Universitat Politècnica de Catalunya
Numerical Methods in Engineering
Coupled Problems

Coupled Problems Deliverable

Eduard Gómez Escandell
June 16, 2020

Contents

1 Transmission Conditions

1.1 Euler-Bernoulli beam theory

The deflection $v(x)$ of an Euler-Bemouilli beam is governed by the differential equation

$$EI\frac{\mathrm{d}^4 v}{\mathrm{d}x^4} = f$$

where EI is a mechanical property of the beam section and the beam material and f is the distributed load. Assuming for example that the beam is clamped at $x = 0$ and $x = L$, the Principle of Virtual Work (PTV) states that the solution $v(x)$ satisfies

$$EI\int_0^L \frac{\mathrm{d}^2\delta v}{\mathrm{d}x^2}\frac{\mathrm{d}^2 v}{\mathrm{d}x^2} = \int_0^L \delta v f \tag{1}$$

for all δv such that $\delta v(0) = \delta v(L) = 0$, $\frac{\mathrm{d}\delta v}{\mathrm{d}x}(0) = \frac{\mathrm{d}\delta v}{\mathrm{d}x}(L) = 0$.

Part A

Postulate the space of functions where both v and δv must belong. Justify the answer.

We need the following integral to be well-defined:

$$\int_0^L \frac{\mathrm{d}^2\delta v}{\mathrm{d}x^2}\frac{\mathrm{d}^2 v}{\mathrm{d}x^2}\, dx \tag{2}$$

This can be rewritten as:

$$\int_0^L \frac{d}{dx}\left(\frac{d\delta v}{dx}\right)\ \frac{d}{dx}\left(\frac{dv}{dx}\right)\ dx \tag{3}$$

The space that fulfills this requirement is:

$$\frac{d\delta v}{dx},\ \frac{dv}{dx} \in L^2(\Omega) \tag{4}$$

The space that contains all functions whose first derivative is $L^2(\Omega)$ is $\mathcal{H}^1(\Omega)$. Hence:

$$\delta v,\ v \in \mathcal{H}^1(\Omega) \tag{5}$$

Part B

If $[0, L] = [0, P] \cup (P, L]$, obtain the transmission conditions at P implied by regularity requirements.

The first regularity condition dictates:

$$[\![u]\!] = 0 \tag{6}$$

This means:

$$\lim_{\epsilon\to 0}\Big(u(P+\epsilon) - u(P-\epsilon)\Big) = 0 \tag{7}$$

We also have to fulfill the second regularity conditions of the first derivative:

$$[\![\frac{du}{dx}]\!] = 0 \tag{8}$$

Which implies

$$\lim_{\epsilon\to 0}\Big(\frac{du}{dx}(P+\epsilon) - \frac{du}{dx}(P-\epsilon)\Big) = 0 \tag{9}$$

Both of the condition need be fulfilled for u, $\frac{du}{dx}$ to be square integrable, and therefore be in the spaces expalained in the previous exercise.

Part C

Obtain the transmission conditions at P that follow by imposing in the PTV that the integral is additive.

We'll start with the strong form of the problem:

$$EI\frac{\mathrm{d}^4 v}{\mathrm{d}x^4} = f \tag{10}$$

Therefore:

$$\int_0^L EI\frac{\mathrm{d}^4 v}{\mathrm{d}x^4}\delta v\,\mathrm{d}x = \int_\Omega f\delta v\,\mathrm{d}x \tag{11}$$

Integration by parts twice yields:

$$EI\left[\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right]_0^L + EI\int_0^L \frac{\mathrm{d}^2\delta v}{\mathrm{d}dx^2}\frac{\mathrm{d}^2 v}{\mathrm{d}dx^2}\,\mathrm{d}x = \int_0^L f\delta v\,\mathrm{d}x \tag{12}$$

Let's now split the domain at point P. We obtain the following set of equations:

$$\left.\begin{array}{ll} EI\left[\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right]_0^{P_L} + EI\int_0^{P_L}\frac{\mathrm{d}^2\delta v}{\mathrm{d}dx^2}\frac{\mathrm{d}^2 v}{\mathrm{d}dx^2}\,\mathrm{d}x = \int_0^{P_L} f\delta v\,\mathrm{d}x & \text{in } \Omega_1 : [0, P_L) \\ EI\left[\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right]_{P_R}^L + EI\int_{P_R}^L\frac{\mathrm{d}^2\delta v}{\mathrm{d}dx^2}\frac{\mathrm{d}^2 v}{\mathrm{d}dx^2}\,\mathrm{d}x = \int_{P_R}^L f\delta v\,\mathrm{d}x & \text{in } \Omega_2 : [P_R, L] \end{array}\right\} \tag{13}$$

Notice that we distinguish P_L and P_R from P. They simply refer to approaching form the left or from the right. We'll now impose the additivity.

$$EI\left[\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right]_0^{P_L} + EI\int_0^{P_L}\frac{\mathrm{d}^2\delta v}{\mathrm{d}dx^2}\frac{\mathrm{d}^2 v}{\mathrm{d}dx^2}\,\mathrm{d}x + EI\left[\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right]_{P_R}^L + EI\int_{P_R}^L\frac{\mathrm{d}^2\delta v}{\mathrm{d}dx^2}\frac{\mathrm{d}^2 v}{\mathrm{d}dx^2}\,\mathrm{d}x$$
$$= \int_0^{P_L} f\delta v\,\mathrm{d}x + \int_{P_R}^L f\delta v\,\mathrm{d}x$$

This result must be equal to that in 12. Let's start by combining the integrals:

$$EI\left[\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right]_0^{P_L} + EI\left[\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3} - \frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right]_{P_R}^L + EI\int_0^L\frac{\mathrm{d}^2\delta v}{\mathrm{d}dx^2}\frac{\mathrm{d}^2 v}{\mathrm{d}dx^2}\,\mathrm{d}x = \int_0^L f\delta v\,\mathrm{d}x \tag{14}$$

Let's also expand the brackets and rearrange them:

$$EI\;\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3}\bigg|_0^{P_L} + EI\;\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3}\bigg|_{P_R}^L - EI\;\frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\bigg|_0^{P_L} - EI\;\frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\bigg|_{P_R}^L + EI\int_0^L\frac{\mathrm{d}^2\delta v}{\mathrm{d}dx^2}\frac{\mathrm{d}^2 v}{\mathrm{d}dx^2}\,\mathrm{d}x = \int_0^L f\delta v\,\mathrm{d}x \tag{15}$$

For additivity to be fulfilled the first two terms must follow:

$$EI\;\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3}\bigg|_0^{P_L} - EI\;\delta v\frac{\mathrm{d}^3 v}{\mathrm{d}x^3}\bigg|_{P_R}^L = EI\;\frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\bigg|_0^L \tag{16}$$

For this to be true the following relationship must hold:

$$\frac{\mathrm{d}^3 v}{\mathrm{d}x^3}\bigg|_{P_R} = \frac{\mathrm{d}^3 v}{\mathrm{d}x^3}\bigg|_{P_L} \tag{17}$$

This can be stated as the continuity of shear force or $[\![\frac{d^3u}{dx^3}]\!] = 0$ Referring back to equation 16 and taking the second pair of terms we reach a similar conclusion:

$$EI \left.\frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right|_0^{P_L} - EI \left.\frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right|_{P_R}^{L} = EI \left.\frac{\mathrm{d}\delta v}{\mathrm{d}dx}\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right|_0^{L} \tag{18}$$

For this to be true the following relationship must hold:

$$EI \left.\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right|_{P_L} = EI \left.\frac{\mathrm{d}^2 v}{\mathrm{d}dx}\right|_{P_R} \tag{19}$$

This can be stated as the continuity of bending moment or $[\![\frac{d^3u}{dx^2}]\!] = 0$.

1.2 The Maxwell problem

The Maxwell problem consists in finding a vector field $\boldsymbol{u} : \Omega \longrightarrow \mathbb{R}^3$ such that

$$\begin{aligned} \nu \nabla \times \nabla \times \boldsymbol{u} &= \boldsymbol{f} \quad \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in } \Omega \\ \boldsymbol{n} \times \boldsymbol{u} &= 0 \quad \text{on } \partial\Omega \end{aligned}$$

where $\nu > 0$, f is a divergence free force field and n the unit external normal. Equation $\nabla \cdot u = 0$ is in fact redundant.

Part A

Write a variational statement of the problem. Postulate the space of functions where u must belong. Justify the answer.

We'll start by multiplying the first equation by a test function $\boldsymbol{w} : \mathbb{R}^3 \mapsto \mathbb{R}^3$:

$$\int_\Omega \boldsymbol{w} \cdot \nu \nabla \times \nabla \times \boldsymbol{u} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w} \tag{20}$$

Let's explore the first term with the help of index notation:

$$\begin{aligned} [\boldsymbol{w} \cdot \nabla \times \nabla \times \boldsymbol{u}] &= w_i \varepsilon_{ijk} \partial_j (\nabla \times \boldsymbol{u})_k \\ &= \varepsilon_{ijk} \partial_j [w_i (\nabla \times \boldsymbol{u})_k] - \varepsilon_{ijk} (\partial_j w_i)(\nabla \times \boldsymbol{u})_k \\ &= -\partial_j \varepsilon_{jik} w_i (\nabla \times \boldsymbol{u})_k + (\nabla \times \boldsymbol{u})_k \varepsilon_{kij} \partial_j w_i \\ &= -\nabla \cdot (\boldsymbol{w} \times \nabla \times \boldsymbol{u}) + (\nabla \times \boldsymbol{u}) \cdot (\nabla \times \boldsymbol{u}) \end{aligned}$$

Plugging this back into equation 20 results in:

$$\int_\Omega (\nabla \times \boldsymbol{w}) \cdot (\nu \nabla \times \boldsymbol{u}) - \int_\Omega \nabla \cdot (\boldsymbol{w} \times \nu \nabla \times \boldsymbol{u}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w} \tag{21}$$

Applying the divergence theorem results in:

$$\int_\Omega (\nabla \times \boldsymbol{w}) \cdot (\nu \nabla \times \boldsymbol{u}) - \int_{\partial\Omega} (\boldsymbol{w} \times \nu \nabla \times \boldsymbol{u}) \cdot \boldsymbol{n} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w} \tag{22}$$

We will now apply the following relationship:

$$\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \boldsymbol{b} \cdot (\boldsymbol{c} \times \boldsymbol{a}) \tag{23}$$

Doing so results in

$$\int_\Omega (\nabla \times \boldsymbol{w}) \cdot (\nu \nabla \times \boldsymbol{u}) - \int_{\partial\Omega} \boldsymbol{w} \cdot (\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w} \tag{24}$$

Our boundary condition was that $\boldsymbol{u} \times \boldsymbol{n} = 0$ on the border, hence the term is zero.

$$\int_\Omega (\nabla \times \boldsymbol{w}) \cdot (\nu \nabla \times \boldsymbol{u}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{w} \tag{25}$$

We see how the first term is the L^2 inner product $\langle \nabla \times \boldsymbol{w}, \nabla \times \boldsymbol{u} \rangle_{L^2}$. These two terms must belong to $L^2(\Omega)$ for their inner product to be well-defined. The space of functions whose curl is in $L^2(\Omega)$ is:

$$\boldsymbol{u}, \boldsymbol{w} \in \mathcal{H}(\text{curl}, \Omega) \tag{26}$$

Part B

If Γ is a surface that intersects Ω, obtain the transmission conditions across $\partial\Omega$ implied by regularity requirements. The regularity condition for an element of $\mathcal{H}(\text{curl}, \Omega)$ is

$$\int_\Omega (\nabla \times \boldsymbol{u})^2 < \infty \tag{27}$$

Let's make an approximation around a postulated discontinuity on $\boldsymbol{x}_0 \in \Gamma$, and $\boldsymbol{n}_1, \boldsymbol{n}_2$ being the unitary normal vectors pointing away from Ω_1 and Ω_2 respectively. Note that $\boldsymbol{n}_1 = -\boldsymbol{n}_2$.

$$\nabla \times \boldsymbol{u}^\epsilon = \begin{cases} \nabla \times \boldsymbol{u} & \boldsymbol{x} \in \Omega_1 \\ \frac{1}{2\epsilon}\left[\nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon) - \nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon)\right] & \varepsilon > 0 \\ \nabla \times \boldsymbol{u} & \boldsymbol{x} \in \Omega_2 \end{cases} \tag{28}$$

This replaces $\nabla \times \boldsymbol{u}$ in the neighbourhood of the border by a linear interpolation from $\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon$ to $\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon$. Hence, the L^2 norm is:

$$\begin{aligned} \int_\Omega (\nabla \times \boldsymbol{u}^\epsilon)^2 &= \int_{\Omega_1} (\nabla \times \boldsymbol{u})^2 + \int_{\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon}^{\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon} \left[\frac{\nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon) - \nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon)}{2\epsilon}\right]^2 + \int_{\Omega_2} (\nabla \times \boldsymbol{u})^2 \\ &= \int_{\Omega_1} (\nabla \times \boldsymbol{u})^2 + \left[\frac{\nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon) - \nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon)}{2\epsilon}\right]^2 + \int_{\Omega_2} (\nabla \times \boldsymbol{u})^2 \end{aligned}$$

Now we can approach the L^2 norm of$\nabla \times \boldsymbol{u}$ by taking the limit:

$$\begin{aligned} \int_\Omega (\nabla \times \boldsymbol{u})^2 &= \lim_{\epsilon \to 0} \int_\Omega (\nabla \times \boldsymbol{u}^\epsilon)^2 \\ &= \lim_{\epsilon \to 0} \left(\int_{\Omega_1} (\nabla \times \boldsymbol{u})^2 + 2\epsilon \left[\frac{\nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon) - \nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon)}{2\epsilon}\right]^2 + \int_{\Omega_2} (\nabla \times \boldsymbol{u})^2 \right) \\ &= \int_\Omega (\nabla \times \boldsymbol{u})^2 + \lim_{\epsilon \to 0} \left(2\epsilon \left[\frac{\nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon) - \nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon)}{2\epsilon}\right]^2 \right) \end{aligned}$$

We see now that the only way for them to be equal id for the top part of the fraction within the limit to be zero, and therefore:

$$\nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_1\epsilon) = \nabla \times \boldsymbol{u}(\boldsymbol{x}_0 - \boldsymbol{n}_2\epsilon) \qquad \text{for } \epsilon \to 0 \tag{29}$$

In other words, the curl must be continuous across the border. Otherwise its L^2 norm could grow ad infinitum as more borders are added, hence breaking expression 27. This condition can be formally expressed as strong condition:

$$[\![\nabla \times \boldsymbol{u}]\!] = 0 \tag{30}$$

Part C

Obtain the transmission conditions across Γ that follow by imposing in the variational form of the problem that the integral is additive.

We'll start by splitting equation 24 into two domains. We'll use Γ_i to refer to the border when approached from subdomain i. As stated previously, the border integral vanishes for all external borders, but there's no reason to believe the same is true for internal borders, so the term stays.

$$\left.\begin{aligned} \textstyle\int_{\Omega_1} (\nabla \times \boldsymbol{w}) \cdot (\nu \nabla \times \boldsymbol{u}) - \int_{\Gamma_1} \boldsymbol{w} \cdot (\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_1) = \int_{\Omega_1} \boldsymbol{f} \cdot \boldsymbol{w} \qquad & \text{in } \Omega_1 \\ \textstyle\int_{\Omega_2} (\nabla \times \boldsymbol{w}) \cdot (\nu \nabla \times \boldsymbol{u}) - \int_{\Gamma_2} \boldsymbol{w} \cdot (\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_2) = \int_{\Omega_2} \boldsymbol{f} \cdot \boldsymbol{w} \qquad & \text{in } \Omega_2 \end{aligned}\right\} \tag{31}$$

If we add up these two equations and subtract the weak form we reach:

$$-\int_{\Gamma_1} \boldsymbol{w} \cdot (\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_1) - \int_{\Gamma_2} \boldsymbol{w} \cdot (\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_2) = 0$$

Since $\Gamma_1 = \Gamma_2$:

$$\int_{\Gamma_1} \Big[\boldsymbol{w} \cdot (\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_1) + \boldsymbol{w} \cdot (\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_2) \Big] = 0$$

This can be stated as a weak transmission condition:

$$[\![\nu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_1]\!] = 0 \tag{32}$$

1.3 The Navier equations

The Navier equations for an elastic material can be written in three different ways:

$$\begin{aligned} -2\mu\nabla\cdot(\boldsymbol{\varepsilon}(\boldsymbol{u})) - \lambda\nabla(\nabla\cdot\boldsymbol{u}) &= \rho\boldsymbol{b} \\ -\mu\Delta\boldsymbol{u} - (\lambda+\mu)\nabla(\nabla\cdot\boldsymbol{u}) &= \rho\boldsymbol{b} \\ \mu\nabla\times(\nabla\times\boldsymbol{u}) - (\lambda+2\mu)\nabla(\nabla\cdot\boldsymbol{u}) &= \rho\boldsymbol{b} \end{aligned}$$

where u is the displacement field, $\varepsilon(\boldsymbol{u})$ the symmetric part of $\nabla\boldsymbol{u}$, λ and μ the Lamé coefficients, ρ the density of the material and b the body forces. Let us assume that $u = 0$ on $\partial\Omega$

Part A

Write down the variational form of the previous equations in the appropriate functional spaces.

Let's start off working on the first equation. We'll use vector test functions $\boldsymbol{w}$:

$$-\int_\Omega 2\mu(\nabla\cdot\nabla^s\boldsymbol{u})\cdot\boldsymbol{w} - \int_\Omega \lambda\boldsymbol{w}\cdot\boldsymbol{\nabla}(\nabla\cdot\boldsymbol{u}) = \int_\Omega \rho\boldsymbol{w}\cdot\boldsymbol{b}$$

We can integrate by parts on both terms of the left hand side and apply the divergence theorem as per usual to obtain:

$$\int_\Omega 2\mu\nabla\boldsymbol{w} : \nabla^s\boldsymbol{u} + \int_\Omega \lambda(\nabla\cdot\boldsymbol{w})(\nabla\cdot\boldsymbol{u}) - \int_{\partial\Omega} 2\mu\boldsymbol{w}(\nabla^s\boldsymbol{u})\boldsymbol{n} - \int_{\partial\Omega}\lambda\boldsymbol{w}\cdot(\nabla\cdot\boldsymbol{u})\boldsymbol{n} = \int_\Omega \rho\boldsymbol{w}\cdot\boldsymbol{b} \tag{33}$$

Since all boundaries have Dirichlet boundary conditions with value 0 we can use a funcion space with compact support for $\boldsymbol{w}$, meaning $\boldsymbol{w}\big|_{\partial\Omega} = 0$. This makes two terms vanish to result in the variational form of the first equation:

$$\int_\Omega 2\mu\nabla\boldsymbol{w} : \nabla^s\boldsymbol{u} + \int_\Omega \lambda(\nabla\cdot\boldsymbol{w})(\nabla\cdot\boldsymbol{u}) = \int_\Omega \rho\boldsymbol{w}\cdot\boldsymbol{b} \tag{34}$$

Moving on to the second equation:

$$-\int_\Omega \boldsymbol{w}\cdot\mu(\nabla\cdot\nabla\boldsymbol{u}) - \int_\Omega \boldsymbol{w}\cdot(\lambda+\mu(\nabla(\nabla\cdot\boldsymbol{u}) = \int_\Omega \rho\boldsymbol{w}\cdot\boldsymbol{b}$$

Repeating the same process as before:

$$\int_\Omega \mu\nabla\boldsymbol{w} : \nabla\boldsymbol{u} + \int_\Omega (\lambda+\mu)(\nabla\cdot\boldsymbol{w})(\nabla\cdot\boldsymbol{u}) - \int_{\partial\Omega}\boldsymbol{w}\cdot\mu(\nabla\boldsymbol{u})\boldsymbol{n} - \int_{\partial\Omega}\boldsymbol{w}\cdot(\lambda+\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n} = \int_\Omega \rho\boldsymbol{w}\cdot\boldsymbol{b} \tag{35}$$

Considering the compact support results in the variational form of the second equation:

$$\int_\Omega \mu\nabla\boldsymbol{w} : \nabla\boldsymbol{u} + \int_\Omega (\lambda+\mu)(\nabla\cdot\boldsymbol{w})(\nabla\cdot\boldsymbol{u}) = \int_\Omega \rho\boldsymbol{w}\cdot\boldsymbol{b} \tag{36}$$

The third equation is simple since we can solve it via analogy. Notice how the first term is identical to the first of the Maxwell equations. The second term is identical to the second term of the second Navier equation, save for a small change in the constant. The term on the right hand side is the same as the previous two equations as well. We can immediately resolve that the variational form is:

$$\int_\Omega (\nabla\times\boldsymbol{w})\cdot(\mu\nabla\times\boldsymbol{u}) + \int_\Omega (\lambda+2\mu)(\nabla\cdot\boldsymbol{w})(\nabla\cdot\boldsymbol{u}) - \int_{\partial\Omega}\boldsymbol{w}\cdot(\mu\nabla\times\boldsymbol{u}\times\boldsymbol{n}) - \int_{\partial\Omega}\boldsymbol{w}\cdot(\lambda+2\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n} = \int_\Omega \rho\boldsymbol{w}\cdot\boldsymbol{b} \tag{37}$$

Once again considering the compact support results in the variational form of the third equation:

$$\int_\Omega (\nabla \times \boldsymbol{w}) \cdot (\mu \nabla \times \boldsymbol{u}) + \int_\Omega (\lambda + 2\mu)(\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) = \int_\Omega \rho \boldsymbol{w} \cdot \boldsymbol{b} \tag{38}$$

Then, the system all put together looks like:

$$\int_\Omega 2\mu \nabla \boldsymbol{w} : \nabla^s \boldsymbol{u} + \int_\Omega \lambda (\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) = \int_\Omega \rho \boldsymbol{w} \cdot \boldsymbol{b} \tag{39}$$

$$\int_\Omega \mu \nabla \boldsymbol{w} : \nabla \boldsymbol{u} + \int_\Omega (\lambda + \mu)(\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) = \int_\Omega \rho \boldsymbol{w} \cdot \boldsymbol{b} \tag{40}$$

$$\int_\Omega (\nabla \times \boldsymbol{w}) \cdot (\mu \nabla \times \boldsymbol{u}) + \int_\Omega (\lambda + 2\mu)(\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) = \int_\Omega \rho \boldsymbol{w} \cdot \boldsymbol{b} \tag{41}$$

The functional spaces can be deduced from these. We see that the L^2 inner products must be defined on the curl, divergence and gradient. We also know we have a compact support. We can therefore conclude that:

$$\boldsymbol{u}, \boldsymbol{w} \in \mathcal{H}_0^1(\text{curl}, \Omega) \cap \mathcal{H}_0^1(\text{div}, \Omega) \cap \mathcal{H}_0^1(\Omega) \tag{42}$$

Part B

If Γ is a surface that intersects Ω, obtain the transmission conditions across Γ that follow by imposing in the variational form of the problem that the integral is additive. We'll have to use equations 33, 35 and 37 but all the surface integrals vanish on the external border. The system is:

$$\begin{aligned}
&\int_{\Omega_1} 2\mu \nabla \boldsymbol{w} : \nabla^s \boldsymbol{u} + \int_{\Omega_1} (\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) - \int_{\Gamma_1} 2\mu \boldsymbol{w} (\nabla^s \boldsymbol{u}) \boldsymbol{n}_1 - \int_{\Gamma_1} \lambda \boldsymbol{w} \cdot (\nabla \cdot \boldsymbol{u}) \boldsymbol{n}_1 \\
+ &\int_{\Omega_2} 2\mu \nabla \boldsymbol{w} : \nabla^s \boldsymbol{u} + \int_{\Omega_2} (\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) - \int_{\Gamma_2} 2\mu \boldsymbol{w} (\nabla^s \boldsymbol{u}) \boldsymbol{n}_2 - \int_{\Gamma_2} \lambda \boldsymbol{w} \cdot (\nabla \cdot \boldsymbol{u}) \boldsymbol{n}_2 \\
&= \int_{\Omega_1} \rho \boldsymbol{w} \cdot \boldsymbol{b} + \int_{\Omega_2} \rho \boldsymbol{w} \cdot \boldsymbol{b}
\end{aligned}$$

$$\begin{aligned}
&\int_{\Omega_1} \mu \nabla \boldsymbol{w} : \nabla \boldsymbol{u} + \int_{\Omega_1} (\lambda + \mu)(\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) - \int_{\Gamma_1} \boldsymbol{w} \cdot \mu (\nabla \boldsymbol{u}) \boldsymbol{n}_1 - \int_{\Gamma_1} \boldsymbol{w} \cdot (\lambda + \mu)(\nabla \cdot \boldsymbol{u}) \boldsymbol{n}_1 \\
+ &\int_{\Omega_2} \mu \nabla \boldsymbol{w} : \nabla \boldsymbol{u} + \int_{\Omega_2} (\lambda + \mu)(\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) - \int_{\Gamma_2} \boldsymbol{w} \cdot \mu (\nabla \boldsymbol{u}) \boldsymbol{n}_2 - \int_{\Gamma_2} \boldsymbol{w} \cdot (\lambda + \mu)(\nabla \cdot \boldsymbol{u}) \boldsymbol{n}_2 \\
&= \int_{\Omega_1} \rho \boldsymbol{w} \cdot \boldsymbol{b} + \int_{\Omega_2} \rho \boldsymbol{w} \cdot \boldsymbol{b}
\end{aligned}$$

$$\begin{aligned}
&\int_{\Omega_1} (\nabla \times \boldsymbol{w}) \cdot (\mu \nabla \times \boldsymbol{u}) + \int_{\Omega_1} (\lambda + 2\mu)(\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) - \int_{\Gamma_1} \boldsymbol{w} \cdot (\mu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_1) - \int_{\Gamma_1} \boldsymbol{w} \cdot (\lambda + 2\mu)(\nabla \cdot \boldsymbol{u}) \boldsymbol{n}_1 \\
+ &\int_{\Omega_2} (\nabla \times \boldsymbol{w}) \cdot (\mu \nabla \times \boldsymbol{u}) + \int_{\Omega_2} (\lambda + 2\mu)(\nabla \cdot \boldsymbol{w})(\nabla \cdot \boldsymbol{u}) - \int_{\Gamma_2} \boldsymbol{w} \cdot (\mu \nabla \times \boldsymbol{u} \times \boldsymbol{n}_2) - \int_{\Gamma_2} \boldsymbol{w} \cdot (\lambda + 2\mu)(\nabla \cdot \boldsymbol{u}) \boldsymbol{n}_2 \\
&= \int_{\Omega_1} \rho \boldsymbol{w} \cdot \boldsymbol{b} + \int_{\Omega_2} \rho \boldsymbol{w} \cdot \boldsymbol{b}
\end{aligned}$$

By taking these expressions and subtracting the weak forms we obtain:

$$-\int_{\Gamma_1} 2\mu \boldsymbol{w}(\nabla^s \boldsymbol{u})\boldsymbol{n}_1 - \int_{\Gamma_1} \lambda \boldsymbol{w}\cdot(\nabla\cdot\boldsymbol{u})\boldsymbol{n}_1 - \int_{\Gamma_2} 2\mu \boldsymbol{w}(\nabla^s \boldsymbol{u})\boldsymbol{n}_2 - \int_{\Gamma_2} \lambda \boldsymbol{w}\cdot(\nabla\cdot\boldsymbol{u})\boldsymbol{n}_2 = 0$$

$$-\int_{\Gamma_1} \boldsymbol{w}\cdot\mu(\nabla\boldsymbol{u})\boldsymbol{n}_1 - \int_{\Gamma_1} \boldsymbol{w}\cdot(\lambda+\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n}_1 - \int_{\Gamma_2} \boldsymbol{w}\cdot\mu(\nabla\boldsymbol{u})\boldsymbol{n}_2 - \int_{\Gamma_2} \boldsymbol{w}\cdot(\lambda+\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n}_2 = 0$$

$$-\int_{\Gamma_1} \boldsymbol{w}\cdot(\mu\nabla\times\boldsymbol{u}\times\boldsymbol{n}_1) - \int_{\Gamma_1} \boldsymbol{w}\cdot(\lambda+2\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n}_1 - \int_{\Gamma_2} \boldsymbol{w}\cdot(\mu\nabla\times\boldsymbol{u}\times\boldsymbol{n}_2) - \int_{\Gamma_2} \boldsymbol{w}\cdot(\lambda+2\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n}_2 = 0$$

Therefore the conditions are:

$$\begin{aligned}
[\![\mu(\nabla^s \boldsymbol{u})\boldsymbol{n}]\!] &= 0\\
[\![\lambda(\nabla\cdot\boldsymbol{u})\boldsymbol{n}]\!] &= 0\\
[\![\mu(\nabla\boldsymbol{u})\boldsymbol{n}]\!] &= 0\\
[\![(\lambda+\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n}]\!] &= 0\\
[\![\mu\nabla\times\boldsymbol{u}\times\boldsymbol{n}]\!] &= 0\\
[\![(\lambda+2\mu)(\nabla\cdot\boldsymbol{u})\boldsymbol{n}]\!] &= 0
\end{aligned}$$

These can be reduced in the cases where μ, λ are continuous, and considering that a continuity of $\nabla\boldsymbol{u}$ implies a continuity on $\nabla^s\boldsymbol{u}$, the reduced version becomes:

$$\begin{aligned}
[\![(\nabla\cdot\boldsymbol{u})\boldsymbol{n}]\!] &= 0\\
[\![(\nabla\boldsymbol{u})\boldsymbol{n}]\!] &= 0\\
[\![\nabla\times\boldsymbol{u}\times\boldsymbol{n}]\!] &= 0
\end{aligned}$$

2 Domain Decomposition Methods

2.1 Schwartz method

Consider Problem 1 of Section 1. Let $[0, L] = [0, L_1] \cup [L_2, L]$, *with* $L_2 < L_1$

Part A

Write down an iteration-by-subdomain scheme based on a Schwarz additive domain decomposition method.

We will start by declaring a differential operator:

$$\mathcal{L} := EI \frac{\mathrm{d}^4}{\mathrm{d}x^4} \tag{43}$$

Hence our problem becomes:

$$\left.\begin{array}{rl} \mathcal{L}u = f & \text{in } \Omega \\ u = 0 & \text{on } \Gamma_1 \\ u = 0 & \text{on } \Gamma_2 \end{array}\right\} \tag{44}$$

If we split it into two semi-overlapping domains Ω_1 and Ω_2, with their new interfaces $\Gamma_1 2$ and $\Gamma_2 1$, we obtain the following system:

$$\left.\begin{array}{rl} \mathcal{L}u_1 = f & \text{in } \Omega_1 \\ u_1 = 0 & \text{on } \Gamma_1 \\ u_1 = u_2 & \text{on } \Gamma_{12} \end{array}\right\} \qquad \left.\begin{array}{rl} \mathcal{L}u_2 = f & \text{in } \Omega_2 \\ u_2 = 0 & \text{on } \Gamma_2 \\ u_2 = u_1 & \text{on } \Gamma_{21} \end{array}\right\} \tag{45}$$

We can then iterate by using the previous iteration's result as interface boundary condition:

$$\left.\begin{array}{rl} \mathcal{L}u_1^k = f & \text{in } \Omega_1 \\ u_1^k = 0 & \text{on } \Gamma_1 \\ u_1^k = u_2^{(k-1)} & \text{on } \Gamma_{12} \end{array}\right\} \qquad \left.\begin{array}{rl} \mathcal{L}u_2^k = f & \text{in } \Omega_2 \\ u_2^k = 0 & \text{on } \Gamma_2 \\ u_2^k = u_1^l & \text{on } \Gamma_{21} \end{array}\right\} \tag{46}$$

where $l = k - 1$ if we plan on using a Jacobi solver or $l = k$ if we choose Gauss-Seidel.

Part B

Obtain the matrix version of the previous scheme once space has been discretized using finite elements.

We can define L as the discretized operator $\mathcal{L}$. We must distinguish nodes within the domains and those lying on the interfaces. To do so I will use the symbol I for interior nodes, Γ_{12} for the interface contained within Ω_2 and $\Gamma_2 1$ for the other interface. The matrix operations to solve are the following:

$$\left(u^{(1)}_{\Gamma_{12}}\right)^k = \left(u^{(2)}_{\Gamma_{12}}\right)^{k-1} \qquad \left(u^{(2)}_{\Gamma_{21}}\right)^k = \left(u^{(1)}_{\Gamma_{21}}\right)^{l}$$

$$L^{(1)}_{II}\left(u^{(1)}_I\right)^k = \mathrm{f}^{(1)}_I - L^{(1)}_{\Gamma_{12}I}\left(u^{(1)}_{\Gamma_{12}}\right)^k \qquad L^{(2)}_{II}\left(u^{(2)}_I\right)^k = \mathrm{f}^{(2)}_I - L^{(2)}_{\Gamma_{21}I}\left(u^{(2)}_{\Gamma_{21}}\right)^k$$

One missing piece in the above expression is that the values of u_Γ have to be initialized before the first step with a first guess. For the Gauss-Seidel method only the values on Γ_{12} are needed, but for the Jacobi the initial guess on Γ_{21} is needed as well. The other missing piece are the outside boundary conditions, for which any method such as Lagrange multipliers, penalty method or master-slave elimination will suffice.

2.2 Dirichlet-Neumann Coupling

Consider Problem 2 of Section 1. *Let* Γ *be a surface that intersects* Ω

Part A

Write down an iteration-by-subdomain scheme based on the Dirichlet-Neumann coupling.

The system before can be described by:

$$\begin{aligned} \nu\nabla\times\nabla\times\boldsymbol{u} &= \boldsymbol{f} \quad \text{in } \Omega \\ \nabla\cdot\boldsymbol{u} &= 0 \quad \text{in } \Omega \\ \boldsymbol{n}\times\boldsymbol{u} &= 0 \quad \text{on } \Gamma \end{aligned}$$

Hence, we must define the following differential operators:

$$\begin{aligned} \mathcal{C} &= \nu\nabla\times\nabla\times \\ \mathcal{D} &= \nabla\cdot \end{aligned}$$

Then the system can be stated as:

$$\left.\begin{aligned} \mathcal{C}\boldsymbol{u} &= \boldsymbol{f} \quad \text{in } \Omega \\ \mathcal{D}\boldsymbol{u} &= 0 \quad \text{in } \Omega \\ \boldsymbol{n}\times\boldsymbol{u} &= 0 \quad \text{on } \Gamma \end{aligned}\right\} \tag{47}$$

We must also add the transmission conditions.

$$\begin{aligned} &\text{1st condition:} && [\![\boldsymbol{n}\times\boldsymbol{u}]\!] \\ &\text{2nd condition:} && [\![\nu\nabla\times\boldsymbol{u}\times\boldsymbol{n}]\!] \end{aligned}$$

In a Dirichlet-Neumann scheme, the first subdomain has a weak Neumann transmission condition and the second one has a strong Dirichlet one.

$$\left.\begin{aligned} \mathcal{C}\boldsymbol{u}_1^k &= \boldsymbol{f} && \text{in } \Omega_1 \\ \mathcal{D}\boldsymbol{u}_1^k &= 0 && \text{in } \Omega_1 \\ \boldsymbol{n}\times\boldsymbol{u}_1^k &= 0 && \text{on } \Gamma_1 \\ \nu_1\nabla\times\boldsymbol{u}_1^k\times\boldsymbol{n} &= \nu_2\nabla\times\boldsymbol{u}_2^{k-1}\times\boldsymbol{n} && \text{on } \Gamma \end{aligned}\right\} \qquad \left.\begin{aligned} \mathcal{C}\boldsymbol{u}_2^k &= \boldsymbol{f} && \text{in } \Omega_2 \\ \mathcal{D}\boldsymbol{u}_2^k &= 0 && \text{in } \Omega_2 \\ \boldsymbol{n}\times\boldsymbol{u}_2^k &= 0 && \text{on } \Gamma_2 \\ \boldsymbol{n}\times\boldsymbol{u}_2^k &= \boldsymbol{n}\times\boldsymbol{u}_1^l && \text{on } \Gamma \end{aligned}\right\} \tag{48}$$

where once again we must use $l = k-1$ if we prefer a Jacobi solver or $l = k$ if we choose Gauss-Seidel.

Part B

Obtain the expression of the Steklov-Poincaré operator of the problem.

We'll start off by separating the system using the expression $\boldsymbol{u}_i = \boldsymbol{u}_i^0 + \tilde{\boldsymbol{u}}_i$, where $i = 1, 2$ and $i \neq j = 1, 2$:

$$\left.\begin{aligned} \mathcal{C}\boldsymbol{u}_i^0 &= \boldsymbol{f} && \text{in } \Omega_i \\ \mathcal{D}\boldsymbol{u}_i^0 &= 0 && \text{in } \Omega_i \\ \boldsymbol{n}\times\boldsymbol{u}_i^0 &= 0 && \text{on } \Gamma_i \\ \boldsymbol{n}\times\boldsymbol{u}_i^0 &= 0 && \text{on } \Gamma_{ij} \end{aligned}\right\} \qquad \left.\begin{aligned} \mathcal{C}\tilde{\boldsymbol{u}}_i &= 0 && \text{in } \Omega_i \\ \mathcal{D}\tilde{\boldsymbol{u}}_i &= 0 && \text{in } \Omega_i \\ \boldsymbol{n}\times\tilde{\boldsymbol{u}}_i &= 0 && \text{on } \Gamma_i \\ \boldsymbol{n}\times\tilde{\boldsymbol{u}}_i &= \varphi && \text{on } \Gamma_{ij} \end{aligned}\right\}$$

We also must ensure the weak continuity on Γ:

$$\nu_1 \nabla \times \boldsymbol{u}_1 \times \boldsymbol{n} = \nu_2 \nabla \times \boldsymbol{u}_2 \times \boldsymbol{n}$$
$$\nu_1 \nabla \times \boldsymbol{u}_1^0 \times \boldsymbol{n} - \nu_2 \nabla \times \boldsymbol{u}_2^0 \times \boldsymbol{n} = -\nu_1 \nabla \times \tilde{\boldsymbol{u}}_1 \times \boldsymbol{n} + \nu_2 \nabla \times \tilde{\boldsymbol{u}}_2 \times \boldsymbol{n}$$

We can now define the Steklov-Pointcaré operator:

$$\mathcal{S}: \quad \begin{array}{c} \mathcal{H}^{1/2}(\Gamma) \longrightarrow \mathcal{H}^{-1/2}(\Gamma) \\ \varphi \longrightarrow \nu_1 \nabla \times \tilde{\boldsymbol{u}}_1 \times \boldsymbol{n} - \nu_2 \nabla \times \tilde{\boldsymbol{u}}_2 \times \boldsymbol{n} \end{array} \tag{49}$$

we will also define:

$$\mathcal{G} = -\nu_1 \nabla \times \boldsymbol{u}_1^0 \times \boldsymbol{n} + \nu_2 \nabla \times \boldsymbol{u}_2^0 \times \boldsymbol{n} \quad \in \mathcal{H}^{-1/2}(\Gamma) \tag{50}$$

The system then looks like:

$$\mathcal{S}\varphi = \mathcal{G} \tag{51}$$

Part C

Obtain the matrix version of the previous scheme once space has been discretized using finite elements.

First off, we'll define finite element space discretized operators $C :=$ FEM$(\mathcal{C})$, $D :=$ FEM$(\mathcal{D})$. Now we can move towards matrix forms. We can write:

$$Ax = b$$
$$\begin{bmatrix} C & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}$$

We are using Lagrange multipliers to impose the non-divergence condition. From this point on we'll work with $Ax = b$ so any other manner of applying this condition is also valid. We'll use matrix E to enforce the Neumann boundary condition:

$$E = \begin{bmatrix} 0_{II} & 0_{I\Gamma} \\ A_{I\Gamma}^T & A_{\Gamma\Gamma}^T \end{bmatrix} \tag{52}$$

It can be seen that it's simply A^T with it's top half removed. This is valid only because A is SPD and done to make the notation less cumbersome in the next expressions. The system then becomes, in domain 1:

$$A^{(1)} x^{(1)} = \mathrm{b}^{(1)} - E \left(x^{(2)} \right)^{k-1} \tag{53}$$

Notice the Neumann boundary condition. Domain 2 with its Dirichlet boundary condition will look like:

$$A_{II}^{(2)} \left(x_I^{(2)} \right)^k = \mathrm{b}_I^{(2)} - A_{\Gamma I}^{(2)} \left(x_\Gamma^{(1)} \right)^l \tag{54}$$

As before, we must use $l = k - 1$ if we prefer a Jacobi solver or $l = k$ if we choose Gauss-Seidel.

2.3 Poisson equation

Consider the problem of finding $u : \Omega \longrightarrow \mathbb{R}$ *such that*

$$\begin{aligned} -k\Delta u &= f \quad \textit{in } \Omega \\ u &= 0 \quad \textit{on } \partial\Omega \end{aligned}$$

where $k > 0$. *Let* Γ *be a surface crossing* Ω

Part A

Write down an iteration-by-subdomain scheme based on the Dirichlet-Robin coupling.

We'll use the differential operator $\mathcal{L} = -k\nabla \cdot \nabla$ to obtain the expression:

$$\left.\begin{aligned} \mathcal{L}u &= f \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \partial\Omega \end{aligned}\right\} \tag{55}$$

Now our problem becomes:

$$\left.\begin{aligned} \mathcal{L}u_1 &= f && \text{in } \Omega_1 \\ u_1 &= 0 && \text{on } \partial\Omega_1 \\ \mathcal{L}u_2 &= f && \text{in } \Omega_2 \\ u_2 &= 0 && \text{on } \partial\Omega_2 \\ k_1\partial_n u_1 + \gamma_1 u_1 &= k_2\partial_n u_2 + \gamma_2 u_2 && \text{on } \Gamma \end{aligned}\right\} \tag{56}$$

To enforce this scheme we solve the following system:

$$\left.\begin{aligned} \mathcal{L}u_1^k &= f && \text{in } \Omega_1 \\ u_1^k &= 0 && \text{on } \partial\Omega_1 \\ (k_1\partial_n + \gamma_1)u_1^k &= (k_2\partial_n + \gamma_2)u_2^{k-1} && \text{on } \Gamma \end{aligned}\right\} \qquad \left.\begin{aligned} \mathcal{L}u_2^k &= f && \text{in } \Omega_2 \\ u_2^k &= 0 && \text{on } \partial\Omega_2 \\ (k_2\partial_n + \gamma_2)u_2^k &= (k_1\partial_n + \gamma_1)u_1^l && \text{on } \Gamma \end{aligned}\right\}$$

Once again, we must use $l = k - 1$ if we prefer a Jacobi solver or $l = k$ if we choose Gauss-Seidel.

Part B

Obtain the matrix version of the previous scheme once space has been discretized using finite elements.

The discretized operator $\mathcal{L}$ is matrix A. The system in domain 1 looks like:

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Gamma}^{(1)} \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} (u_I^{(1)})^k \\ (u_\Gamma^{(1)})^k \end{bmatrix} = \begin{bmatrix} \mathrm{f}_I^{(1)} \\ \mathrm{f}_\Gamma^{(1)} - A_{\Gamma I}^{(2)}(u_I^{(2)})^{k-1} - A_{\Gamma\Gamma}^{(2)}(u_\Gamma^{(2)})^{k-1} \end{bmatrix} \tag{57}$$

For domain number 2 it looks like:

$$\begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} (u_I^{(2)})^k \\ (u_\Gamma^{(2)})^k \end{bmatrix} = \begin{bmatrix} \mathrm{f}_I^{(2)} \\ \mathrm{f}_\Gamma^{(2)} - A_{\Gamma I}^{(1)}(u_I^{(2)})^l - A_{\Gamma\Gamma}^{(1)}(u_\Gamma^{(1)})^l \end{bmatrix} \tag{58}$$

Yet again, we must use $l = k - 1$ if we prefer a Jacobi solver or $l = k$ if we choose Gauss-Seidel. Our initial guess will consist of $(u^{(2)})^0$ for the Gauss-Seidel method, but for the Jacobi we'll also have to add $(\lambda^{(1)})^0$.

Part C

Obtain the Schur complement as discrete version of the Steklov-Poincare operator.

Our problem looks like:

$$\begin{bmatrix} A_{II}^{(1)} & A_{I\Gamma}^{(1)} & 0 \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma} & A_{\Gamma I}^{(2)} \\ 0 & A_{\Gamma I}^{(2)} & A_{II}^{(2)} \end{bmatrix} \begin{bmatrix} u_I^{(1)} \\ u_\Gamma \\ u_I^{(2)} \end{bmatrix} = \begin{bmatrix} \mathrm{f}_I^{(1)} \\ \mathrm{f}_\Gamma \\ \mathrm{f}_I^{(2)} \end{bmatrix} \tag{59}$$

By re-arranging the first and last rows we obtain the following two equations:

$$\begin{aligned} u_I^{(1)} &= (A_{II}^{(1)})^{-1} \left(\mathrm{f}_I^{(1)} - A_{I\Gamma}^{(1)} u_\Gamma \right) \\ u_I^{(2)} &= (A_{II}^{(2)})^{-1} \left(\mathrm{f}_I^{(2)} - A_{I\Gamma}^{(2)} u_\Gamma \right) \end{aligned} \tag{60}$$

Using them in the second row of the matrix equation results in:

$$\left(A_{\Gamma\Gamma} - A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} A_{I\Gamma}^{(1)} - A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} A_{I\Gamma}^{(2)} \right) u_\Gamma = \mathrm{f}_\Gamma - A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} \mathrm{f}_I^{(1)} - A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} \mathrm{f}_I^{(2)} \tag{61}$$

Which can be abreviated as:

$$S u_\Gamma = G \tag{62}$$

Where the Schur complement S is the discrete version of the Steklov-Pointcaré operator $\mathcal{S}$:

$$S = A_{\Gamma\Gamma} - A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} A_{I\Gamma}^{(1)} - A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} A_{I\Gamma}^{(2)} \tag{63}$$

and G is:

$$G = \mathrm{f}_\Gamma - A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} \mathrm{f}_I^{(1)} - A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} \mathrm{f}_I^{(2)} \tag{64}$$

Part D

Identify the preconditioner for the Schur complement equation arising from the iterative scheme of section A.

We can first split the Schur complement into two parts:

$$\begin{aligned} S^{(1)} &= A_{\Gamma\Gamma}^{(1)} - A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} A_{I\Gamma}^{(1)} \\ S^{(2)} &= A_{\Gamma\Gamma}^{(2)} - A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} A_{I\Gamma}^{(2)} \end{aligned}$$

If we take equation 57 and 58 we can see it yields:

$$\begin{aligned} (u_I^{(1)})^k &= (A_{II}^{(1)})^{-1} \left(\mathrm{f}_I^{(1)} - A_{I\Gamma}^{(1)} u_\Gamma^k \right) \\ (u_I^{(2)})^k &= (A_{II}^{(2)})^{-1} \left(\mathrm{f}_I^{(2)} - A_{I\Gamma}^{(2)} (u_\Gamma^{(2)})^k \right) \\ A_{\Gamma I}^{(1)} (u_I^{(1)})^k + A_{\Gamma\Gamma}^{(1)} u_\Gamma^k &= \mathrm{f}_\Gamma^{(1)} - A_{\Gamma I}^{(2)} (u_I^{(2)})^{k-1} - A_{\Gamma\Gamma}^{(2)} (u_\Gamma^{(2)})^{k-1} \end{aligned}$$

We can take the first and second equations and substitute them in the third one:

$$A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} \left(\mathrm{f}_I^{(1)} - A_{I\Gamma}^{(1)} u_\Gamma^k \right) + A_{\Gamma\Gamma}^{(1)} u_\Gamma^{(1)} = \mathrm{f}_\Gamma^{(1)} - A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} \left(\mathrm{f}_I^{(2)} - A_{I\Gamma}^{(2)} (u_\Gamma^{(2)})^{k-1} \right) - A_{\Gamma\Gamma}^{(2)} (u_\Gamma^{(2)})^{k-1} \tag{65}$$

Expanding the terms results in:

$$\begin{aligned} A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} \mathrm{f}_I^{(1)} - A_{\Gamma I}^{(1)} (A_{II}^{(1)})^{-1} A_{I\Gamma}^{(1)} u_\Gamma^k + A_{\Gamma\Gamma}^{(1)} (u_\Gamma^{(1)})^k \\ = \mathrm{f}_\Gamma^{(1)} - A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} \mathrm{f}_I^{(2)} + A_{\Gamma I}^{(2)} (A_{II}^{(2)})^{-1} A_{I\Gamma}^{(2)} (u_\Gamma^{(2)})^{k-1} - A_{\Gamma\Gamma}^{(2)} (u_\Gamma^{(2)})^{k-1} \end{aligned}$$

Rearranging them results in:

$$\Big(A_{\Gamma\Gamma}^{(1)} - A_{\Gamma I}^{(1)}(A_{II}^{(1)})^{-1}A_{I\Gamma}^{(1)}\Big)u_\Gamma^k \\ = \mathrm{f}_\Gamma^{(1)} - A_{\Gamma I}^{(1)}(A_{II}^{(1)})^{-1}\mathrm{f}_I^{(1)} - A_{\Gamma I}^{(2)}(A_{II}^{(2)})^{-1}\mathrm{f}_I^{(2)} + \Big(A_{\Gamma I}^{(2)}(A_{II}^{(2)})^{-1}A_{I\Gamma}^{(2)} - A_{\Gamma\Gamma}^{(2)}\Big)(u_\Gamma^{(2)})^{k-1}$$

Notice that this is equivalent to:

$$S^{(1)}u_\Gamma^k = G - S^{(2)}u_\Gamma^{k-1} \tag{66}$$

Left-multiplying by the inverse of S, and expanding $S^{(2)} = S - S^{(1)}$:

$$u_\Gamma^k = (S^{(1)})^{-1}G - (S^{(1)})^{-1}Su_\Gamma^{k-1} + u_\Gamma^{k-1} \tag{67}$$

Hence, rearranging finally results in:

$$u_\Gamma^k = u_\Gamma^{k-1} + (S^{(1)})^{-1}\left(G - Su_\Gamma^{k-1}\right) \tag{68}$$

This is clearly a Richardson fixed-point iteration with preconditioner $S^{(1)}$.

3 Coupling of Heterogeneous Problems

3.1 Plane stress

Consider the beam described in Problem 1 of Section 1. Apart from being clamped at $x = 0$ and $x = L$, the beam is supported on an elastic wall that occupies the square $[0, L] \times [-L, 0]$, where $y = 0$ corresponds to the beam axis. The wall is clamped everywhere except on the upper wall, where the beam is.

The wall displacements in the x- and y-directions are u and v, respectively, and the elastic properties E (Young modulus) and ν (Poisson's coefficient). No loads are applied on the wall, except for those coming from the beam.

Part A

Write down the equations in the wall assuming a plane stress behavior.

First we must state the constitutive law for plane stress:

$$D = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{69}$$

Then our stress-strain relationshitp looks like:

$$\boldsymbol{\sigma} = D\boldsymbol{\varepsilon} \tag{70}$$

note that we are using Voigt notation, hence the way to operate the differential operators will be different than usually. For this reason we'll label them ∇^V.

$$\nabla^V = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \tag{71}$$

We also know the following relationship:

$$\boldsymbol{\varepsilon} = \nabla^V \boldsymbol{u} \tag{72}$$

And the balance of forces equation:

$$-\nabla^V \cdot \boldsymbol{\sigma} = \boldsymbol{f} \tag{73}$$

Combining these three results, and acknowledging that there are no body loads, results in:

$$-\nabla^V \cdot D\nabla^V \boldsymbol{u} = 0 \tag{74}$$

Now adding the boundary conditions results in the full boundary value problem:

$$\left.\begin{aligned} -\nabla^V \cdot D\nabla^V \boldsymbol{u}_2 = 0 \quad & \text{in } \Omega_2 \\ u_2 = 0 \quad & \text{on } \partial\Omega_2 \\ v_2 = 0 \quad & \text{on } \partial\Omega_2 \backslash \Gamma \\ v_2 = v_1 \quad & \text{on } \Gamma \end{aligned}\right\} \tag{75}$$

we use subindex 2 to refer to the plate and reserve subindex 1 for the beam. We'll use Γ to refer to the shared boundary, and $\partial\Omega_i$ to refer to the outside boundaries.

Part B

Write down the equations for the beam modified because of the presence of the wall.

We can start with the original equation, and add to it the force of the plate:

$$EI\frac{\mathrm{d}^4 v_1}{\mathrm{d}x^4} = f + T \tag{76}$$

where T is the traction imparted by the wall onto the beam. Hence the full boundary problem will be:

$$\left.\begin{aligned} EI\frac{\mathrm{d}^4 v_1}{\mathrm{d}x^4} = f + T \quad & \text{in } \Omega_1 \\ v_1 = 0 \quad & \text{on } \partial\Omega_1 \\ v_1 = v_2 \quad & \text{on } \Gamma \end{aligned}\right\} \tag{77}$$

Part C

Obtain the adequate transmission conditions for v and the normal component of the traction on the wall at $y = 0$

The transmission condition for v looks like:

$$[\![v]\!] = v_2 - v_1 = 0 \tag{78}$$

And the transmission condition regarding the traction is:

$$[\![\tau]\!] = T - \boldsymbol{\sigma}_2 \cdot \boldsymbol{n}_2 = 0$$

This can be further expanded into:

$$[\![\tau]\!] = T - \boldsymbol{n}_2 \cdot D\nabla^V \boldsymbol{u}_2 = 0 \tag{79}$$

Part D

Suggest transmission conditions for u and the tangent component of the traction on the wall at $y = 0$. Discuss the implications if this component is not assumed to be zero.

Euler-Bernoulli beam theory is based on the premise that the displacements are so small that their tangent components are neglected. Hence any attempt at coupling u_2 with some derived u_1 would lead to meaningless results. If one wanted to impose such conditions, then the model should be upgraded to a more generalized one, such as Timoshenko-Ehrenfest beam theory. This theory accounts for shear deformation and uses two variables:

- v is the vertical displacement, just as before.
- θ is the rotation.

Hence we could express $u_1 = v_1 \sin(\theta_1) \approx v_1\theta_1$. Then the new transmission condition would look like:

$$[\![u]\!] = v_1\theta_1 - u_2 = 0 \tag{80}$$

This will barely affect the result, since u_1 will be very small in comparisson to v_1, but it will make the solving process more difficult and expensive since this is a non-linear transmission condition. All in all it's important to consider whether the gain is worth the cost.

One is free to chose any other model for beam deformation, this one was chosen because it's the simplest one that fits the requirements.

3.2 Dirichlet to Neumann operators

Let $\mathcal{S}_D$ and $\mathcal{S}_S$ be the Dirichlet-to-Neumann operators for the Darcy and the Stokes problems, respectively (see the class notes, chapter 3). The Steklov-Poincaré equation can be written as

$$\mathcal{S}_S(\lambda) = \mathcal{S}_D(\lambda) \tag{81}$$

where λ is the normal velocity on Γ, the interface between the Darcy and the Stokes regions.

Part A

Obtain the discrete version of the previous equation when space is discretized using finite elements. Relate the resulting matrices to those arising from the discretization of the Darcy and the Stokes problems separately.

First off let's write the matrix form of the Stokes problem:

$$\begin{bmatrix} K_S & G_S \\ G_S^T & 0 \end{bmatrix} \begin{bmatrix} u_S \\ p_S \end{bmatrix} = \begin{bmatrix} \mathrm{f}_S \\ \mathrm{h}_S \end{bmatrix} \tag{82}$$

where the matrices are defined as:

$$\begin{aligned} (K_S)_{ij} &= \int_\Omega \nabla \boldsymbol{w}_i \cdot \nu \nabla \boldsymbol{w}_j \\ (G_S)_{ij} &= \int_\Omega \boldsymbol{w}_i \nabla q_j \\ (\mathrm{f}_S)_{ij} &= \int_\Omega \boldsymbol{w}_i f + \int_{\partial\Omega} w_i (\boldsymbol{n}_S \cdot (-p_S \boldsymbol{I} + \nu \nabla \boldsymbol{u}_S)) \\ (\mathrm{h}_S)_{ij} &= \int_{\partial\Omega} q_i \boldsymbol{n} \cdot \boldsymbol{u}_S \end{aligned}$$

where $\boldsymbol{w} \in \mathcal{H}^1(\Omega)$ and $q \in \mathcal{L}^2(\Omega)$ are test vector and test scalar functions respectively. The Darcy problem is similar and results in the following system:

$$\begin{bmatrix} M_D & G_D \\ G_D^T & 0 \end{bmatrix} \begin{bmatrix} u_D \\ \varphi_D \end{bmatrix} = \begin{bmatrix} 0 \\ \mathrm{h}_D \end{bmatrix} \tag{83}$$

where the matrices are defined as:

$$\begin{aligned} (M_D)_{ij} &= \int_\Omega \boldsymbol{w}_i \cdot k^{-1} \boldsymbol{w}_j \\ (G_D)_{ij} &= \int_\Omega \boldsymbol{w}_i \nabla q_j \\ (\mathrm{h}_D)_{ij} &= \int_{\partial\Omega} q_i \boldsymbol{n} \cdot \boldsymbol{u}_S \end{aligned}$$

We can now Rewrite the two systems by splitting between velocities at the interior and those at the interface. For Stokes:

$$\begin{bmatrix} K_{SS} & K_{S\Gamma} & G_{S\Gamma} \\ K_{\Gamma S} & K_{\Gamma\Gamma} & G_{\Gamma S} \\ G_{S\Gamma}^T & G_{\Gamma S}^T & 0 \end{bmatrix} \begin{bmatrix} u_S \\ \lambda \\ p_S \end{bmatrix} = \begin{bmatrix} \mathrm{f}_{SS} \\ \mathrm{f}_{S\Gamma} \\ \mathrm{h}_S \end{bmatrix} \tag{84}$$

For Darcy:

$$\begin{bmatrix} M_{DD} & M_{D\Gamma} & G_{D\Gamma} \\ M_{\Gamma D} & M_{\Gamma\Gamma} & G_{\Gamma D} \\ G_{D\Gamma}^T & G_{\Gamma D}^T & 0 \end{bmatrix} \begin{bmatrix} u_D \\ \lambda \\ \varphi_D \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathrm{h}_D \end{bmatrix} \tag{85}$$

The we can combine the two systems into one:

$$\begin{bmatrix} K_{SS} & G_{S\Gamma} & K_{S\Gamma} & 0 & 0 \\ G_{S\Gamma}^T & 0 & K_{\Gamma S}^T & 0 & 0 \\ K_{\Gamma S} & G_{\Gamma S} & K_{\Gamma\Gamma}^{(S)} + M_{\Gamma\Gamma}^{(D)} & M_{\Gamma D} & G_{\Gamma D} \\ 0 & 0 & M_{D\Gamma} & M_{DD} & G_{D\Gamma} \\ 0 & 0 & G_{\Gamma D}^T & G_{D\Gamma}^T & 0 \end{bmatrix} \begin{bmatrix} u_S \\ p_S \\ \lambda \\ U_d \\ \varphi_D \end{bmatrix} = \begin{bmatrix} \mathrm{f}_{SS} \\ \mathrm{h}_S \\ \mathrm{f}_{S\Gamma} \\ 0 \\ \mathrm{h}_D \end{bmatrix} \tag{86}$$

We can now merge the degrees of freedom of velocity at the interior and pressure into a vectors x_S and x_D. The combined system becomes:

$$\begin{bmatrix} A_{II}^{(S)} & A_{I\Gamma}^{(S)} & 0 \\ A_{\Gamma I}^{(S)} & A_{\Gamma\Gamma} & A_{\Gamma I}^{(D)} \\ 0 & A_{I\Gamma}^{(D)} & A_{II}^{(D)} \end{bmatrix} \begin{bmatrix} x_I^{(S)} \\ \lambda \\ x_I^{(D)} \end{bmatrix} = \begin{bmatrix} \mathrm{b}_I^{(S)} \\ \mathrm{b}_\Gamma \\ \mathrm{b}_I^{(D)} \end{bmatrix} \tag{87}$$

Which is the form we're accustumed to. Skipping the steps this results in:

$$\left(A_{\Gamma\Gamma} - A_{\Gamma I}^{(S)}(A_{II}^{(S)})^{-1}A_{I\Gamma}^{(S)} - A_{\Gamma I}^{(D)}(A_{II}^{(D)})^{-1}A_{I\Gamma}^{(D)}\right)\lambda = \mathrm{b}_\Gamma - A_{\Gamma I}^{(S)}(A_{II}^{(S)})^{-1}\mathrm{b}_I^{(S)} - A_{\Gamma I}^{(D)}(A_{II}^{(D)})^{-1}\mathrm{b}_I^{(D)} \tag{88}$$

Which can be abreviated as:

$$S\lambda = G \tag{89}$$

Then we can split S into:

$$S_S = A_{\Gamma\Gamma}^{(S)} - A_{\Gamma I}^{(S)}(A_{II}^{(S)})^{-1}A_{I\Gamma}^{(S)} \tag{90}$$

$$S_D = A_{\Gamma\Gamma}^{(D)} - A_{\Gamma I}^{(D)}(A_{II}^{(D)})^{-1}A_{I\Gamma}^{(D)} \tag{91}$$

Since the problem statement has stated that $S_S(\lambda) = S_D(\lambda)$, we can conclude that G is 0:

$$G = \mathrm{b}_\Gamma - A_{\Gamma I}^{(S)}(A_{II}^{(S)})^{-1}\mathrm{b}_I^{(S)} - A_{\Gamma I}^{(D)}(A_{II}^{(D)})^{-1}\mathrm{b}_I^{(D)} = 0 \tag{92}$$

Part B

Write down the matrix form of a Dirichlet-Neumann iteration-by-subdomain using the matrices of the Darcy and the Stokes problems.

First we must impose the Neumann boundary condition on the Stokes equation. Using the same notation as in equation 84, the Stokes equation looks like the following:

$$\begin{bmatrix} K_{SS} & K_{S\Gamma} & G_{S\Gamma} \\ K_{\Gamma S} & K_{\Gamma\Gamma} & G_{\Gamma S} \\ G_{S\Gamma}^T & G_{\Gamma S}^T & 0 \end{bmatrix} \begin{bmatrix} u_S \\ \lambda \\ p_S \end{bmatrix} = \begin{bmatrix} \mathrm{f}_{SS} \\ \mathrm{f}_{S\Gamma} - M_{\Gamma\Gamma}^{(D)}\lambda^{k-1} - M_{\Gamma D}u_D^{k-1} - G_{D\Gamma}^T\varphi^{k-1} \\ \mathrm{h}_S \end{bmatrix} \tag{93}$$

For the Darcy equation we use the same notation as in equation 85, and we impose a Dirichlet boundary condition:

$$\begin{bmatrix} M_{DD} & G_{D\Gamma} \\ G_{D\Gamma}^T & 0 \end{bmatrix} \begin{bmatrix} u_D^k \\ \varphi^k \end{bmatrix} = \begin{bmatrix} -M_{\Gamma D}^T\lambda^l \\ \mathrm{h}_D \end{bmatrix} \tag{94}$$

where λ^l can be chosen in order to use a Jacobi scheme ($l = k-1$) or a Gauss-Seidel ($l = k$).

Part C

Identify the Richardson iteration for the algebraic problem in (a) resulting from (b).

The Neumann equation can be re-written so as to make u and p a function of λ:

$$\begin{bmatrix} K_{SS} & G_{S\Gamma} \\ G_{S\Gamma}^T & 0 \end{bmatrix} \begin{bmatrix} u_S^k \\ p_S^k \end{bmatrix} = \begin{bmatrix} \mathrm{f}_{SS} \\ \mathrm{h}_S \end{bmatrix} - \begin{bmatrix} K_{S\Gamma} \\ G_{\Gamma S}^T \end{bmatrix} \lambda^k \tag{95}$$

The Dirichlet equation was already a function of λ:

$$\begin{bmatrix} M_{DD} & G_{D\Gamma} \\ G_{D\Gamma}^T & 0 \end{bmatrix} \begin{bmatrix} u_D^k \\ \varphi^k \end{bmatrix} = \begin{bmatrix} 0 \\ \mathrm{h}_D \end{bmatrix} - \begin{bmatrix} M_{\Gamma D}^T \\ 0 \end{bmatrix} \lambda^l \tag{96}$$

Now from the second row of the Neumann equation we have:

$$K_{\Gamma S} u_S^k + K_{\Gamma\Gamma}\lambda^k + G_{\Gamma S} p_S^k = \mathrm{f}_{S\Gamma} - M_{\Gamma\Gamma}^{(D)}\lambda^{k-1} - M_{\Gamma D} u_D^{k-1} - G_{D\Gamma}^T \varphi^{k-1} \tag{97}$$

We can rewrite this as:

$$\begin{bmatrix} K_{\Gamma S} & G_{\Gamma S} \end{bmatrix} \begin{bmatrix} u_S^k \\ p_S^k \end{bmatrix} + K_{\Gamma\Gamma}\lambda^k = \mathrm{f}_{S\Gamma} - M_{\Gamma\Gamma}^{(D)}\lambda^{k-1} - \begin{bmatrix} M_{\Gamma D} & G_{D\Gamma}^T \end{bmatrix} \begin{bmatrix} u_D^{k-1} \\ \varphi^{k-1} \end{bmatrix} \tag{98}$$

Now these two vectors appeared in the previous two linear systems of equations. As such, we can replace them.

$$\begin{aligned} \begin{bmatrix} K_{\Gamma S} & G_{\Gamma S} \end{bmatrix} \begin{bmatrix} K_{SS} & G_{S\Gamma} \\ G_{S\Gamma}^T & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} \mathrm{f}_{SS} \\ \mathrm{h}_S \end{bmatrix} - \begin{bmatrix} K_{S\Gamma} \\ G_{\Gamma S}^T \end{bmatrix} \lambda^k \right) + K_{\Gamma\Gamma}\lambda^k \\ = \mathrm{f}_{S\Gamma} - M_{\Gamma\Gamma}^{(D)}\lambda^{k-1} - \begin{bmatrix} M_{\Gamma D} & G_{D\Gamma}^T \end{bmatrix} \begin{bmatrix} M_{DD} & G_{D\Gamma} \\ G_{D\Gamma}^T & 0 \end{bmatrix}^{-1} \left(\begin{bmatrix} 0 \\ \mathrm{h}_D \end{bmatrix} - \begin{bmatrix} M_{\Gamma D}^T \\ 0 \end{bmatrix} \lambda^l \right) \end{aligned}$$

At this point we must decide to go for Jacobi or Gauss-Seidel. I will go for Jacobi because of personal preference.

$$\begin{aligned} \left(K_{\Gamma\Gamma} - \begin{bmatrix} K_{\Gamma S} & G_{\Gamma S} \end{bmatrix} \begin{bmatrix} K_{SS} & G_{S\Gamma} \\ G_{S\Gamma}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} K_{S\Gamma} \\ G_{\Gamma S}^T \end{bmatrix} \right) \lambda^k = \left(\begin{bmatrix} M_{\Gamma D} & G_{D\Gamma}^T \end{bmatrix} \begin{bmatrix} M_{DD} & G_{D\Gamma} \\ G_{D\Gamma}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} M_{\Gamma D}^T \\ 0 \end{bmatrix} - M_{\Gamma\Gamma}^{(D)} \right) \lambda^{k-1} \\ + \left(\mathrm{f}_{S\Gamma} - \begin{bmatrix} K_{\Gamma S} & G_{\Gamma S} \end{bmatrix} \begin{bmatrix} K_{SS} & G_{S\Gamma} \\ G_{S\Gamma}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{f}_{SS} \\ \mathrm{h}_S \end{bmatrix} \right. \\ \left. - \begin{bmatrix} M_{\Gamma D} & G_{D\Gamma}^T \end{bmatrix} \begin{bmatrix} M_{DD} & G_{D\Gamma} \\ G_{D\Gamma}^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathrm{h}_D \end{bmatrix} \right) \end{aligned}$$

This very large system becomes much smaller if we compare the large parentheses with expressions 90, 91 and 92:

$$S_S \lambda^k = -S_D \lambda^{k-1} + G \tag{99}$$

Now we can manipulate it much more easily to obtain:

$$\lambda^k = \lambda^{k-1} + S_S^{-1}(G - S\lambda^{k-1}) \tag{100}$$

which is clearly a Richardson iteration with preconditioner S_S.

4 Monolithic and Partitioned Schemes in Time

Consider the one-dimensional, transient, heat transfer equation:

$$\begin{aligned} \frac{\partial u}{\partial t} - \kappa \frac{\partial^2 u}{\partial x^2} &= f \quad \textit{in } [0,1] \\ u(x=0,t) &= 0 \\ u(x=1,t) &= 0 \\ u(x,t=0) &= 0 \end{aligned}$$

4.1 FEM discretization

Discretize it using the finite element method (linear elements, element size h) for the discretization in space, and a BDF1 scheme for the discretization in time. Write down the weak form of the problem and the resulting matrix form of the problem, including the corresponding boundary integrals if necessary. Consider $\kappa = 1, f = 1, \delta t = 1$.

BDF1 time discretization is comonly known as backwards Euler. The expression for an arbitrary function y is:

$$y^{n+1} - y^n = \delta t \left.\frac{dy}{dt}\right|^{n+1} \tag{101}$$

Moving on to our problem in particular, the expression becomes:

$$u^{n+1} - u^n = \delta t\, f + \delta t\, \kappa \left.\frac{\partial^2 u}{\partial x^2}\right|^{n+1} \tag{102}$$

This can be rearrange and rewritten as:

$$-\delta t\, \kappa \left.\frac{\partial^2 u}{\partial x^2}\right|^{n+1} + u^{n+1} = u^n + \delta t\, f \tag{103}$$

We can multiply both sides by a test function $w \in \mathcal{H}_0^1(\Omega)$ and integrate on the domain:

$$-\delta t\, \kappa \int_\Omega w \left.\frac{\partial^2 u}{\partial x^2}\right|^{n+1} + \int_\Omega w u^{n+1} = \int_\Omega w u^n + \delta t \int_\Omega w f \tag{104}$$

Integrating by parts results in:

$$\delta t\, \kappa \int_\Omega \frac{\partial w}{\partial x} \left.\frac{\partial u}{\partial x}\right|^{n+1} + \int_\Omega w u^{n+1} = \int_\Omega w u^n + \delta t \int_\Omega w f + \delta t\, \kappa \left[w \left.\frac{\partial u}{\partial n}\right|^{n+1} \right]_{x=x_0}^{x=x_L} \tag{105}$$

The test function w has compact support and as such the last term vanishes. We can now use a Galerkin discretization:

$$N \approx w \qquad u \approx \sum_{j=2}^{n_{\text{nodes}}-1} N_j u_j \tag{106}$$

Notice that we are skipping the two nodes at the boundary since their value is 0. The result is the following:

$$\left(\delta t\, \kappa \int_\Omega \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \int_\Omega N_i N_j\right) u_j^{n+1} = \left(\int_\Omega N_i N_j\right) u_j^n + \delta t \int_\Omega N_i f \tag{107}$$

This can be rewritten as the following matrix system:

$$(\delta t\,\kappa K + M)U^{n+1} = MU^n + F \tag{108}$$

If we substitute the values:

$$(K + M)U^{n+1} = MU^n + F \tag{109}$$

4.2 Domain decomposition

Consider a domain decomposition approach for the previous problem. The left subdomain is composed of 2 elements $(h = 0.2)$*, while the right subdomain is composed of 3 elements (*$h = 0.2$*). Show that, if a monolithic approach is adopted, no boundary integrals are required at the interface. From now on, we denote the values at the nodes of the mesh as* $u_0, u_1, u_2, u_3, u_4, u_5$ *The interface is at* u_2

Let's first recover equation 105 and adapt it to our problem:

$$\left.\begin{aligned} \delta t\,\kappa \int_0^{0.4} \frac{\partial w}{\partial x}\,\frac{\partial u}{\partial x}\bigg|^{n+1} + \int_0^{0.4} wu^{n+1} = \int_0^{0.4} wu^n + \delta t \int_0^{0.4} wf + \delta t\,\kappa \left[w\,\frac{\partial u}{\partial n}\bigg|^{n+1} \right]_{x=0}^{x=0.4} \\ \delta t\,\kappa \int_{0.4}^{1} \frac{\partial w}{\partial x}\,\frac{\partial u}{\partial x}\bigg|^{n+1} + \int_{0.4}^{1} wu^{n+1} = \int_{0.4}^{1} wu^n + \delta t \int_{0.4}^{1} wf + \delta t\,\kappa \left[w\,\frac{\partial u}{\partial n}\bigg|^{n+1} \right]_{x=0.4}^{x=1} \end{aligned}\right\} \tag{110}$$

The values at $x = 0$ and $x = 1$ the boundary values vanish just as before:

$$\left.\begin{aligned} \delta t\,\kappa \int_0^{0.4} \frac{\partial w}{\partial x}\,\frac{\partial u}{\partial x}\bigg|^{n+1} + \int_0^{0.4} wu^{n+1} = \int_0^{0.4} wu^n + \delta t \int_0^{0.4} wf + \delta t\,\kappa w\,\frac{\partial u}{\partial n}\bigg|_{x=0.4}^{n+1} \\ \delta t\,\kappa \int_{0.4}^{1} \frac{\partial w}{\partial x}\,\frac{\partial u}{\partial x}\bigg|^{n+1} + \int_{0.4}^{1} wu^{n+1} = \int_{0.4}^{1} wu^n + \delta t \int_{0.4}^{1} wf - \delta t\,\kappa w\,\frac{\partial u}{\partial n}\bigg|_{x=0.4}^{n+1} \end{aligned}\right\} \tag{111}$$

The matrix system looks like:

$$\left.\begin{aligned} (K^{(1)} + M^{(1)})(U^{(1)})^{n+1} = M^{(1)}(U^{(1)})^n + F^{(1)} + Q^{(1)} \\ (K^{(2)} + M^{(2)})(U^{(2)})^{n+1} = M^{(2)}(U^{(2)})^n + F^{(2)} - Q^{(2)} \end{aligned}\right\} \tag{112}$$

The new vector Q is the result of the interface value. Notice that $Q^{(1)} = Q^{(2)}$ as can be easily seen in equation 111. If we expand the matrices we obtain the following systems. For system 1 (left part of the domain):

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} \\ A_{21}^{(1)} & A_{22}^{(1)} \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \end{bmatrix} = \begin{bmatrix} F_1^{(1)} \\ F_2^{(1)} + Q^{(1)} \end{bmatrix} + \begin{bmatrix} M_{11}^{(1)} & M_{12}^{(1)} \\ M_{21}^{(1)} & M_{22}^{(1)} \end{bmatrix} \begin{bmatrix} u_1^{n} \\ u_2^{n} \end{bmatrix} \tag{113}$$

And system 2:

$$\begin{bmatrix} A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ A_{21}^{(2)} & A_{22}^{(2)} & A_{23}^{(2)} \\ A_{31}^{(2)} & A_{32}^{(2)} & A_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \end{bmatrix} = \begin{bmatrix} F_1^{(2)} - Q^{(2)} \\ F_2^{(2)} \\ F_3^{(2)} \end{bmatrix} + \begin{bmatrix} M_{11}^{(2)} & M_{12}^{(2)} & M_{13}^{(2)} \\ M_{21}^{(2)} & M_{22}^{(2)} & M_{23}^{(2)} \\ M_{31}^{(2)} & M_{32}^{(2)} & M_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_2^{n} \\ u_3^{n} \\ u_4^{n} \end{bmatrix} \tag{114}$$

Note that the notes at the boundary (u_0 and u_5) are not present because they have been eliminated thanks to their Dirichlet boundary condition. Combining both systems in a monolithic manner results in:

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & 0 & 0 \\ A_{21}^{(1)} & A_{22}^{(1)} + A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ 0 & A_{21}^{(2)} & A_{22}^{(2)} & A_{23}^{(2)} \\ 0 & A_{31}^{(2)} & A_{32}^{(2)} & A_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \end{bmatrix} =$$
$$\begin{bmatrix} F_1^{(1)} \\ F_2^{(1)} + F_1^{(2)} + Q^{(1)} - Q^{(2)} \\ F_2^{(2)} \\ F_3^{(2)} \end{bmatrix} + \begin{bmatrix} M_{11}^{(1)} & M_{12}^{(1)} & 0 & 0 \\ M_{21}^{(1)} & M_{22}^{(1)} + M_{11}^{(2)} & M_{12}^{(2)} & M_{13}^{(2)} \\ 0 & M_{21}^{(2)} & M_{22}^{(2)} & M_{23}^{(2)} \\ 0 & M_{31}^{(2)} & M_{32}^{(2)} & M_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \end{bmatrix}$$

Now we see how $Q^{(1)}$ and $Q^{(2)}$ are subtracting each other, which, as observed earlier, are equal. Then the system becomes:

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & 0 & 0 \\ A_{21}^{(1)} & A_{22}^{(1)} + A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ 0 & A_{21}^{(2)} & A_{22}^{(2)} & A_{23}^{(2)} \\ 0 & A_{31}^{(2)} & A_{32}^{(2)} & A_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \end{bmatrix} = \begin{bmatrix} F_1^{(1)} \\ F_2^{(1)} + F_1^{(2)} \\ F_2^{(2)} \\ F_3^{(2)} \end{bmatrix} + \begin{bmatrix} M_{11}^{(1)} & M_{12}^{(1)} & 0 & 0 \\ M_{21}^{(1)} & M_{22}^{(1)} + M_{11}^{(2)} & M_{12}^{(2)} & M_{13}^{(2)} \\ 0 & M_{21}^{(2)} & M_{22}^{(2)} & M_{23}^{(2)} \\ 0 & M_{31}^{(2)} & M_{32}^{(2)} & M_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \end{bmatrix} \tag{115}$$

As it can be seen, all the matrices are domain integrals and none are on the boundary.

□

4.3 Dirichlet-to-Neumann operator

Obtain the algebraic form of the Dirichlet-to-Neumann operator (Steklov-Poincaré's operator) for the left subdomain, departing from given values of u_i^n at time step n, and an interface value u_2^{n+1}

Let's start with equation 113:

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} \\ A_{21}^{(1)} & A_{22}^{(1)} \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \end{bmatrix} = \begin{bmatrix} F_1^{(1)} \\ F_2^{(1)} + Q^{(1)} \end{bmatrix} + \begin{bmatrix} M_{11}^{(1)} & M_{12}^{(1)} \\ M_{21}^{(1)} & M_{22}^{(1)} \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \end{bmatrix}$$

Now we can rearrange this as:

$$A_{11}^{(1)} u_1^{n+1} = \left(F_1^{(1)} + M_{11}^{(1)} u_1^n + M_{12}^{(1)} u_2^n \right) - A_{12}^{(1)} u_2^{n+1} \tag{116}$$

Here we see the system with the Dirichlet operator (last term on the RHS).

4.4 Neumann-to-Dirichlet operator

Obtain the algebraic form of the Neumann-to-Dirichlet operator for the right subdomain, departing from given values of u_i^n and an interface value for the fluxes $\phi^{n+1} = \kappa \partial_x u^{n+1}$ at the coordinate of node 2

Let's start off in equation 114:

$$\begin{bmatrix} A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ A_{21}^{(2)} & A_{22}^{(2)} & A_{23}^{(2)} \\ A_{31}^{(2)} & A_{32}^{(2)} & A_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \end{bmatrix} = \begin{bmatrix} F_1^{(2)} - Q^{(2)} \\ F_2^{(2)} \\ F_3^{(2)} \end{bmatrix} + \begin{bmatrix} M_{11}^{(2)} & M_{12}^{(2)} & M_{13}^{(2)} \\ M_{21}^{(2)} & M_{22}^{(2)} & M_{23}^{(2)} \\ M_{31}^{(2)} & M_{32}^{(2)} & M_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_2^n \\ u_3^n \\ u_4^n \end{bmatrix} \tag{117}$$

We will now park this expression here and work in system 1. Notice that in the previous section we ignored the second row of the matrix system:

$$A_{22}^{(1)} u_2^{n+1} = F_2^{(1)} + Q^{(1)} + M_{21}^{(1)} u_1^n + M_{22}^{(1)} u_2^n - A_{21}^{(1)} u_1^{n+1}$$

We can rearrange this as:

$$Q^{(1)} = A_{22}^{(1)} u_2^{n+1} + A_{21}^{(1)} u_1^{n+1} - F_2^{(1)} - M_{21}^{(1)} u_1^n - M_{22}^{(1)} u_2^n \tag{118}$$

We must now remember that:

$$Q^{(1)} = Q^{(2)} = \phi^{n+1} = \kappa \partial_x u^{n+1} \tag{119}$$

Hence equation 117 becomes:

$$\begin{aligned} \begin{bmatrix} A_{11}^{(2)} & A_{12}^{(2)} & A_{13}^{(2)} \\ A_{21}^{(2)} & A_{22}^{(2)} & A_{23}^{(2)} \\ A_{31}^{(2)} & A_{32}^{(2)} & A_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \end{bmatrix} &= \begin{bmatrix} F_2^{(1)} + F_1^{(2)} \\ F_2^{(2)} \\ F_3^{(2)} \end{bmatrix} \\ &+ \begin{bmatrix} M_{21}^{(1)} & M_{11}^{(2)} + M_{22}^{(1)} & M_{12}^{(2)} & M_{13}^{(2)} \\ 0 & M_{21}^{(2)} & M_{22}^{(2)} & M_{23}^{(2)} \\ 0 & M_{31}^{(2)} & M_{32}^{(2)} & M_{33}^{(2)} \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \end{bmatrix} \\ &- \begin{bmatrix} A_{22}^{(1)} u_2^{n+1} + A_{21}^{(1)} u_1^{n+1} \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{120}$$

where this last matrix is the Neumann operator. One can compare with the monolithic system in equation 115 to see that the rest of the system is unaltered here.

4.5 Staggered iteration scheme

Write down the iterative algorithm for a staggered approach applying Dirichlet boundary conditions at the interface to the left subdomain and Neumann boundary conditions at the interface for the right subdomain.

Let's start with our two equations 116 and 120. They can be simplified with the help of notation to:

$$\left.\begin{aligned} A_{II}^{(1)} u_{I_1}^{n+1} &= B_I^{(1)} - A_{I\Gamma}^{(1)} u_\Gamma^{n+1} \\ \begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} u_{I_2}^{n+1} \\ u_\Gamma^{n+1} \end{bmatrix} &= \begin{bmatrix} B_I^{(2)} \\ B_\Gamma \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} u_{I_1}^{n+1} \\ u_\Gamma^{n+1} \end{bmatrix} \end{aligned}\right\} \tag{121}$$

where I refers to interior nodes and Γ to the interface node. The issue presented here is the circular logical dependency: they both need each other's solution. Hence we'll follow an iterative algorithm with predictive variables $\tilde{u}$:

$$\left.\begin{aligned} A_{II}^{(1)} u_{I_1}^{n+1,k} &= B_I^{(1)} - A_{I\Gamma}^{(1)} \tilde{u}_\Gamma^{n+1,k} \\ \begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} u_{I_2}^{n+1,k} \\ u_\Gamma^{n+1,k} \end{bmatrix} &= \begin{bmatrix} B_I^{(2)} \\ B_\Gamma \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} \tilde{u}_{I_1}^{n+1,k-1} \\ \tilde{u}_\Gamma^{n+1,k-1} \end{bmatrix} \end{aligned}\right\} \tag{122}$$

where k is the iterator counter. A good starter for $k = 0$ is $\tilde{u}^{n+1,0} = u^n$. Convergence is reached when $u \to \tilde{u}$ which yields the same result as the monolithic scheme. Neither convergence nor stability are guaranteed.

4.6 Substitution and iteration by subdomain

Do the same for a substitution and an iteration by subdomains scheme.

The substitution scheme is quite straight-forward:

$$\left.\begin{aligned} A_{II}^{(1)} u_{I_1}^{n+1,k} &= B_I^{(1)} - A_{I\Gamma}^{(1)} \tilde{u}_\Gamma^{n+1,k-1} \\ \begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} u_{I_2}^{n+1,k} \\ u_\Gamma^{n+1,k} \end{bmatrix} &= \begin{bmatrix} B_I^{(2)} \\ B_\Gamma \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u_{I_1}^{n+1,k}} \\ \boldsymbol{u_\Gamma^{n+1,k}} \end{bmatrix} \end{aligned}\right\} \tag{123}$$

The difference has been highlighted with boldface. As it can be seen, this time only subdomain 1 uses a predictive variable.

Iteration by subdomain is not too different either:

$$\left.\begin{aligned} A_{II}^{(1)} u_{I_1}^{n+1,k} &= B_I^{(1)} - A_{I\Gamma}^{(1)} \tilde{u}_\Gamma^{n+1,\boldsymbol{\ell}} \\ \begin{bmatrix} A_{II}^{(2)} & A_{I\Gamma}^{(2)} \\ A_{\Gamma I}^{(2)} & A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} u_{I_2}^{n+1,k} \\ u_\Gamma^{n+1,k} \end{bmatrix} &= \begin{bmatrix} B_I^{(2)} \\ B_\Gamma \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ A_{\Gamma I}^{(1)} & A_{\Gamma\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} \tilde{u}_{I_1}^{n+1,\boldsymbol{k-1}} \\ \tilde{u}_\Gamma^{n+1,\boldsymbol{k-1}} \end{bmatrix} \end{aligned}\right\} \tag{124}$$

The changes with respect to the staggered iterative scheme are highlited in boldface again. As is a recurring theme already, $\ell = k-1$ is for Jacobi and $\ell = k$ for Gauss-Seidel iterative scheme.

4.7 Nitsche method for boundary conditons

Rewrite the algebraic system associated to the left subdomain (Dirichlet boundary conditions at the interface), using Nitsche's method for applying the boundary conditions. How does the condition number of the resulting system of equations vary with the penalty parameter α ?

The weak form of our problem in subdomain 1 looks like:

$$\big(w, u_t\big)_{\Omega_1} + \big(w_x, \kappa u_x\big)_{\Omega_1} - \big\langle w, n_1 \kappa u_x \big\rangle_{\partial\Omega_1} = \big(w, f\big)_{\Omega_1} \tag{125}$$

where $w \in \mathcal{V}_h$. We can now add Nitsche's terms:

$$\begin{aligned} \big(w, u_t\big)_{\Omega_1} + \kappa\big(w_x, u_x\big)_{\Omega_1} - \kappa\big\langle w, n_1 u_x\big\rangle_{\partial\Omega_1} + \alpha\frac{\kappa}{h}\big(w, u\big)_{\partial\Omega_1} - \kappa\big\langle n_1 w_x, u\big\rangle_{\partial\Omega_1} \\ = \alpha\frac{\kappa}{h}\big(w, \bar{u}\big)_{\partial\Omega_1} - \kappa\big\langle n_1 w_x, \bar{u}\big\rangle_{\partial\Omega_1} + \big(w, f\big)_{\Omega_1} \end{aligned} \tag{126}$$

Here $\bar{u}$ is the weakly enforced dirichlet condition, h is the element size and α is the penalty parameter. We can now substitute in $\partial\Omega_1$:

$$\begin{aligned} \big(w, u_t\big)_{\Omega_1} + \kappa\big(w_x, u_x\big)_{\Omega_1} - \kappa\Big[w n_1 u_x\Big]_0^{0.4} + \alpha\frac{\kappa}{h}\Big[wu\Big]_0^{0.4} - \kappa\Big[n_1 w_x u\Big]_0^{0.4} \\ = \alpha\frac{\kappa}{h}\Big[w\bar{u}\Big]_0^{0.4} - \kappa\Big[n_1 w_x \bar{u}\Big]_0^{0.4} + \big(w, f\big)_{\Omega_1} \end{aligned} \tag{127}$$

Finite element matrices K and M have already been defined, as well as vectors U and F. Let's now define the following:

$$B_i = \partial_x N_i \tag{128}$$

and of course N is simply the shape functions arranged in a column vector. Substituting $\kappa = 1$, $h = 0.2$ and $B_2 = -B_0 = \frac{h}{2} = \frac{1}{10}$ the space- and time-discretized system becomes:

$$\frac{1}{\delta t}M(U^{n+1} - U^n) + KU^{n+1} + \frac{1}{10}N(U_2^{n+1} + U_0^{n+1}) + 5\alpha N(U_2^{n+1} - U_0^{n+1}) - B(U_2^{n+1} + U_0^{n+1}) \\ = 5\alpha N(\bar{U}_2^{n+1} - \bar{U}_0^{n+1}) - B(\bar{U}_2^{n+1} + \bar{U}_0^{n+1}) + F$$

This can be simplified down to:

$$(\frac{1}{\delta t}M + K)U^{n+1} + (5.1N - B)U_2^{n+1} - (4.9N + B)U_0^{n+1} \\ = (5\alpha N - B)\bar{U}_2^{n+1} - (5\alpha N + B)\bar{U}_0^{n+1} + \frac{1}{\delta t}MU^n + F \tag{129}$$

In our case we had that $\bar{U}_0 = 0$:

$$(\frac{1}{\delta t}M + K)U^{n+1} + (5.1N - B)U_2^{n+1} = (5\alpha N - B)\bar{U}_2^{n+1} + \frac{1}{\delta t}MU^n + F \tag{130}$$

Therefore our system's matrix is:

$$A = \begin{bmatrix} M_{00}^{(1)} + K_{00}^{(1)} & M_{01}^{(1)} + K_{01}^{(1)} & M_{02}^{(1)} + K_{02}^{(1)} + 5.1\alpha N_0 - B_0 \\ M_{10}^{(1)} + K_{10}^{(1)} & M_{21}^{(1)} + K_{21}^{(1)} & M_{12}^{(1)} + K_{12}^{(1)} + 5.1\alpha N_1 - B_1 \\ M_{20}^{(1)} + K_{20}^{(1)} & M_{21}^{(1)} + K_{21}^{(1)} & M_{22}^{(1)} + K_{22}^{(1)} + 5.1\alpha N_2 - B_2 \end{bmatrix} \tag{131}$$

As we increase the value of alpha the condition number of A is gona increase and hence the system is going to become more ill-condition. The goal therefore is to find the smallest α that stabilizes the system. For the Poisson problem, $\alpha > 2c_i$ where c_i is a parameter dependent on element shapes. For non-streched elements it grows at a rate of $\mathcal{O}(1)$.

We can compare it to the penalty method's $\alpha > c_i \frac{\kappa}{h}$, which grows at a rate of $\mathcal{O}(n^d)$ for d spatial dimensions. It is clear that Nitsche's method is better since it allows for smaller values of α and therefore it can stabilize much larger systems.

5 Operator Splitting Techniques

Consider the one dimensional, transient, convection-diffusion equation:

$$\begin{aligned} \frac{\partial u}{\partial t} - \kappa \frac{\partial^2 u}{\partial x^2} + a_x \frac{\partial u}{\partial x} = f \quad & \textit{in } [0,1] \\ u(x=0,t) = 0 \\ u(x=1,t) = 0 \\ u(x,t=0) = 0 \end{aligned}$$

with $\kappa = 1, a_x = 1, f = 1$

5.1 FEM discretization

Discretize it in space using finite elements (3 elements) and in time (finite differences, BDF1). Solve the first step of the problem, writing the solution as a function of the time step size δt

Let's start re-writing the equation in more compact notation and substituting the values:

$$u_t - u_{xx} + u_x = 1 \tag{132}$$

We can multiply it all with $w \in \mathcal{H}_0^1(\Omega)$ and integrate over the domain:

$$\int_\Omega w u_t - \int_\Omega w u_{xx} + \int_\Omega w u_x = \int_\Omega w \tag{133}$$

Using integration by parts yields:

$$\int_\Omega w u_t + \int_\Omega w_x u_x + \int_\Omega w u_x = \int_\Omega w + \int_\Gamma w u_x n \tag{134}$$

Because of the compact support the flux term vanishes.

$$\int_\Omega w u_t + \int_\Omega w_x u_x + \int_\Omega w u_x = \int_\Omega w \tag{135}$$

We now interpolate with shape functions.

$$\left(\int_\Omega N_i N_j\right)(U_t)_j + \left(\int_\Omega (N_x)_i (N_j)_x\right) U_j + \left(\int_\Omega N_i (N_j)_x\right) U_j = \left(\int_\Omega N_i\right) \tag{136}$$

The terms in parentheses can be replaced by matrices:

$$M U_t + (K + C) U = F \tag{137}$$

The formula for BDF1 time discretization is:

$$y_t^{n+1} \approx \frac{y^{n+1} - y^{n+1}}{\delta t} \tag{138}$$

Therefore our discretized equation becomes:

$$M(U^{n+1} - U^n) + \delta t\,(K + C) U^{n+1} = \delta t\, F$$

And finally:

$$\left[\frac{1}{\delta t}M + K + C\right] U^{n+1} = F + \frac{1}{\delta t} M U^n \tag{139}$$

Let's now obtain the numerical values:

$$M = \frac{1}{18}\begin{bmatrix}2&1&0&0\\1&4&1&0\\0&1&4&1\\0&0&1&2\end{bmatrix} \quad K = 3\begin{bmatrix}1&-1&0&0\\-1&2&-1&0\\0&-1&2&-1\\0&0&-1&1\end{bmatrix} \quad C = \frac{1}{2}\begin{bmatrix}1&1&0&0\\-1&0&1&0\\0&-1&0&1\\0&0&-1&-1\end{bmatrix} \quad F = \frac{1}{6}\begin{bmatrix}1\\2\\2\\1\end{bmatrix} \tag{140}$$

Since nodes 0 and 3 have Dirichlet BC with value 0 we can write the system as:

$$\begin{bmatrix}\frac{2}{9\delta t}+8 & \frac{1}{18\delta t}-2.5\\ \frac{1}{18\delta t}-3.5 & \frac{2}{9\delta t}+8\end{bmatrix}\begin{bmatrix}U_1^{n+1}\\U_2^{n+1}\end{bmatrix} = \begin{bmatrix}\frac{1}{3}\\\frac{1}{3}\end{bmatrix} + \begin{bmatrix}\frac{2}{9\delta t} & \frac{1}{18\delta t}\\ \frac{1}{18\delta t} & \frac{2}{9\delta t}\end{bmatrix}\begin{bmatrix}U_1^{n}\\U_2^{n}\end{bmatrix}$$

The result is:

$$U^{n+1} = \frac{6\delta t}{2943\delta t^{\,2} + 324\delta t + 5}\begin{bmatrix}0\\51\delta t + 1\\57\delta t + 1\\0\end{bmatrix} \tag{141}$$

These numerical values have been obtained with a matlab script that can be found in appendix A.1.

5.2 Operator splitting technique

Solve the same time step by using a first order operator splitting technique.

The operator splitting technique consists of splitting the system in two:

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_\nu$$

These can be defined as:

$$\mathcal{L}_a = \boldsymbol{a}\cdot\nabla$$
$$\mathcal{L}_\nu = -\nabla\cdot\kappa\nabla$$

For our problem in particular this becomes:

$$\mathcal{L}_a = \partial_x \tag{142}$$
$$\mathcal{L}_\nu = -\partial_{xx} \tag{143}$$

The transient convection-diffusion equation becomes:

$$u_t + \mathcal{L}_a u + \mathcal{L}_\nu u = 1 \tag{144}$$

The operator splitting technique then mandates first solving the convective therm:

$$\begin{aligned} u_a^n &= u^n \\ (u_a)_t + \mathcal{L}_a u_a &= 0 \end{aligned} \tag{145}$$

Once this has been solved we solve:

$$\begin{aligned} u_\nu^n &= u_a^{n+1} \\ (u_\nu)_t + \mathcal{L}_\nu u_\nu &= 1 \end{aligned} \tag{146}$$

And the final result is:

$$u^{n+1} = u_\nu \tag{147}$$

Let's now solve our problem. First we obtain U_a^0:

$$U_a^0 = U^0 = 0 \tag{148}$$

Hence next expression looks like:

$$\left[\frac{1}{\delta t}M + C\right] U_a^1 = \frac{1}{\delta t} M U_a^0 = 0 \tag{149}$$

Expanding it:

$$\begin{bmatrix} \frac{2}{9\delta t} & \frac{1}{18\delta t} + \frac{1}{2} \\ \frac{1}{18\delta t} - \frac{1}{2} & \frac{2}{9\delta t} \end{bmatrix} \begin{bmatrix} U_{a1}^1 \\ U_{a2}^1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The result is trivial given the empty RHS:

$$\begin{bmatrix} U_{a1}^1 \\ U_{a2}^1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{150}$$

Now moving on to the next step:

$$U_\nu^0 = U_a^1 = 0 \tag{151}$$

Therefore the step looks like:

$$\left[\frac{1}{\delta t}M + K\right] U_\nu^1 = F + \frac{1}{\delta t} M U_\nu^0 = F \tag{152}$$

Expanding it:

$$\begin{bmatrix} \frac{2}{9\delta t} + 6 & \frac{1}{18} - 3 \\ \frac{1}{18\delta t} - 3 & \frac{2}{9\delta t} + 6 \end{bmatrix} \begin{bmatrix} U_{\nu 1}^1 \\ U_{\nu 2}^1 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \end{bmatrix}$$

We obtain as result:

$$U^{n+1} = \frac{6\delta t}{54\delta t + 5} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{153}$$

This is clearly different from equation 141, the result from the monolithic scheme. We do see that, as $\delta t \to 0$, both results approach:

$$U^{n+1} = \frac{6\delta t}{5} \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{154}$$

These numerical values have also been obtained with the matlab script in appendix A.1.

5.3 Splitting error

Evaluate the error of the splitting approach with respect to the monolithic approach. Plot the splitting error vs. the time step size for $\delta t = 1, \delta t = 0.5, \delta t = 0.25$. Comment on the results.

The error can be computed as the norm of:

$$E = ||U_{monolithic} - U_{split}||_{\mathcal{L}^2} \tag{155}$$

The result is:

δ_t	Norm of error
0.25	5.937×10^{-3}
0.50	7.118×10^{-3}
1.00	7.869×10^{-3}

Table 1: Caption

This can be graphed and the result is the following:

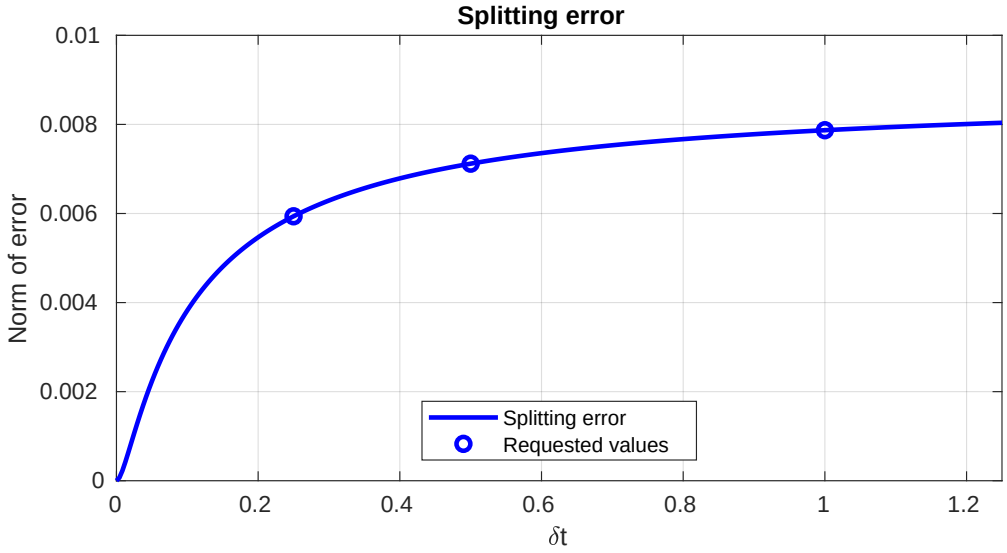


Figure 1: Splitting error as a function of δt. Requested values are $\delta t = 0.25$, 0.5 and 1.

It is clear from the graph that the error decreases with δt. On the other hand, it is clear that the error approaches a maximum value assymptoticaly. For instance at $\delta t = 30$ the error is still only 8.736×10^{-3}, well within the same order of magnitude as the previous results. The error does grow with the number of elements however ($E = 1.697 \times 10^{-2}$ with 12 elements and $\delta t = 0.5$), so it is something one must be careful with.

6 Fractional Step Methods

Consider the fractional step approach for the incompressible Navier-Stokes equations (Yosida scheme):

$$M\frac{1}{\delta t}\left(\hat{U}^{n+1}-U^{n}\right)+K\hat{U}^{n+1}=f-G\tilde{P}^{n+1} \tag{156}$$

$$DM^{-1}GP^{n+1}=\frac{1}{\delta t}D\hat{U}^{n+1}-DM^{-1}G\tilde{P}^{n+1} \tag{157}$$

$$M\frac{1}{\delta t}\left(U^{n+1}-\hat{U}^{n+1}\right)+\alpha K\left(U^{n+1}-\hat{U}^{n+1}\right)+G\left(P^{n+1}-\tilde{P}^{n+1}\right)=0 \tag{158}$$

6.1 Optimal alpha

Which is the optimal value for the α parameter?

Let's start by discretizing the Navier-Stokes momentum equation:

$$\frac{\partial u}{\partial t}+(u\cdot\nabla)u-\nu\Delta u+\nabla p=f$$

After FEM discretization:

$$MU_t+KU+GP=F$$

After using backwards Euler this becomes:

$$\frac{1}{\delta t}M(U^{n+1}-U^{n})KU^{n+1}+GP^{n+1}=F \tag{159}$$

Now if we add equations 156 and 158 we obtain:

$$\frac{1}{\delta t}M(U^{n+1}-U^{n})+K\Big((1-\alpha)\hat{U}^{n+1}+\alpha U^{n+1}\Big)+GP^{n+1}=F \tag{160}$$

It is clear that equations 159 and 160 will only be equal if:

$$\alpha=1$$

6.2 Error

What is the source of error of the scheme?

The source of the error is due to the splitting of the system. We first calculate $\hat{U}^{n+1}$ based not on pressure but on a guess on what the pressure will be ($\tilde{P}^{n+1}$). The we coorect the pressure with this already incorrect $\hat{U}$ to obtain a slightly incorrect P^{n+1}. Then we use this to obtain U^{n+1}. The issue is therefore due to the fact that $\tilde{P}$ is not the real pressure and therefore all that follows will be just as incorrect.

The only way to solve this system is iterating until convergence, however this scheme offers a good middle ground between accurate computational expense and less accurate but faster solving times.

7 ALE Formulations

7.1 Introduction

Given the spatial description of a property

$$\gamma(x, y, z, t) = \left[2x, ye^t, z\right]$$

the equations of movement:

$$\begin{aligned} x &= Xe^t \\ y &= Y + e^t - 1 \\ z &= Z \end{aligned}$$

and the equations of the movement of the mesh:

$$\begin{aligned} x_m &= \mathcal{X} + \alpha t \\ y_m &= \mathcal{Y} - \beta t \\ z_m &= \mathcal{Z} \end{aligned}$$

Part A

Obtain the description of the property in terms of the ALE coordinates $(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$

Our goal is to find $\gamma_{ALE}(\boldsymbol{\mathcal{X}}, t)$. We simply have to replace in the original equations to obtain:

$$\gamma_{ALE}(\boldsymbol{\mathcal{X}}, t) = \left[2\mathcal{X} + 2\alpha t, (\mathcal{Y} - \beta t)e^t, \mathcal{Z}\right] \tag{161}$$

Part B

Compute the velocity of the particles and the mesh velocity.

The velocity of a particle can be obtained via

$$\boldsymbol{v} = \frac{\partial \boldsymbol{x}(\boldsymbol{X}, t)}{\partial t} \tag{162}$$

For particles of the domain the result is:

$$\boldsymbol{v} = \begin{bmatrix} Xe^t \\ e^t \\ 0 \end{bmatrix} \tag{163}$$

And the mesh moves at:

$$\boldsymbol{v}_m = \begin{bmatrix} \alpha \\ -\beta \\ 0 \end{bmatrix} \tag{164}$$

Part C

Compute the ALE description of the material temporal derivative of γ.

The material derivative in ALE formulation looks like:

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma_{ALE}(\boldsymbol{\mathcal{X}}(\boldsymbol{X},t),t) = \frac{\partial \gamma_{ALE}(\boldsymbol{\mathcal{X}},t)}{\partial t} + (\boldsymbol{v}-\boldsymbol{v}_m)\cdot\nabla\gamma(\boldsymbol{x},t) \tag{165}$$

Expanding this results in:

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma_{ALE}(\boldsymbol{\mathcal{X}}(\boldsymbol{X},t),t) = \begin{bmatrix} 2\alpha \\ (\mathcal{Y}-\beta(t+1))e^t \\ 0 \end{bmatrix} + \begin{bmatrix} Xe^t-\alpha & e^t+\beta & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & e^t & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then solving results in:

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma_{ALE}(\boldsymbol{\mathcal{X}}(\boldsymbol{X},t),t) = \begin{bmatrix} 2Xe^t \\ (\mathcal{Y}-\beta t)e^t + e^{2t} \\ 0 \end{bmatrix} \tag{166}$$

We must now use the appropiate set of coordinates. Recalling that $\mathcal{X}+\alpha t = Xe^t$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma_{ALE}(\boldsymbol{\mathcal{X}},t) = \begin{bmatrix} 2\mathcal{X}+2\alpha t \\ (\mathcal{Y}-\beta t)e^t + e^{2t} \\ 0 \end{bmatrix} \tag{167}$$

7.2 Navier-Stokes

Write down the ALE form of the incompressible Navier-Stokes equations. Where (in time and space) is each of the terms of the equation evaluated? How are temporal derivatives computed?

Let's start with mass conservation:

$$\frac{\partial \rho_{ALE}}{\partial t} + \boldsymbol{c} \cdot \nabla \rho(\boldsymbol{x}, t) = -\rho \nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t) \tag{168}$$

Since the fluid is incompressible, all derivatives of density are 0, and the resulting equation is:

$$\nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t) = 0 \tag{169}$$

This is the same result as with spatial formulation. Moving on to the momentum equation:

$$\frac{\partial \boldsymbol{u}_{ALE}(\boldsymbol{\mathcal{X}}, t)}{\partial t} + \boldsymbol{c} \cdot \nabla \boldsymbol{u}(\boldsymbol{x}, t) = \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}, t) + \rho(\boldsymbol{x}, t)\boldsymbol{b}(\boldsymbol{x}, t) \tag{170}$$

Since the fluid is incompressible we know the density to be constant:

$$\frac{\partial \boldsymbol{u}_{ALE}(\boldsymbol{\mathcal{X}}, t)}{\partial t} + \boldsymbol{c} \cdot \nabla \boldsymbol{u}(\boldsymbol{x}, t) = \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{x}, t) + \rho\boldsymbol{b}(\boldsymbol{x}, t)$$

The Navier-Stokes equations describe Newtonian fluids, hence we must make the following substitution:

$$\boldsymbol{\sigma}(\boldsymbol{x}, t) = -p(\boldsymbol{x}, t)\boldsymbol{I} + 2\mu \nabla^s \boldsymbol{u}(\boldsymbol{x}, t) \tag{171}$$

Leading to our final result:

$$\left.\begin{aligned} \frac{\partial \boldsymbol{u}_{ALE}(\boldsymbol{\mathcal{X}}, t)}{\partial t} + \boldsymbol{c} \cdot \nabla \boldsymbol{u}(\boldsymbol{x}, t) - \mu \Delta u + \nabla p(\boldsymbol{x}, t) = \rho \boldsymbol{b}(\boldsymbol{x}, t) \\ \nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t) = 0 \end{aligned}\right\} \tag{172}$$

The energy equation is redundant and therefore not necessary when solving for incompressible Newtonian fluids.

7.3 Bibliographical search

Do a bibliographical research on existing methods for the definition of the mesh movement in ALE formulations (Poisson problem, Elasticity problem, etc.). Describe the main advantages of each of these methods.
There are two ways to guide mesh movement. These are mesh regularization and adaption. The former consists int moving the mesh in such a way that we avoid it from being too distorted. To do so, we need information about displacements of the underlying material with which to calculate the displacement of the mesh. There are multiple ways of accomplishing this.

The easiest case is when the displacements are prescribed, then they are all known a priori and the mesh can be moved accordingly. This is, however, not common, and when displacement is not known beforehand, than mesh displacement has to be computed alongside the material displacement. This is specially important near interfaces, where fluids are generally treated in an Eulerian manner whereas solids are described in a Lagrangian fashion.

The other devised strategy, mesh adaption, consists in optimizing the mesh density such that spots where more

more accuracy is required can be allocated the resources that would be wasted on less-demanding regions of the domain. The need for more or less accuracy must be measured and this is often done by solving an elliptic or parabolic differential equation on the domain.

Sources

1. HUERTA AND LIU (1988) - Viscous flow structure interaction. Journal of pressure vessel technology.
2. RODRIGUEZ-FERRAN (2002) - Arbitrary Lagrangian-Eulerian (ALE) formulation for hyperelasticity - International Journal for Numerical methods in engineering.

8 Fluid-Structure Interaction

8.1 Added mass effect

Describe the added mass effect problem for fluid structure interaction problems. When does it appear, what kind of problems suffer from it? What are the main methods for dealing with it?

The issue of added mas effect appears in fluid-structure interaction problems solved with iterative methods and coupled via a Neumann-Neumann cheme. It is problematic when the density of the solid is similar or lesser that that of the fluid.

Most engineering problems are therefore exempt from this problem since structures are generally made of steel, concrete, or other dense materials interacting with water or air. Fields like biology or bio-engineering are more prone to this problem since biological tissue is generally of similar density to water.

There are different ways of combating this problem, such as changing the coupling method to Robin-Robin or using a relaxation scheme.

8.2 Aitken relaxation scheme

Consider the iteration by subdomain scheme for the heat transfer problem described in problem 1. Apply 2 iterations of the AITKEN relaxation scheme to it.

This problem has been solved via a Matlab script (see appendix A.2). A comparisson for $\delta t = 0.1$ is presented in figure 2. Note that the colors go from blue to red for each iteration, hence the closer to red the color, the more accurate. One can see how Aitken starts worse but quickly surpasses the standard relaxed shceme in accuracy. Just to fulfill the requirements, here is the result after iteration 2:

```
>> simplify(U)
ans =
0
(9*dt*(196875*dt^2 + 56100*dt + 812))/(50*(421875*dt^3 + 270000*dt^2 + 17100*dt + 208))
(27*dt*(83056640625*dt^5 + 96250781250*dt^4 + 13037625000*dt^3 + 630090000*dt^2 + 12459600*dt + 83872)
(18*dt*(32906250*dt^4 + 33294375*dt^3 + 2274750*dt^2 + 39732*dt + 200))/((75*dt + 4)*(5625*dt^2 + 3300
(81*dt*(18193359375*dt^5 + 19005468750*dt^4 + 2484000000*dt^3 + 126600000*dt^2 + 2618000*dt + 18208))/
0
```

Not a nice expression. Particularized for $\delta t = 0.1$ it becomes much more manageable:

```
>> u
u =
         0
    0.0300
    0.0662
    0.0518
    0.0390
         0
```

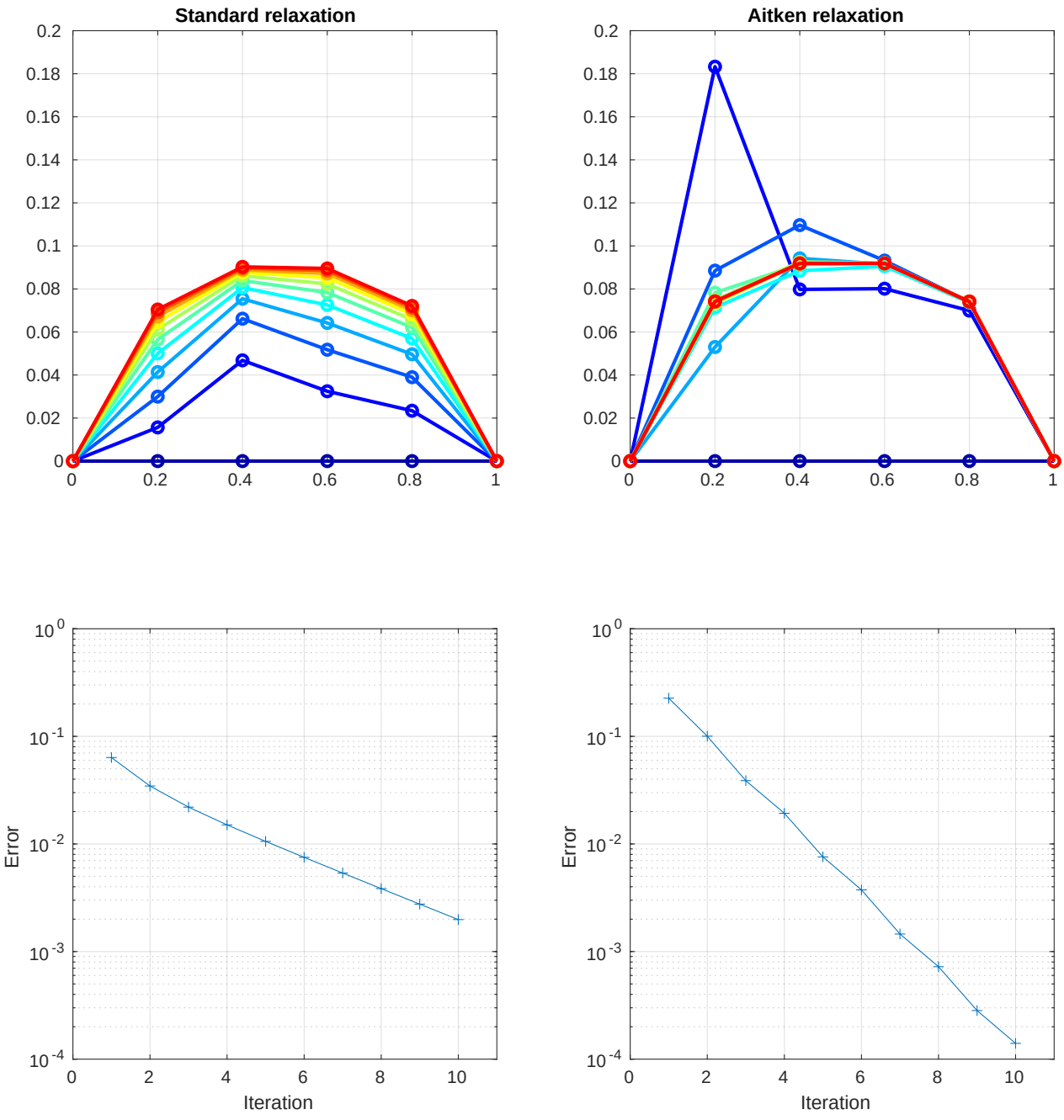


Figure 2: Comparisson between normal relaxation and Aitken relaxation

8.3 Lagrange multipliers

Consider the monolithic (1 domain), transient (BDF1), finite element (linear elements, $h = 1/4$) approximation of the heat transfer equation in problem 1. Enforce the Dirichlet boundary conditions in $x = 0$ and $x = 1$ by using Lagrange multipliers. What is the form of the discrete system? What is the condition number of the resulting matrix?

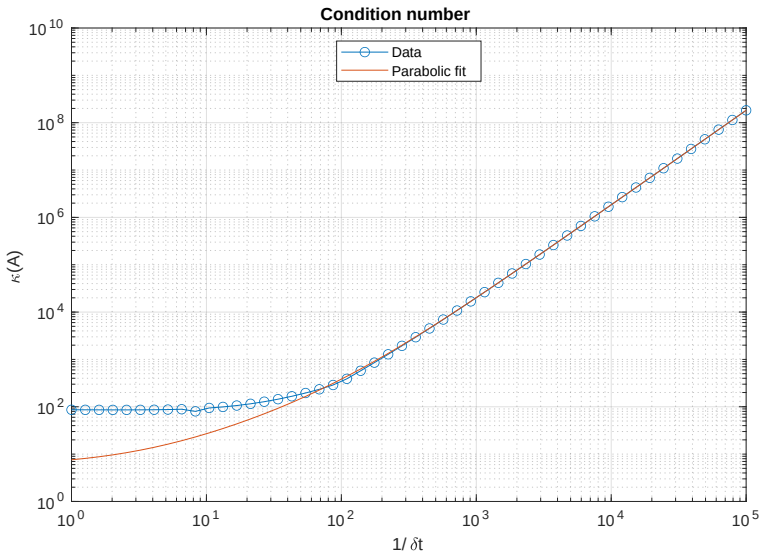
The matrix system has been derived before and it looks like:

$$(\frac{1}{\delta t}M + K)U^{n+1} = F + \frac{1}{\delta t}MU^n \tag{173}$$

This problem has been solved with a Matlab script (see appendix A.3). The resulting matrix, after appending the Lagrange multipliers looks like:

$$A = \begin{bmatrix} \frac{1}{12\delta t}+4 & \frac{1}{24\delta t}-\frac{7}{2} & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{24\delta t}-\frac{9}{2} & \frac{1}{6\delta t}+8 & \frac{1}{24\delta t}-\frac{7}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{24\delta t}-\frac{9}{2} & \frac{1}{6\delta t}+8 & \frac{1}{24\delta t}-\frac{7}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{24\delta t}-\frac{9}{2} & \frac{1}{6\delta t}+8 & \frac{1}{24\delta t}-\frac{7}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{24\delta t}-\frac{9}{2} & \frac{1}{12\delta t}+4 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{174}$$

Calculating the condition number of a matrix with non-constant entries can be quite an odyssey, so we have to particularize for a certain value of δt. The following plot shows the result of a sweep of δt from 1×10^{-5} to 1:



$$p : \tilde{\kappa} = 0.0182 + 1.95\delta t^{\,-1} + 5.62\delta t^{\,-2}$$

Figure 3: Analysis of condition number as a function of δt

First observe that the x axis shows the reciprocal of δt. We can see how well a parabola fits, hence we can estimate that the condition number grows at a rate of $\mathcal{O}(\delta t^{-2})$. This means that the more accuracy we need, the worst the matrix will behave. It is worth mentioning that increasing the number of elements will mitigate the effect somewhat, however the growth is still parabolic.

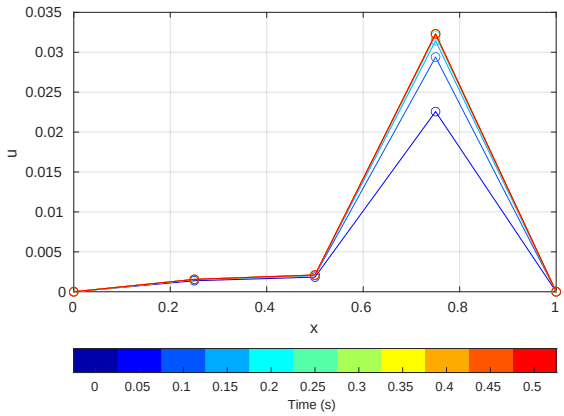
8.4 Split elements

Consider the monolithic (1 domain), transient (BDF1), finite element (linear elements, $h = 1/4$) approximation of the heat transfer equation in problem 1. Suppose that a level set function ($\psi = 0$ at $x = 0.4$) divides the domain into a high thermal conductivity ($\kappa = 100$) subdomain ($x \in [0, 0.4]$) and a low thermal conductivity ($\kappa = 1$) subdomain ($x \in (0.4, 1]$). Build the system matrix for this problem. Take into account the need for subintegrating the element cut by the level set function.

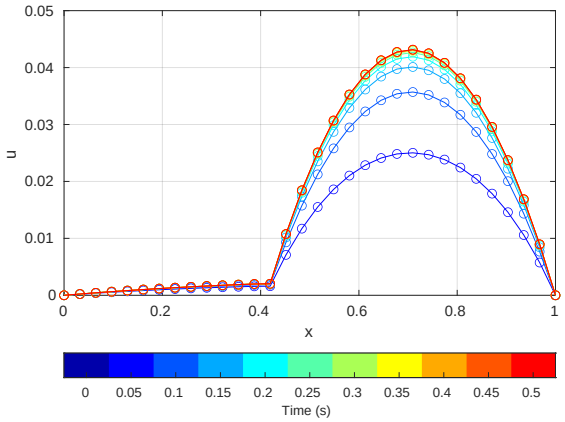
This problem is once again more indicated to solve in Matlab. The whole code is shown in appendix A.4. The most interesting part are the following lines (corresponding to lines 59-85):

```
for i = 1:n_elem
        x1 = dom.x(i);
        x2 = dom.x(i+1);

        if x2 <= x0
           % Left half
           [Me, Ke, Fe] = local_matrices(q, k(1), f(1), h);
        elseif x1 >= x0
           % Right half
           [Me, Ke, Fe] = local_matrices(q, k(2), f(2), h);
        else
           % Hybrid element
           h1 = x0 - x1;
           h2 = x2 - x0;
           [Me1, Ke1, Fe1] = local_matrices(q, k(1), f(1), h1);
           [Me2, Ke2, Fe2] = local_matrices(q, k(2), f(2), h2);
           Me = Me1 + Me2;
           Ke = Ke1 + Ke2;
           Fe = Fe1 + Fe2;
        end

        nodes = [i, i+1];
        K(nodes, nodes) = K(nodes, nodes) + Ke;
        M(nodes, nodes) = M(nodes, nodes) + Me;
        F(nodes) = F(nodes) + Fe;
end
```

This shows the assembly loop. As it can be seen when we find ourselves in the critical element we split it in two smaller elements and we add up the resulting matrices. This is what is referred to as subintegrating. Figure 4 shows the solution for 10 steps with $\delta t = 0.05$. It shows the requested 4 element discretization as well as another with 31 elements to show that it work as expected.The requested system matrix A looks like the following:



(a) Solution with 4 elements



(b) Solution with 31 elements

Figure 4: Results with 4 and 31 elements.

$$A = \begin{bmatrix} \frac{1}{6\delta t} + \frac{3230}{3} & \frac{1}{24\delta t} - \frac{2030}{3} & 0 \\ \frac{1}{24\delta t} - \frac{2030}{3} & \frac{1}{6\delta t} + \frac{2042}{3} & \frac{1}{24\delta t} - 4 \\ 0 & \frac{1}{24\delta t} - 4 & \frac{1}{6\delta t} + 8 \end{bmatrix} \tag{175}$$

A Appendix

A.1 Code for Operator splitting

```
clearvars;
n = 2;
dt = sym('dt', 'positive');

% Matrices
M = 1/(6*(n+1)) * tridiag(1,4,1,n);
K = (n+1) * tridiag(-1,2,-1,n);
C = 0.5 * tridiag(-1,0,1,n);
F = 1/(n+1) * ones(n,1);

% Traditional FEM
A = M/dt + K + C;
X = simplify(A\F);

% Split operator
Aa = simplify(M/dt + C);
Xa = simplify(Aa\zeros(size(F)));

Ak = M/dt + K;
Xk = simplify(Ak\(F + M/dt*Xa));

% Plot
error = [];
k = 0.25;
DT = linspace(0.25 - k,1+k, 300);
for dt = DT
    y_mono = eval(X);
    y_split = eval(Xk);
    error(end+1) = norm(y_split - y_mono);
end

error_star = [];
DTstar = [.25,.5,1];
for dt = DTstar
    y_mono = eval(X);
    y_split = eval(Xk);
    error_star(end+1) = norm(y_split - y_mono);
    fprintf('%.5e\n',error_star(end));
end

plot(DT, error,'b', 'LineWidth',2);
hold on
```

```
scatter(DTstar, error_star,'ob',  'LineWidth',2);
grid on
hold off
title('Splitting error');
legend('Splitting error','Requested values','Location','South');
ylabel('Norm of error');
xlabel(['\delta','t']);
xlim([-inf inf])


function mat = tridiag(a,b,c, n)
    mat = diag(a*ones(1,n-1),-1) + diag(b*ones(1,n))+ diag(c*ones(1,n-1),1);
end
```

A.2 Code for Aitken relaxation scheme comparisson

```
clearvars;
global dt
n_elem1 = 2;
n_elem2 = 3;
inter = 0.4;
maxIter = 10;
tol = 1e-8;
relaxation = 0.3;
f1 = 1;
f2 = 1;

%% Discretization
% Subdomains
s1 = create_subdomain(n_elem1, [0, inter], 1, n_elem1+1, f1);
s2 = create_subdomain(n_elem2, [inter, 1], n_elem2+1, 1, f2);


dt = sym('dt', 'positive');
x1 = linspace(0,inter, n_elem1+1);
x2 = linspace(inter,1, n_elem2+1);
x = [x1(1:end-1), x2];

%% Initial state
s1.X_pre = 0 * sym('x', [s1.n_nodes, 1]);
s2.X_pre = 0 * sym('x', [s2.n_nodes, 1]);

Utilde0 = [s1.X_pre(s1.I); s2.X_pre];
Utilde = Utilde0;


%% Standard relaxation

% Plot initial state
U =  [0; Utilde; 0];
u = eval(subs(U, dt, 0.1));

clf;
subplot(221);
Colors = jet(maxIter+1);
plot(x,u,'o-','Color',Colors(1,:),'LineWidth',2);
Error = zeros(1,maxIter);

% Iteration loop

for i=1:maxIter
    [s1, s2] = iterate_once(Utilde, s1, s2);
```

```
47
48      Utilde_old = Utilde;
49      Utilde = Utilde_old + relaxation * ([s1.X(s1.I); s2.X] - Utilde_old);
50
51      U =  [0; Utilde; 0];
52
53      % Plotting and error
54      x_old = u;
55      u = eval(subs(U, dt, 0.1));
56      error = norm(u - x_old);
57      fprintf('Iteration %2d | Error %.3e | Border %4g\n',i, error, u(s1.G));
58
59      hold on;
60      plot(x,u,'o-','Color',Colors(i+1,:),'LineWidth',2);
61
62      Error(i) = error;
63
64      if error < tol
65          break
66      end
67  end
68  title('Standard relaxation');
69  subplot(223);
70  semilogy(Error(1:i),'+-');
71  grid on
72
73  %% Aitken comparisson
74  fprintf('\nAitken:\n');
75  U0 = Utilde0;
76
77  U =  [0; U0; 0];
78  u = eval(subs(U, dt, 0.1));
79  xlabel('Iteration');
80  ylabel('Error');
81
82  subplot(222);
83  Colors = jet(maxIter+1);
84  plot(x,u,'o-','Color',Colors(1,:),'LineWidth',2);
85  Error = zeros(1,maxIter);
86  title('Aitken relaxation');
87
88  for i = 1:maxIter
89      [s1, s2] = iterate_once(U0, s1, s2);
90      U1 = [s1.X(s1.I); s2.X];
91
92      [s1, s2] = iterate_once(U1, s1, s2);
93      U2 = [s1.X(s1.I); s2.X];
94
95      denominator = (U2 - U1) - (U1 - U0);
```

```
96      U_new = U2 - (U2-U1).^2./denominator;
97
98      % Plotting and error
99      u_old = u;
100     U =  [0; U_new; 0];
101     u = eval(subs(U, dt, 0.1));
102     error = norm(u - u_old);
103     fprintf('Iteration %2d | Error %.3e | Border %4g\n',i, error, u(s1.G));
104     hold on;
105     plot(x,u,'o-','Color',Colors(i+1,:),'LineWidth',2);
106
107     Error(i) = error;
108
109     if error < tol
110         break
111     end
112     U0 = U_new;
113 end
114
115 subplot(224);
116 semilogy(Error(1:i),'+-');
117 grid on
118 xlabel('Iteration');
119 ylabel('Error');
120
121 %% Various functions
122
123 function [s1, s2] = iterate_once(Xtilde, s1, s2)
124     global dt
125
126     % Easier reading
127     I1 = s1.I;  % Interior 1
128     G1 = s1.G;  % Gamma 1
129     G2 = s2.G;  % Gamma 2
130
131     %% Domain 1
132     A = s1.A(I1,I1);
133     B = s1.F(I1) + 1/dt*s1.M(I1, :) * s1.X_pre;
134     % Dirichlet BC
135     B(I1) = B(I1) - s1.A(I1, G1) * Xtilde(G1);
136     % Solving
137     s1.X(I1,1) = simplify(A\B);
138
139     %% Domain 2
140     A = s2.A;
141     B = s2.F + 1/dt*s2.M * s2.X_pre;
142     B(G2) = B(G2) + s1.F(G1) + 1/dt*s1.M(G1,:)*s1.X_pre;
143     % Neumann BC
144     B(G2) = B(G2) - s1.A(G1,I1)*Xtilde(I1) - s1.A(G1, G1) * Xtilde(G1);
```

```
    % Solving
    s2.X = simplify(A\B);

    %% Border value sharing
    s1.X(G1) = s2.X(G2);
end



function subd = create_subdomain(n_elem, xrange, reduce, intersect, f)
    global dt

    G = create_quadrature();

    K = zeros(n_elem+1);
    M = zeros(n_elem+1);
    F = zeros(n_elem+1,1);

    J = (xrange(2) - xrange(1))/n_elem;

    for i = 1:n_elem
        nodes = [i, i+1];

        Ke = zeros(2);
        Me = zeros(2);
        Fe = zeros(2,1);

        for g = G
            gradN = J\g.gradN;

            Ke = Ke + g.w * (gradN' * gradN);
            Me = Me + g.w * (g.N' * g.N);
            Fe = Fe + g.w * g.N' * f;
        end

        K(nodes, nodes) = K(nodes, nodes) + abs(J) * Ke;
        M(nodes, nodes) = M(nodes, nodes) + abs(J) * Me;
        F(nodes) = F(nodes) + abs(J) *  Fe;
    end

    keep = setdiff(1:n_elem+1, reduce);

    subd.A = 1/dt * M(keep, keep) + K(keep, keep);
    subd.M = M(keep, keep);
    subd.F = F(keep);

    subd.I = setdiff(keep, intersect)';
    subd.G = intersect';

```

```
194     if reduce==1
195         subd.I = subd.I - 1;
196         subd.G = subd.G - 1;
197     end
198 
199     subd.n_nodes = size(subd.A, 1);
200 end
201 
202 function G = create_quadrature()
203     gpoints = [-1,1]*sqrt(1/3);
204     Nfun = {@(x)(0.5 - 0.5*x), @(x)(0.5 + 0.5*x)};
205     Nder = [-0.5,0.5];
206 
207     G = [[],[]];
208 
209     for i = 1:2
210         x = gpoints(i);
211         G(i).w = 0.5;
212         G(i).x = x;
213         G(i).N = [Nfun{1}(x), Nfun{2}(x)];
214         G(i).gradN = Nder;
215     end
216 end
```

A.3 Code for Lagrange multipliers conditioning analysis

```
clearvars;
n = 20;
f = 1;
dt = sym('dt', 'positive');
DT = logspace(-5,0,50);

% Matrices
M = 1/(6*n) * laplacian_like(1,4,1,n+1);
K = n * laplacian_like(-1,2,-1,n+1);
C = 0.5 * laplacian_like(-1,0,1,n+1);
F = 1/(n-1) * f * ones(n+1,1);

X0 = zeros(size(F));

% Boundary conditions
BC(1).i = 1;
BC(1).u = 0;

BC(2).i = n+1;
BC(2).u = 0;

% Assembly
A = M/dt + K + C;
b = F + M/dt*X0;

% Lagrange multipliers
for bc=BC
    % Growing matrix
    A(end+1,end+1) = 0;

    %Enforcing values
    A(end, bc.i) = 1;
    A(bc.i, end) = 1;
    b(end+1) = bc.u;
end

X = A\b;
x = X(1:end-length(BC));

k = zeros(size(DT));
for i=1:length(k)
    dt = DT(i);
    a = eval(A);
    k(i) = 1/rcond(a);
end

```

```
clf
loglog(1./DT, k,'o-');
xlabel(['1/\delta','t']);
ylabel('\kappa(A)');
grid on
hold on

% Parabolic fit
p = polyfit(1./DT, k,2);
loglog(1./DT, polyval(p, 1./DT));
fprintf('Best fit:\ny = %f + %f x + %f x^2\n',p(1),p(2),p(3))

title('Condition number');
legend('Data','Parabolic fit','Location','North');

function mat = laplacian_like(a,b,c, n)
    mat = diag(a*ones(1,n-1),-1) + diag(b*ones(1,n))+ diag(c*ones(1,n-1),1);
    mat(1,1) = b/2;
    mat(end,end) = b/2;
end
```

A.4 Code for elements with heterogeneous physical properties

```
clearvars;
global dt

% Simulation parameters
dt = 0.1;% sym('dt','positive');
xrange = [0,1];
n_elem = 4;
n_steps = 10;

% Physiscal properties
f = [1, 1];
k = [100, 1];
x0 = 0.4;

% Boundaries and ititial value
reduce = [1, n_elem+1];
X0 = zeros(n_elem+1, 1);

% Assembling system
domain = init_domain(n_elem, xrange, k, x0, f, X0, reduce);

% Solving a few steps
Colors = jet(n_steps+1);
clf
plot(domain.x, domain.u,'o-','Color', Colors(1,:));
for i=1:n_steps
    domain = step_forward(domain);
    plot(domain.x, domain.u,'o-','Color', Colors(i+1,:));
    hold on
end
grid on
xlabel('x');
ylabel('u');

%%%%%%% End of main program %%%%%%%

function domain = step_forward(domain)
    global dt
    domain.X0 = domain.X;
    domain.b = domain.F + 1/dt * domain.M * domain.X0;
    domain.X = domain.A \ domain.b;
    domain.u = [0; domain.X; 0];
end


function dom = init_domain(n_elem, xrange, k, x0, f, X0, reduce)
```

```
47      global dt
48
49      dom.x = linspace(xrange(1), xrange(2), n_elem+1);
50
51      q = create_quadrature();
52
53      K = zeros(n_elem+1);
54      M = zeros(n_elem+1);
55      F = zeros(n_elem+1,1);
56
57      h = (xrange(2) - xrange(1))/n_elem;
58
59      for i = 1:n_elem
60          x1 = dom.x(i);
61          x2 = dom.x(i+1);
62
63          if x2 <= x0
64             % Left half
65             [Me, Ke, Fe] = local_matrices(q, k(1), f(1), h);
66          elseif x1 >= x0
67             % Right half
68             [Me, Ke, Fe] = local_matrices(q, k(2), f(2), h);
69          else
70             % Hybrid element
71             h1 = x0 - x1;
72             h2 = x2 - x0;
73             [Me1, Ke1, Fe1] = local_matrices(q, k(1), f(1), h1);
74             [Me2, Ke2, Fe2] = local_matrices(q, k(2), f(2), h2);
75             Me = Me1 + Me2;
76             Ke = Ke1 + Ke2;
77             Fe = Fe1 + Fe2;
78          end
79
80          nodes = [i, i+1];
81
82          K(nodes, nodes) = K(nodes, nodes) + Ke;
83          M(nodes, nodes) = M(nodes, nodes) + Me;
84          F(nodes) = F(nodes) + Fe;
85      end
86
87      keep = setdiff(1:n_elem+1, reduce);
88
89      dom.A = 1/dt * M(keep, keep) + K(keep, keep);
90      dom.M = M(keep, keep);
91      dom.F = F(keep);
92      dom.X = X0(keep);
93      dom.u = X0;
94      dom.n_nodes = size(dom.A, 1);
95  end
```

```
96
97  function [Me, Ke, Fe] = local_matrices(G, k, f, h)
98      Ke = zeros(2);
99      Me = zeros(2);
100     Fe = zeros(2,1);
101
102     for g = G
103         gradN = g.gradN * 2/h;
104
105         Ke = Ke + g.w * k * (gradN' * gradN);
106         Me = Me + g.w * (g.N' * g.N);
107         Fe = Fe + g.w * g.N' * f;
108     end
109
110     Me = h * Me;
111     Ke = h * Ke;
112     Fe = h * Fe;
113 end
114
115 function q = create_quadrature()
116     gpoints = [-1,1]*sqrt(1/3);
117     Nfun = {@(x)(0.5 - 0.5*x), @(x)(0.5 + 0.5*x)};
118     Nder = [-0.5,0.5];
119
120     q = [[],[]];
121
122     for i = 1:2
123         x = gpoints(i);
124         q(i).w = 0.5;
125         q(i).x = x;
126         q(i).N = [Nfun{1}(x), Nfun{2}(x)];
127         q(i).gradN = Nder;
128     end
129 end
```