

Sheet 9

Discussion of the sheet: Tue., 23.05.2023

1. (Prager-Synge theorem) Let $\Omega \subset \mathbb{R}^d$. Consider the solution $u \in H_0^1(\Omega)$ of the Poisson equation

$$\begin{aligned} -\Delta u &= f & \text{in } \Omega \\ u &= 0 & \text{on } \partial\Omega. \end{aligned}$$

Let σ be a vector field satisfying $\operatorname{div} \sigma = -f$. Show that for any $v \in H_0^1(\Omega)$ there holds

$$(\nabla(u-v), \nabla u - \sigma)_{L^2(\Omega)} = 0$$

and consequently

$$\|\nabla(u-v)\|_{L^2(\Omega)}^2 + \|\nabla u - \sigma\|_{L^2(\Omega)}^2 = \|\nabla v - \sigma\|_{L^2(\Omega)}^2.$$

Note that the idea stems from a rewriting of the Poisson problem as $\nabla u = \sigma$, $\operatorname{div} \sigma = -f$, where the new variable σ can be seen as the flux of u . Prager-Synge theorem can then be used to define an error estimator by choosing $v = u_h$ and constructing σ as a FEM approximation of the flux (i.e. post-process ∇u_h to σ_h so that $\nabla \cdot \sigma_h = f$).

2. Go to gitlab.tuwien.ac.at/asc/praetorius/mooafem and download the MATLAB package MooAFEM (an object orientated adaptive FEM code). Run 'setup' in your Matlab to install it.

Go to the examples folder and run the codes 'adaptiveFEM.m' and 'convergenceRates.m' and explain: what is the PDE? on which domain? what kind of FEM space is used? how are the adaptive steps (error estimator, marking, refinement) implemented? where does most of the refinement take place? what do we see from the convergence rates?

3. Consider the advection equation. Determine the numerical domain of dependence for the forward time/backward space method. Argue that the CFL-condition is $\frac{k}{h} \leq 1$.
4. Write a program to solve the heat equation

$$\begin{aligned} \partial_t u - \partial_{xx} u &= f & \text{in } (0,1) \times (0,T) \\ u(0,t) &= u(1,t) = 0 & \forall t \in (0,T) \\ u(x,0) &= u_0(x) \end{aligned}$$

by using a semi-discretization by classical FEM in space and discretize the resulting ODE with the implicit Euler method. Take piecewise linear Lagrangian finite elements of order 1 on an uniform mesh of mesh.width h in space and take uniform time-steps of length k .

Test your program with the exact solution $u(x,t) = e^{-t}x(1-x)$ at $T = 1$ using $k = ch$ with different constants $c > 0$. For the error you can take the error measure $\sqrt{h} \sqrt{\sum_j |u(x_j, T) - u_j^T|^2}$, where you should see convergence of order h .

5. Modify your program from the previous example by using the explicit Euler method instead of the implicit Euler method. Take two choices of time steps k :

$$k = \frac{2.01}{\lambda_{\max}}, \quad k = \frac{1.99}{\lambda_{\max}},$$

where λ_{\max} is the largest eigenvalue of $M^{-1}A$ (can e.g. be computed in Matlab using `eigs`).

Plot your computed solutions, the error versus h , and λ_{\max} versus h .

1. (Prager-Synge theorem) Let $\Omega \subset \mathbb{R}^d$. Consider the solution $u \in H_0^1(\Omega)$ of the Poisson equation

$$\begin{aligned} -\Delta u &= f & \text{in } \Omega \\ u &= 0 & \text{on } \partial\Omega. \end{aligned}$$

Let σ be a vector field satisfying $\operatorname{div} \sigma = -f$. Show that for any $v \in H_0^1(\Omega)$ there holds

$$(\nabla(u-v), \nabla u - \sigma)_{L^2(\Omega)} = 0$$

and consequently

$$\|\nabla(u-v)\|_{L^2(\Omega)}^2 + \|\nabla u - \sigma\|_{L^2(\Omega)}^2 = \|\nabla v - \sigma\|_{L^2(\Omega)}^2.$$

Note that the idea stems from a rewriting of the Poisson problem as $\nabla u = \sigma$, $\operatorname{div} \sigma = -f$, where the new variable σ can be seen as the flux of u . Prager-Synge theorem can then be used to define an error estimator by choosing $v = u_h$ and constructing σ as a FEM approximation of the flux (i.e. post-process ∇u_h to σ_h so that $\nabla \cdot \sigma_h = f$).

$$(\nabla(u-v), \nabla u - \sigma)_{L^2(\Omega)} = \int_{\Omega} \nabla(u-v) \cdot (\nabla u - \sigma) dx \quad [f'g = fg - fg']$$

$$= \underbrace{(u-v)(\nabla u - \sigma)}_{=0} \Big|_{\partial\Omega} - \int_{\Omega} (u-v)(\Delta u - \nabla \cdot \sigma) dx \quad \begin{array}{l} \Delta u = -f \\ \nabla \cdot \sigma = f \end{array}$$

$$= - \int_{\Omega} (u-v)(-f + f) dx = 0 \quad \left| \begin{array}{l} \text{and same goes} \\ \text{for} \\ (\nabla(u-v), \nabla u - \sigma)_{L^2(\Omega)} \end{array} \right.$$

$$\|\nabla(u-v)\|_{L^2(\Omega)}^2 + \|\nabla u - \sigma\|_{L^2(\Omega)}^2 =$$

$$\left| \begin{array}{l} \text{cancel in } 2 \int \nabla(u-v) \cdot (\nabla u - \sigma) dx = 0 \\ \|\nabla(u-v)\|_{L^2(\Omega)}^2 = | -1 | \|\nabla(u-v)\|_{L^2(\Omega)}^2 \end{array} \right.$$

$$= \int_{\Omega} (\nabla(u-v))^2 + 2 \underbrace{\nabla(u-v) \cdot (\nabla u - \sigma)}_{=0} + (\nabla u - \sigma)^2 dx \quad [(a+b)^2 = a^2 + 2ab + b^2]$$

$$= \int_{\Omega} (\nabla(u-v) + (\nabla u - \sigma))^2 dx$$

$$= \|\nabla(u-v) + \nabla u - \sigma\|_{L^2(\Omega)}^2 = \|\cancel{\nabla u - \nabla u} + \cancel{\nabla u} - \sigma\|_{L^2(\Omega)}^2$$

$$= \|\nabla u - \sigma\|_{L^2(\Omega)}^2$$

2. Go to gitlab.tuwien.ac.at/asc/praetorius/mooafem and download the MATLAB package MooAFEM (an object orientated adaptive FEM code). Run 'setup' in your Matlab to install it.

Go to the examples folder and run the codes 'adaptiveFEM.m' and 'convergenceRates.m' and explain: what is the PDE? on which domain? what kind of FEM space is used? how are the adaptive steps (error estimator, marking, refinement) implemented? where does most of the refinement take place? what do we see from the convergence rates?

① We use 1. order Lagrangian elements

$$\square: \Omega = [-1, 1] \times [0, 1] + [-1, 0] \times [-1, 0]$$

Bilinear Form in

$$A_{ij} = \int_{\Omega} \nabla \varphi_i \cdot a \nabla \varphi_j + b \nabla \varphi_j \cdot \varphi_i + c \varphi_j \varphi_i \, dx \\ + \int_{\Gamma_R} r \varphi_j \varphi_i \, ds$$

robin

$$a = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad c = x^2, \quad r = 0$$

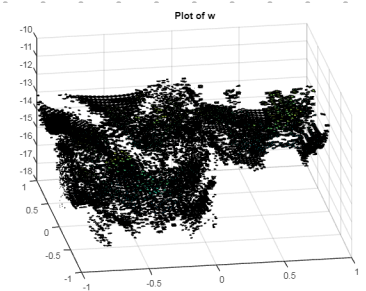
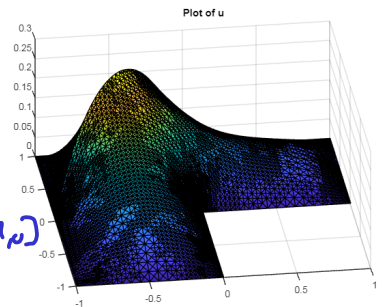
Linear Form in

$$f_i = \int_{\Omega} f \varphi_i + f^{vec} \nabla \varphi_i \, dx + \int_{\Gamma_u} n \varphi_i \, ds + \int_{\Gamma_R} r \varphi_i \, ds$$

neumann robin

$$f = 1, \quad n = 0, \quad r = 0, \quad f^{vec} = (\text{composite function } Q(w) [u; -u])$$

$$-\text{div}(A \nabla u) + b \nabla u + c \cdot u = f - f^{vec}, \quad \Omega \\ u = 0, \quad \Gamma$$



$$h_T^2 = h_T^2 \cdot \| -\nabla a \cdot \nabla u + b \cdot \nabla u + c \cdot u - (f - \text{div}(f^{vec})) \|_{L^2(\Omega)}^2 + h_T \| a \cdot \nabla u - f^{vec} \|_{L^2(\partial \Omega)}^2$$

We use Dörfler marking and Newest Volume Bisection

② We use high order Lagrangian elements

$$\square: \Omega = [-1, 1] \times [0, 1] + [-1, 0] \times [-1, 0]$$

Bilinear Form in

$$A_{ij} = \int_{\Omega} \nabla \varphi_i \cdot a \nabla \varphi_j + b \nabla \varphi_j \cdot \varphi_i + c \varphi_j \varphi_i \, dx \\ + \int_{\Gamma_R} r \varphi_j \varphi_i \, ds$$

robin

$$a = (1)$$

Linear Form in

$$f_i = \int_{\Omega} f \varphi_i + f^{\text{vec}} \nabla \varphi_i \, dx + \int_{\Gamma_u} h \varphi_i \, ds + \int_{\Gamma_R} r \varphi_i \, ds$$

neumann robin

$$f = 0, f^{\text{vec}} = 0, r = 0, h = \text{Mesh Functions (mesh, exact solution values)}$$

$$\rightarrow -\Delta u = 1$$

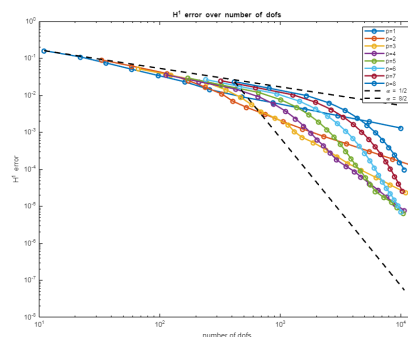
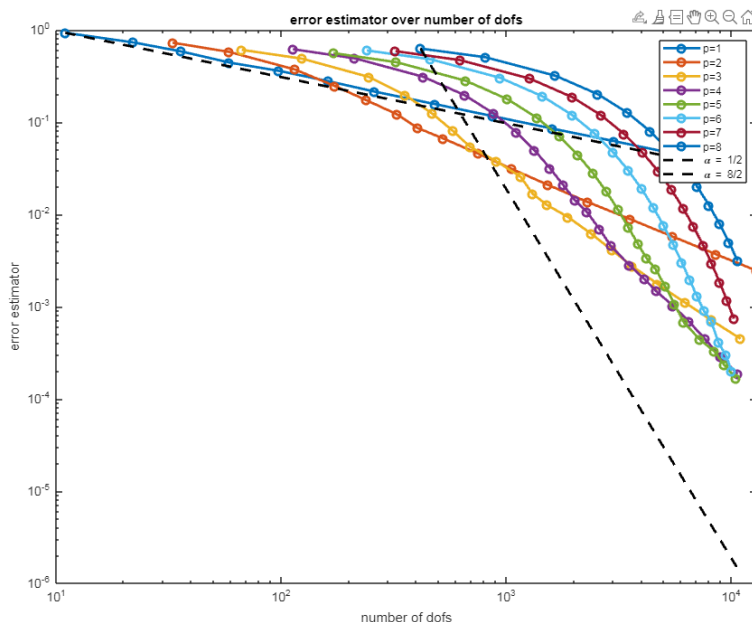
$$\eta^2 = \left(\sum \eta_T^2 \right)^{\frac{1}{2}}$$

again Dörfler marking
and NVB

From the convergence

rates we see a

$L^{-\frac{p}{2}}$ convergence for
elements of order p



3. Consider the advection equation. Determine the numerical domain of dependence for the forward time/backward space method. Argue that the CFL-condition is $\frac{k}{h} \leq 1$.

Forward time/backward space:

$$j = x$$

$$n = t$$

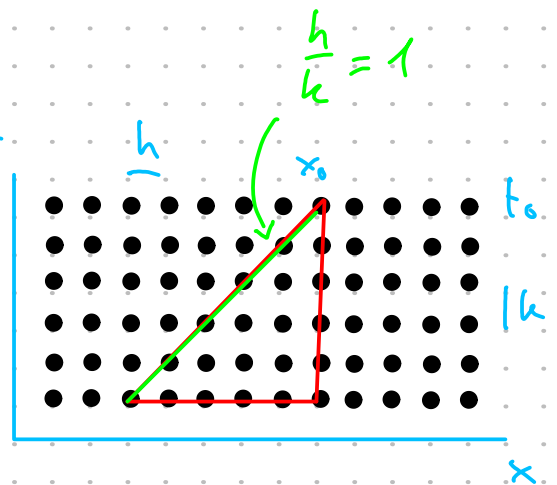
$$\frac{u_j^{n+1} - u_j^n}{k} + \frac{u_j^n - u_{j-1}^n}{h} = f(x_j, t_n)$$

We take $f=0$

$$\frac{u_j^{n+1} - u_j^n}{k} = - \frac{u_j^n - u_{j-1}^n}{h}$$

$$\rightarrow u_j^{n+1} = -\frac{k}{h} (u_j^n - u_{j-1}^n) + u_j^n$$

$$= \left(1 - \frac{k}{h}\right) u_j^n + \frac{k}{h} u_{j-1}^n$$



Dependency of the approximation u_h at x_0, t_0

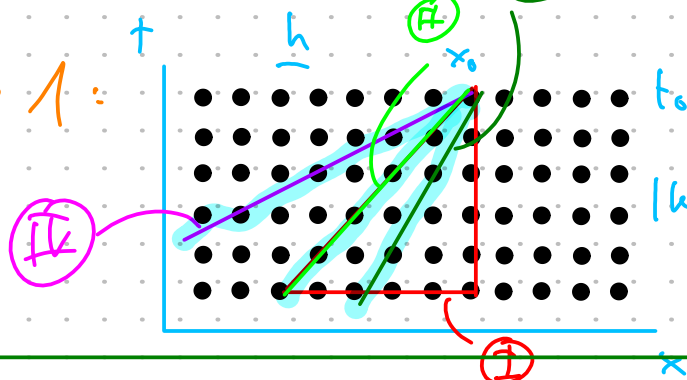
on $u_h(x_0, t_0 - k)$ and $u_h(x_0 - h, t_0 - k)$ - gives us numerical domain of dependence.

For the domain of dependence, we need

look at the Courant number:

If $\frac{h}{k}$ was > 1 :

case (IV)



$$\textcircled{II} \triangle \Rightarrow 1 = \frac{h}{k}$$

$$\textcircled{III} \triangle \Rightarrow \frac{1}{2} = \frac{h}{k}$$

$$\textcircled{IV} \triangle \Rightarrow \frac{1}{2} = \frac{h}{k}$$

We see that (IV) \neq (I) (hence we'd get unstable solutions)

whereas (II), (III) \subseteq (I) $\rightarrow \frac{h}{k} \leq 1$

