESS OF A SOLUTION

$$w := u - \bar{u}$$

$$w_t - w_{xx} = 0 \quad \text{on } (0, 1) \times (0, T)$$

$$w(x, 0) = 0 \quad \text{on } (0, 1)$$

$$w(0, t) = w(1, t) = 0 \quad \text{on } (0, T)$$

$$\bar{\mathcal{E}}(t) = \frac{1}{2} \int_0^1 w^2(x, t) dx \Rightarrow \bar{\mathcal{E}}(t) \leq \bar{\mathcal{E}}(0)$$

$$\int_0^1 w^2(x, t) dx \leq \int_0^1 w^2(x, 0) dx \Rightarrow \int_0^1 x^2(x, t) dx \leq 0$$

$$w(x, t) \equiv 0 \Rightarrow u(x, t) = \bar{u}(x, t)$$

10.4 MAXIMUM PRINCIPLES

$$\max_{\substack{0 \leq x \leq 1, \\ 0 \leq t \leq T}} u(x, t) \leq \max(0, \max_x(u_0(x)))$$

10.5 EXPLICIT FINITE DIFFERENCE SCHEMES FOR THE HEAT EQUATION

1. Discretising the Domain:

$$\Delta x > 0 \quad \delta t > 0 \quad N = \frac{1}{\Delta x} - 1$$

$$x_0 = 0, \quad x_j = j\Delta x, \quad j = 1, \dots, N, \quad x_{N+1} = 1$$

$\rightarrow N + 2$ equally spaced points

$$t_0 = 0, \quad t^n = n\Delta t, \quad n = 1, \dots, M, \quad t^{M+1} = T$$

2. Discretising the Solution u

$$U_j^n \approx u(x_j, t^n)$$

3. Discretising the Derivatives

$$u_{xx}(x_j, t^n) \approx \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2}$$

$$u_t(x_j, t^n) \approx \frac{U_j^{n+1} - U_j^n}{\Delta t}$$

4. The Finite Difference Scheme

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

$$\lambda = \frac{\Delta t}{\Delta x^2}$$

$$U_j^{n+1} = (1 - 2\lambda)U_j^n + \lambda U_{j+1}^n + \lambda U_{j-1}^n$$

Initial and boundary conditions:

$$U_j^0 = u(x_j, 0) = u_0(x_j), \quad \forall 1 \leq j \leq N$$

$$U_0^n = U_{N+1}^n \equiv 0 \quad \forall 0 \leq n \leq M + 1$$

This method is called „explicit“ since we use the explicit euler method for time stepping.

$$\Delta t \leq \frac{1}{2} \Delta x^2 \quad \text{Stability condition}$$

10.5.1 SIMPLIFIED NOTATION

$$\text{Forward difference in space} \quad D_x^+ w_j^n = \frac{w_{j+1}^n - w_j^n}{\Delta x}$$

$$\text{Backward difference in space} \quad D_x^- w_j^n = \frac{w_j^n - w_{j-1}^n}{\Delta x}$$

$$\text{Forward difference in time} \quad D_t^+ w_j^n = \frac{w_j^{n+1} - w_j^n}{\Delta t}$$

$$\text{Backward difference in time} \quad D_t^- w_j^n = \frac{w_j^n - w_j^{n-1}}{\Delta t}$$

The finite difference scheme can then be recast as:

$$D_t^+ U_j^n - D_x^- D_x^+ U_j^n = 0$$

10.5.2 TRUNCATION ERROR

$$\tau_j^n = \frac{u_j^{n+1} - u_j^n}{\Delta t} - \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2}$$

or equivalently $\tau_j^n = D_t^+ u_j^n - D_x^- D_x^+ u_j^n$

$$|\tau_j^n| \leq C(\Delta t + \Delta x^2) \text{ or } \sqrt{\frac{\Delta x}{2} \sum_{j=1}^N |u_j^n - U_j^n|^2} \leq \bar{C}(\Delta t + \Delta x^2)$$

Thus the explicit finite difference scheme has a first-order rate of convergence in time and a second-order rate of convergence in space.

10.6 AN IMPLICIT FINITE DIFFERENCE SCHEME

$$D_t^- U_j^{n+1} = D_x^- D_x^+ U_j^{n+1}$$

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

$$\lambda = \frac{\Delta t}{\Delta x^2}$$

$$-\lambda U_{j-1}^{n+1} + (1 + 2\lambda)U_j^{n+1} - \lambda U_{j+1}^{n+1} = U_j^n$$

$$AU^{n+1} = F^n$$

$$U^{n+1} = \{U_j^{n+1}\}_{j=1}^N \quad F^n = \{U_j^n\}_{j=1}^N$$

$$A = \begin{bmatrix} 1+2\lambda & -\lambda & 0 & \cdot & 0 \\ -\lambda & 1+2\lambda & -\lambda & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & \\ \cdot & -\lambda & 1+2\lambda & -\lambda & \\ 0 & \cdot & 0 & -\lambda & 1+2\lambda \end{bmatrix}$$

Unconditional stability!

10.7 CRANK-NICLOSON SCHEME

$$D_t^+ U_j^n = \frac{1}{2} D_x^- D_x^+ U_j^n + \frac{1}{2} D_x^2 D_x^+ U_j^{n+1}$$

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

Crank-Nicholson is the formal average of the explicit finite difference scheme and the implicit finite difference scheme.

$$U_0^n = U_{N+1}^n = 0 \quad U_j^0 = u_j^0 = u_0(x_j) \quad \lambda = \frac{\Delta t}{\Delta x^2}$$

$$-\frac{\lambda}{2} U_{j-1}^{n+1} + (1 + \lambda)U_j^{n+1} - \frac{\lambda}{2} U_{j+1}^{n+1} = \frac{\lambda}{2} U_{j-1}^n + (1 - \lambda)U_j^n - \frac{\lambda}{2} U_{j+1}^n = F_j^n$$

$$AU^{n+1} = F^n$$

$$U^{n+1} = \{U_j^{n+1}\}_{j=1}^N \quad F^n = \{F_j^n\}_{j=1}^N$$

$$A = \begin{bmatrix} 1+\lambda & -\frac{\lambda}{2} & 0 & \cdot & 0 \\ -\frac{\lambda}{2} & 1+\lambda & -\frac{\lambda}{2} & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & -\frac{\lambda}{2} & 1+\lambda & -\frac{\lambda}{2} \\ 0 & \cdot & 0 & -\frac{\lambda}{2} & 1+\lambda \end{bmatrix}$$

Unconditionally stable!

Despite the unconditional stability the CN-scheme **only satisfies the discrete maximum principle** if $\lambda = \frac{\Delta t}{\Delta x^2} \leq 1$. For large values of λ the solution may contain spurious oscillations.

10.7.1 TRUNCATION ERROR

$$\tau_j^n = \frac{u_j^{n+1} - u_j^n}{\Delta t} - \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{2\Delta x^2} - \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{2\Delta x^2}$$

$$\tau_j^n = D_t^+ u_j^n - \left(\frac{1}{2} D_x^- D_x^+ u_j^n + \frac{1}{2} D_x^- D_x^+ u_j^{n+1} \right)$$

$$|\tau_j^n| \leq C(\Delta t^2 + \Delta x^2)$$

CN has a second-order rate of convergence in both time and space.

10.8 CONVERGENCE STUDIES

11 LINEAR TRANSPORT EQUATIONS (HYPERBOLIC PDES)

$$u_t + a(x, t)u_x = 0 \quad \forall (x, t) \in \mathbb{R} \times \mathbb{R}_+$$

11.1 METHOD OF CHARACTERISTICS

Assume that we have a curve along which the solution u is constant.

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

$$0 = \frac{d}{dt} u(x(t), t) = u_t(x(t), t) + u_x(x(t), t)x'(t)$$

where $x(t)$ is a characteristic curve.

Since this has to fulfil the linear transport equation we know that:

$$\text{characteristics} \quad \begin{cases} x'(t) = a(x(t), t) \\ x(0) = x_0 \end{cases}$$

11.1.1 GRONWALL'S INEQUALITY

$$u'(t) \leq \beta(t)u(t) \quad \forall t \in (a, b)$$

where $\beta(t)$ continuous, $u(t)$ differentiable on some $[a, b]$.

$$u(t) \leq u(a) \exp \left(\int_a^t \beta(t) dt \right) \quad \forall t \in [a, b]$$

Let $u(x, t)$ be a smooth solution, $\lim_{|x| \rightarrow \infty} u(x, t) = 0$. Then u fulfils the following energy bound:

$$\int_{\mathbb{R}} u^2(x, t) dx \leq e^{\|a\|_{C^1} t} \int_{\mathbb{R}} u_0^2(x) dx$$

11.2 FINITE DIFFERENCE SCHEMES

11.2.1 DISCRETIZATION

$[x_l, x_r]$ is discretized with a mesh size Δx into a sequence of $N + 1$ points.

Divide $[0, T]$ into M points $t^n = n\Delta t$

Then a finite difference scheme is applied:

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

By an energy analysis we conclude that this central difference scheme does not produce valid but oscillatory solutions with increasing energies.

11.2.2 UPWIND SCHEMES

The central scheme does not respect the direction of propagation of information for the transport equation. This is corrected by the upwind scheme.

As found by the method of characteristics the characteristic curves are defined by:

$$x'(t) = a$$

Thus if $a > 0$ the direction of information propagation is from left to right, then we use a backward difference in space to obtain our scheme. We do the opposite if $a < 0$.

$$a^+ = \max\{a, 0\}, \quad a^- = \min\{a, 0\}, \quad |a| = a^+ - a^-$$

Thus the combined (backwards/forwards) upwind scheme can be written as:

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

or

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

However the upwind scheme still is only stable for some $\frac{\Delta t}{\Delta x}$.

11.2.3 STABILITY FOR THE UPWIND SCHEME

If $|a|\frac{\Delta t}{\Delta x} \leq 1$ the upwind scheme satisfies the energy estimate $E^{n+1} \leq E^n$ and is thus **conditionally stable**.