Programming project in TMA4220 - part 2

Do real-life experimentation using your FEM code

In the second part of the project you are going to make use of your finite element library which you have now built. We are going to apply this to one of several real-life applications. The three first tasks introduce the main equations you can choose from

- $\frac{\partial^2 u}{\partial t^2} = \alpha \nabla^2 u$ - the heat equation
- $\nabla \boldsymbol{\sigma}(\boldsymbol{u}) = -\boldsymbol{f}$ - the linear elasticity equation
- $\rho \frac{\partial^2 u}{\partial t^2} = \nabla \boldsymbol{\sigma}(\boldsymbol{u})$ - the free vibration equation

and include some very rough ”getting-started” theory on these.

You should choose **one** of the following exercises.

Exercise 1: The heat equation

Figure 1: A candle

In this exercise we will model heat transfer in a thin metal sheet when a candle is placed under it.

The heat equation reads

$$\begin{aligned} \frac{\partial u}{\partial t} &= \alpha \nabla^2 u + f \\ u(t,x,y)|_{\partial\Omega} &= u^D(t,x,y) \\ u(t,x,y)|_{t=0} &= u_0(x,y) \end{aligned} \tag{1}$$

where α is a positive constant defined by

$$\alpha = \frac{\kappa}{c_p \rho}$$

with κ being the thermal conductivity, ρ the mass density, c_p the specific heat capacity of the material and f a heat source term.

We will let the computational domain Ω be a two dimensional metal sheet, and we want to find the temperature distribution in the metal sheet when a candle fire is placed under it.

a) Semidiscretization

We are going to *semidiscretize* the system by projecting the spatial variables to a finite element subspace X_h. First, assume the source term f is 0, and multiply (1) by a test function v and integrate over the domain Ω to get

$$\iint\limits_{\Omega} \frac{\partial u}{\partial t} v \, dA = -\iint\limits_{\Omega} \alpha \nabla u \nabla v \, dA.$$

Note that we have only semidiscretized the system, so our unknown u is given as a linear combination of the *spatial* basis functions, and is continuous in time, i.e.

$$u_h(x, y, t) = \sum_{i=1}^{n} u_h^i(t) \varphi_i(x, y).$$

The variational form of the problem then reads: Find $u_h \in X_h^D$ such that

$$\begin{aligned} &\iint\limits_{\Omega} \tfrac{\partial u}{\partial t} v \, dA = -\iint\limits_{\Omega} \alpha \nabla u \nabla v \, dA, \quad \forall v \in X_h \\ \Rightarrow \quad &\sum_i \iint\limits_{\Omega} \varphi_i \varphi_j dA \tfrac{\partial u_h^i}{\partial t} = -\sum_i \iint\limits_{\Omega} \alpha \nabla \varphi_i \nabla \varphi_j dA \, u_h^i \quad \forall j \end{aligned}$$

which in turn can be written as the linear system

$$\boldsymbol{M} \frac{\partial \boldsymbol{u}}{\partial t}(t) = -\boldsymbol{A}\boldsymbol{u}(t) \tag{2}$$

which is an ordinary differential equation (ODE) with the matrices defined as

$$\begin{aligned} \boldsymbol{A} = [A_{ij}] &= \iint\limits_{\Omega} \alpha \nabla \varphi_i \nabla \varphi_j \, dV \\ \boldsymbol{M} = [M_{ij}] &= \iint\limits_{\Omega} \varphi_i \varphi_j \, dV. \end{aligned}$$

Construct the matrix $\boldsymbol{A}$ and $\boldsymbol{M}$ as defined above.

b) Include the source term

Assume the heat from the candle light is given by a function

$$f(t, x, y) = e^{-\beta((x^2 + y^2)},$$

for some parameter $\beta > 0$.

What changes in the variational formulation from (a) do you need in order to include this?

c) Time integration

The system (2) is an ODE, which should be familiar from previous courses. Very briefly an ODE is an equation on the form

$$\frac{\partial y}{\partial t} = g(t, y)$$

where y may be a vector. The simplest ODE solver available is Eulers method

$$y_{n+1} = y_n + hg(t_n, y_n).$$

More sophisticated include the improved Eulers methods

$$y_{n+1} = y_n + \frac{h}{2}\left(g(t_n, y_n) + g(t_{n+1}, y_n + hg(t_n, y_n))\right)$$

or the implicit trapezoid rule

$$y_{n+1} = y_n + \frac{h}{2}\left(g(t_n, y_n) + g(t_{n+1}, y_{n+1})\right)$$

and the famous Runge-Kutta methods.

Choose an ODE scheme (based on your previous experience and expertise) and implement your time integration. Why did you choose the solver you did?

d) Experimentation

The parameters κ, ρ and c_p are problem specific. Try to find (by for example googling it) typical values for these for different types of metal.

The free parameters that you can change in this problem are the boundary conditions

$$u(t, x, y)|_{\partial\Omega} = u^D(t, x, y)$$

and the initial condition

$$u(t, x, y)|_{t=0} = u_0(x, y).$$

What do you think are appropriate boundary and initial conditions? Why?

The parameter β in the source term decides how large the candle light is. What happens if you let β be large (for example $\beta = 1000$)?

Use a source function

$$f(t, x, y) = e^{-\beta((x-a\sin(t))^2+y^2)}$$

for different values of a. What happens to the solution when a is large? Do you have to change the time integration somehow?

d)

When the analytical solution is not known, a way to compute the convergence rate is to compare the numerical solution with the numerical solution computed using very small elements.

Compute the solution for element sizes $h \in \{\frac{1}{2^i}, i = 0, 1, 2, 3\}$ and compare these with the solution using element size $h = \frac{1}{2^5}$. Compute the relative error and the convergence rate.

Exercise 2: Linear elasticity equation

We are in this problem going to consider the linear elasticity equation. The equations describe deformation and motion in a continuum. While the entire theory of continuum mechanics is an entire course by itself, it will here be sufficient to only study a small part of this: the linear elasticity. This is governed by three main variables $\boldsymbol{u}, \boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ (see Table 1).

$\boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}$ -	the *displacement* vector measures how much each spatial point has moved in (x, y)-direction
$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{xy} & \varepsilon_{yy} \end{bmatrix}$ -	the *strain* tensor measures how much each spatial point has deformed or stretched
$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$ -	the *stress* tensor measures how much forces per area are acting on a particular spatial point

Table 1: Linear elasticity variables in two dimensions

Note that the subscript denotes vector component and *not* derivative, i.e. $u_x \neq \frac{\partial u}{\partial x}$.

These three variables can be expressed in terms of each other in the following way:

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{u}(\boldsymbol{x}) && (3) \\ \boldsymbol{\varepsilon} &= \boldsymbol{\varepsilon}(\boldsymbol{u}) && (4) \\ \boldsymbol{\sigma} &= \boldsymbol{\sigma}(\boldsymbol{\varepsilon}) && (5) \end{aligned}$$

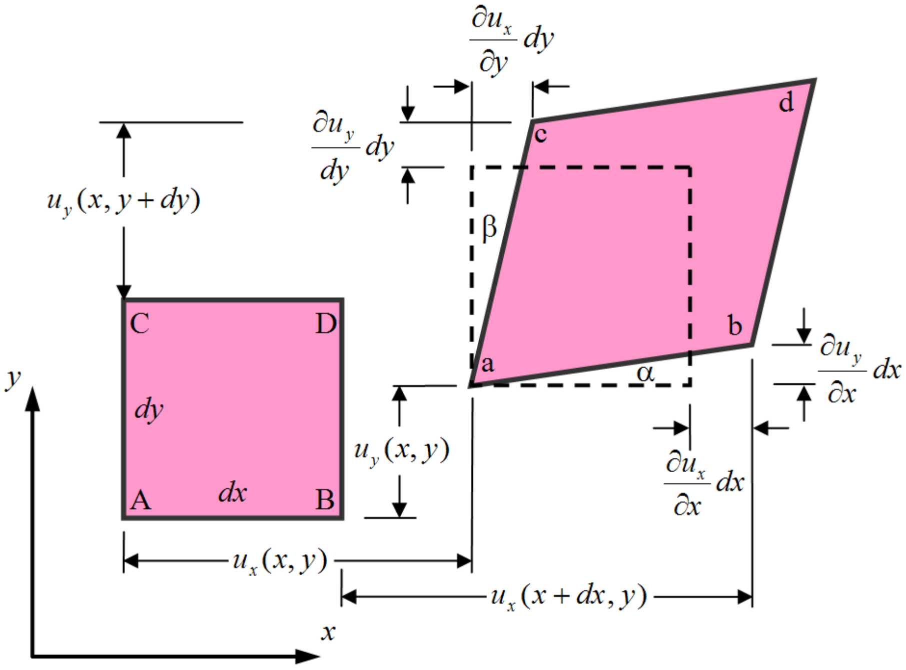
The primary unknown $\boldsymbol{u}$ (the displacement) is the one we are going to find in our finite element implementation. From (3) we will have two displacement values for each finite element "node", one in each of the spatial directions.

The relation (4) is a purely geometric one. Consider an infinitesimal small square of size dx and dy, and its deformed geometry as depicted in figure 2. The strain is defined as the stretching of the element, i.e. $\varepsilon_{xx} = \frac{\mathit{length}(ab) - \mathit{length}(AB)}{\mathit{length}(AB)}$. The complete derivations of these quantities is described well in the Wikipedia article on strain, and the result is the following relations

$$\begin{aligned} \varepsilon_{xx}(\boldsymbol{u}) &= \frac{\partial u_x}{\partial x} \\ \varepsilon_{yy}(\boldsymbol{u}) &= \frac{\partial u_y}{\partial y} \\ \varepsilon_{xy}(\boldsymbol{u}) &= \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}. \end{aligned} \qquad (6)$$

Note that these relations are the *linearized* quantities, which will only be true for small deformations.

For the final relation, which connects the deformation to the forces acting upon it, we turn to the material properties. Again, there is a rich literature on the subject, and different



$\frac{\partial u_x}{\partial y} dy$
$\frac{\partial u_y}{dy} dy$
$u_y(x, y+dy)$
c
d
β
C
D
dy
A
dx
B
a
α
b
$\frac{\partial u_y}{\partial x} dx$
$u_y(x,y)$
$\frac{\partial u_x}{\partial x} dx$
$u_x(x,y)$
$u_x(x+dx,y)$
y
x

Figure 2: An infinitesimal small deformed rectangle

relations or physical laws to describe different materials. In our case, we will study small deformations on solid materials like metal, wood or concrete. It is observed that such materials behave elastically when under stress of a certain limit, i.e. a deformed geometry will return to its initial state if all external forces are removed. Experiment has shown that the Generalized Hooks Law is proving remarkable accurate under such conditions. It states the following. Consider a body being dragged to each side by some stress σ_{xx} as depicted in Figure 3. Hooks law states that the forces on a spring is linearly dependant on the amount of stretching multiplied by some stiffness constant, i.e. $\sigma_{xx} = E\varepsilon_{xx}$. The constant E is called Young's modulus. Generalizing upon this law, we see that materials typically contract in the y-direction, while being dragged in the x-direction. The ratio of compression vs expansion is called Poisson's ratio ν and is expressed as $\varepsilon_{yy} = -\nu\varepsilon_{xx}$. This gives the following relations

$$\begin{aligned} \varepsilon_{xx} &= \frac{1}{E}\sigma_{xx} \\ \varepsilon_{yy} &= -\frac{\nu}{E}\sigma_{xx} \end{aligned}$$

Due to symmetry conditions, we see that when applying a stress σ_{yy} in addition to σ_{xx} we get

$$\begin{aligned} \varepsilon_{xx} &= \frac{1}{E}\sigma_{xx} - \frac{\nu}{E}\sigma_{yy} \\ \varepsilon_{yy} &= \frac{1}{E}\sigma_{yy} - \frac{\nu}{E}\sigma_{xx} \end{aligned}$$

Finally, it can be shown (but we will not) that the relation between the shear strain and shear stress is $\varepsilon_{xy} = 2\frac{1+\nu}{E}\sigma_{xy}$. Collecting the components of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ in a vector, gives us

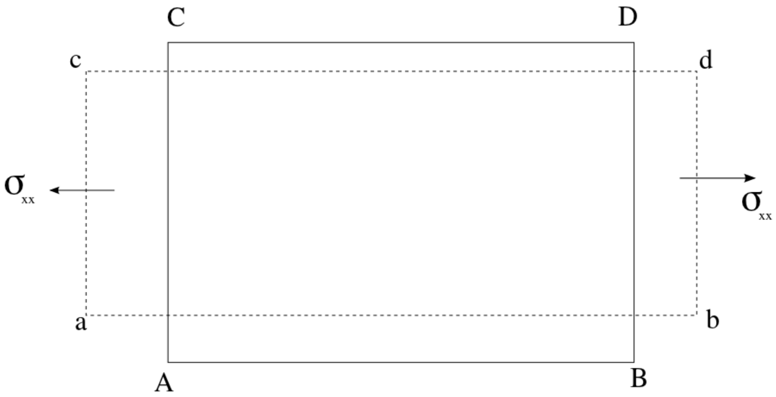


Figure 3: Deformed geometry under axial stresses

the compact notation

$$\begin{aligned}
\bar{\varepsilon} &= C^{-1}\bar{\sigma} \\
\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix} &= \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & 0 \\ 0 & 0 & 2\frac{1+\nu}{E} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}
\end{aligned}$$

or conversely

$$\begin{aligned}
\bar{\sigma} &= C\bar{\varepsilon} \qquad (7) \\
\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} &= \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}
\end{aligned}$$

For a body at static equilibrium, we have the governing equations

$$\begin{aligned}
\nabla \boldsymbol{\sigma}(\boldsymbol{u}) &= -\boldsymbol{f} \qquad (8) \\
\left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right] \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix} &= -\left[f_x, f_y\right]
\end{aligned}$$

and some appropriate boundary conditions

$$\begin{aligned}
\boldsymbol{u} &= \boldsymbol{g}, \quad on \quad \partial\Omega_D \qquad (9) \\
\boldsymbol{\sigma} \cdot \hat{\boldsymbol{n}} &= \boldsymbol{h}, \quad on \quad \partial\Omega_N \qquad (10)
\end{aligned}$$

a) Weak form

Show that (8) can be written as the scalar equation

$$\sum_{i=1}^{2}\sum_{j=i}^{2}\int_{\Omega}\varepsilon_{ij}(\boldsymbol{v})\sigma_{ij}(\boldsymbol{u})\,dA = \sum_{i=1}^{2}\int_{\Omega} v_i f_i\,dA + \sum_{i=1}^{2}\sum_{j=1}^{2}\int_{\partial\Omega} v_i\sigma_{ij}\hat{\boldsymbol{n}}\,d\boldsymbol{S}$$

(where we have exchanged the subscripts (x, y) with $(1, 2)$) by multiplying with a test function $\boldsymbol{v} = \left[\begin{array}{c} v_1(x,y) \\ v_2(x,y) \end{array}\right]$ and integrating over the domain Ω Moreover, show that this can be written in compact vector form as

$$\begin{aligned} \int_{\Omega}\bar{\varepsilon}(\boldsymbol{v})^T C\bar{\varepsilon}(\boldsymbol{u})\,dA &= \int_{\Omega}\boldsymbol{v}^T\boldsymbol{f}\,dA + \int_{\partial\Omega}\boldsymbol{v}^T\boldsymbol{\sigma}\hat{\boldsymbol{n}}\,d\boldsymbol{S} \\ &= \int_{\Omega}\boldsymbol{v}^T\boldsymbol{f}\,dA + \int_{\partial\Omega}\boldsymbol{v}^T\boldsymbol{h}\,d\boldsymbol{S} \end{aligned}$$

b) Galerkin projection

As in 2b) let $\boldsymbol{v}$ be a test function in the space X_h of piecewise linear functions on some triangulation T. Note that unlike before, we now have *vector* test functions. This means that for each node $\hat{i}$, we will have two test functions

$$\begin{aligned} \boldsymbol{\varphi}_{\hat{i},1}(\boldsymbol{x}) &= \left[\begin{array}{c} \varphi_{\hat{i}}(\boldsymbol{x}) \\ 0 \end{array}\right] \\ \boldsymbol{\varphi}_{\hat{i},2}(\boldsymbol{x}) &= \left[\begin{array}{c} 0 \\ \varphi_{\hat{i}}(\boldsymbol{x}) \end{array}\right] \end{aligned}$$

Let these functions be numbered by a single running index $i = 2\hat{i} + d$, where i is the node number in the triangulation and d is the vector component of the function.

Show that by inserting $\boldsymbol{v} = \boldsymbol{\varphi}_j$ and $\boldsymbol{u} = \sum_i \boldsymbol{\varphi}_i u_i$ into (11) you get the system of linear equations

$$A\boldsymbol{u} = \boldsymbol{b}$$

where

$$\begin{aligned} A = [A_{ij}] &= \int_{\Omega}\bar{\varepsilon}(\boldsymbol{\varphi}_i)^T C\bar{\varepsilon}(\boldsymbol{\varphi}_j),\ dA \\ b = [b_i] &= \int_{\Omega}\boldsymbol{\varphi}_i^T\boldsymbol{f}\,dA + \int_{\partial\Omega}\boldsymbol{\varphi}_i^T\boldsymbol{h}\,d\boldsymbol{S} \end{aligned}$$

(**Hint:** $\bar{\varepsilon}(\cdot)$ is a linear operator)

c) Test case

Show that

$$\boldsymbol{u} = \left[\begin{array}{c} (x^2-1)(y^2-1) \\ (x^2-1)(y^2-1) \end{array}\right]$$

is a solution to the problem

$$\begin{aligned} \nabla \boldsymbol{\sigma}(\boldsymbol{u}) &= -\boldsymbol{f} \; in \; \Omega \\ \boldsymbol{u} &= \boldsymbol{0} \; on \; \partial\Omega \end{aligned} \tag{11}$$

where

$$\begin{aligned} f_x &= \frac{E}{1-\nu^2}\left(-2y^2 - x^2 + \nu x^2 - 2\nu xy - 2xy + 3 - \nu\right) \\ f_y &= \frac{E}{1-\nu^2}\left(-2x^2 - y^2 + \nu y^2 - 2\nu xy - 2xy + 3 - \nu\right) \end{aligned}$$

and $\Omega = \{(x, y) \; : \; \max(|x|, |y|) \leq 1\}$ is the refereance square $(-1, 1)^2$.

d) Implementation

Modify your Poisson solver to solve the problem (11). Verify that you are getting the correct result by comparing with the exact solution. The mesh may be obtained through the Grid function `getPlate( )`.

e) Stress analysis

Solving (8) with a finite element method gives you the primary unknown: the displacement u. If you are interested in derived quantities such as the stresses, these can be calculated from (7). Note that σ is in essence the derivative of u which means that since u is C^0 across element boundaries, then σ will be discontinuous. To get stresses at the nodal values, we propose to average the stresses over all neighbouring elements.

Loop over all elements and evaluate (the constant) stresses on that element. For each node, assign the stresses to be the average stress over all neighbouring elements. This method is called "Stress Recovery".

f)

When the analytical solution is not known, a way to compute the convergence rate is to compare the numerical solution with the numerical solution computed using very small elements.

Compute the solution for element sizes $h \in \{\frac{1}{2^i}, i = 0, 1, 2, 3\}$ and compare these with the solution using element size $h = \frac{1}{2^5}$. Compute the relative error and the convergence rate.

Exercise 3: Vibration analysis

Do problem **2a) - 2d)** and read the theory on linear elasticity.

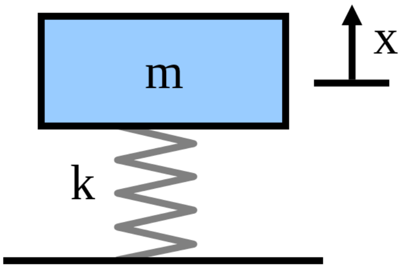


Figure 4: Mass-spring-model

The forces acting on a point mass m by a spring is given by the well known Hooks law:

$$m\ddot{x} = -kx$$

This can be extended to multiple springs and multiple bodies as in figure 5

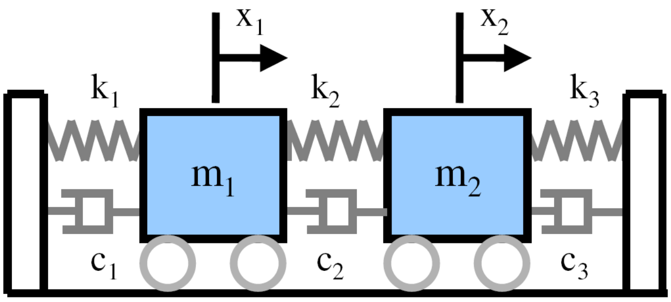


Figure 5: 2 degree-of-freedom mass spring model

The physical laws will now become a system of equations instead of the scalar one above. The forces acting on m_1 is the spring k_1 dragging in negative direction and k_2 dragging in the positive direction.

$$m_1\ddot{x_1} = -k_1x_1 + k_2(x_2 - x_1)$$

This is symmetric, and we have an analogue expression for m_2. The system can be written in matrix form as

$$\begin{aligned} \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \ddot{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} &= \begin{bmatrix} -k_1 - k_2 & k_2 \\ k_2 & -k_2 - k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ M\ddot{\boldsymbol{x}} &= A\boldsymbol{x} \end{aligned}$$

When doing continuum mechanics, it is the exact same idea, but the actual equations differ some. Instead of discrete equations, we have continuous functions in space and the governing equations are

$$\rho\ddot{\boldsymbol{u}} = \nabla\boldsymbol{\sigma}(\boldsymbol{u})$$

semi-discretization yields the following system of equations

$$M\ddot{\boldsymbol{u}} = -A\boldsymbol{u} \tag{12}$$

with the usual stiffness and mass matrix

$$\begin{aligned} \boldsymbol{A} = [A_{ij}] &= \iint\limits_{\Omega} \bar{\varepsilon}(\boldsymbol{\varphi}_i)^T C \bar{\varepsilon}(\boldsymbol{\varphi}_j)\, dV \\ \boldsymbol{M} = [M_{ij}] &= \iint\limits_{\Omega} \rho\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_j\, dV. \end{aligned}$$

e)

Build the 3d mass matrix as given above.

We are now going to search for solutions of the type:

$$\boldsymbol{u}e^{\omega i t} \tag{13}$$

which inserted into (12) yields

$$\omega^2 M\boldsymbol{u} = A\boldsymbol{u} \tag{14}$$

f)

Equation (14) is called a generalized eigenvalue problem (the traditional being with $M = I$). Find the 20 first eigenvalues ω_i and eigenvectors $\boldsymbol{u}_i$ corresponding to this problem.

g)

Let x_0 be your initial geometric description (the nodal values). Plot an animation of the eigenmodes by

$$\boldsymbol{x} = \boldsymbol{x}_0 + \alpha\boldsymbol{u}_i \sin(t)$$

You may want to scale the vibration amplitude by some visually pleasing scalar α, and choose the time steps appropriately. Note that for visualization purposes, you will not use the eigenfrequency ω_i since you are interested in viewing (say) 1-5 complete periods of the vibration, but for engineering purposes this is a very important quantity.

h)

An interesting question is with regards to "harmonic" frequencies. The 1st nonzero frequency is often called the fundamental frequency, and the rest is called overtones. If the overtones are multiplies of the natural frequency (i.e. $f_i = nf_0$, where n is an integer and f_0 is the fundamental frequency) the sound is said to be harmonic. This is an important part in all musical instruments. More information can be found in the Wikipedia article on pitch.

Exercise 3: The best answer

In this problem you are tasked to give the most accurate finite element code you can make. It is going to be a technical problem, and will focus on improving your finite element library. You are going to compute a finite element solution to problems with known exact solutions u and the goal is to get the error $\|u - u_h\|$ as small as possible. From the theory we know that $\|u - u_h\| \approx Ch^p$, so there are two ways of improving the error

- decrease h
- increase p

Both strategies will require major revisions to your existing library from task 1; either from implementing new basis functions (p-refinement), or optimizing existing algorithms to handle large number of unknowns (h-refinement).

a) Profile your code

Read about *profiling* at the Python documentation page. Use some profile library to find out which part of your program is the slowest.

b) Compute the error

To compute the error of you finite element solution, you will need to integrate this in some norm. One usually measures the error of finite element problems in one of the three norms,

$$\begin{aligned}
|u - u_h|_{H^1}^2 &= \int_\Omega \nabla(u - u_h) \cdot \nabla(u - u_h) dA \\
\|u - u_h\|_{L^2}^2 &= \int_\Omega (u - u_h)^2 dA \\
\|u - u_h\|_{L^\infty} &= \sup_{x \in \Omega} |u(x) - u_h(x)|.
\end{aligned}$$

The h^1-seminorm is usually denoted as the energy norm, and satisfies $a(u,u) = |u|_{H^1}^2$. From the Galerkin orthogonality, we know that the finite element solution is the best possible solution (in our solution space) as measured in the energy norm. It is the most natural, and predictive way of computing the error.

Compute your error in energy norm on the problem from task 2 and task 3 in part 1 of the project. To compute the continuous integral of the norms, you will need to split it into a sum of integration over single elements, and evaluate the error functions here. You will find your assembly procedure from task 2a to be helpful.

c) Get convergence plots

Again, we will use the analytical solutions from part I. Compute your solution on a series of meshes of different sizes. Plot the error you are getting vs the element size h in a log-log plot. What do you see? Explain your findings.

d)

When the analytical solution is not known, a way to compute the convergence rate is to compare the numerical solution with the numerical solution computed using very small elements.

Compute the solution for element sizes $h \in \{\frac{1}{2^i}, i = 0, 1, 2, 3\}$ and compare these with the solution using element size $h = \frac{1}{2^5}$. Compute the relative error and the convergence rate.

e) Machine precision

Computers operate with a limited number of decimal digits. For your typical computing environment, this is of the order $\mathcal{O}(10^{-15})$. Can you create a 2D finite element solution which reaches machine precision? How large system did you need, and how long time did the computations take? You can use the `time`-module in Python to measure this.

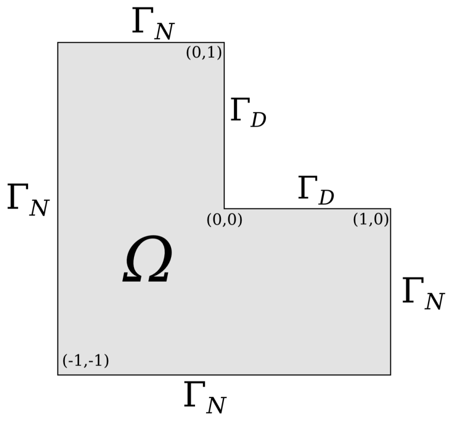


Figure 6: The L shape

f) The L-shape

Let the domain $\Omega = [-1, 1]^2/[0, 1]^2$ be the L-shaped domain around the origin (see Figure 6). Solve the Poisson problem

$$\begin{aligned} \nabla^2 u &= 0 \text{ in } \Omega, \\ u &= 0 \text{ on } \Gamma_D, \\ \frac{\partial u}{\partial n} &= g \text{ on } \Gamma_N. \end{aligned}$$

With the known exact solution

$$u(r, \theta) = r^{2/3} \sin(\frac{2\theta + 2\pi}{3}), \qquad \theta \in \big[\frac{\pi}{2}, 2\pi\big] \tag{15}$$

compute g from (15) and run your finite element program on this problem. How fine solution are you able to obtain? How long did your simulation take?