

AME50541: Finite Element Methods
Homework 4: Due Friday, March 22 2019

Be sure to email all code to the instructor and TA.

Problem 1: (20 points) Consider the standard second-order PDE in one-dimension

$$-\frac{d}{dx} \left(a(x) \frac{du}{dx} \right) + c(x)u(x) = f(x), \quad 0 < x < L$$
$$u(0) = \bar{u}_0, \quad u(L) = \bar{u}_L$$

where $a(x)$, $c(x)$, and $f(x)$ are polynomials over $(0, L)$ of degrees p_a , p_c , and p_f , respectively. The stiffness matrix and force vector are

$$K_{ij}^e = \int_{\Omega^e} \left(\frac{d\phi_i^e}{dx} a(x) \frac{d\phi_j^e}{dx} + \phi_i^e(x) c(x) \phi_j^e(x) \right) dV, \quad F_i^e = \int_{\Omega^e} \phi_i^e(x) f(x) dV,$$

assuming a general element basis $\{\phi_i^e\}_{i=1, \dots, p+1}$.

- (a) If we use basis functions of order p in the physical domain, how many Gaussian quadrature points are needed to integrate the stiffness matrix exactly? How many are needed to integrate force vector exactly? Your answer should be in terms of p , p_a , p_c , p_f . What happens if either $a(x)$, $c(x)$, or $f(x)$ are non-polynomial functions?
- (b) Instead, suppose we use basis functions of order p in the reference domain and use the isoparametric concept to define basis functions over the physical element. That is, let $\{\psi_i\}_{i=1, \dots, p+1}$ be the basis functions of order p defined over the parent element $\Omega^\square := (-1, 1)$ with equally spaced nodes. The isoparametric mapping is

$$x^e(\xi) = \mathcal{G}^e(\xi) = \sum_{i=1}^{p+1} x_i^e \psi_i(\xi),$$

where $\{x_i^e\}_{i=1, \dots, p+1}$ are the positions of the nodes of element e in the physical space and $x^e(\xi)$ is the position in the physical element corresponding to the position ξ in the reference element. The basis functions in the physical space are then defined as

$$\phi_i^e(x) = \psi_i(\xi^e(x)),$$

where $\xi^e = (\mathcal{G}^e)^{-1}$ is the inverse of the isoparametric mapping. Re-write the integrals over the physical element defining the element stiffness matrix and force vector to integrals over the reference domain. You will need to use the following relationship:

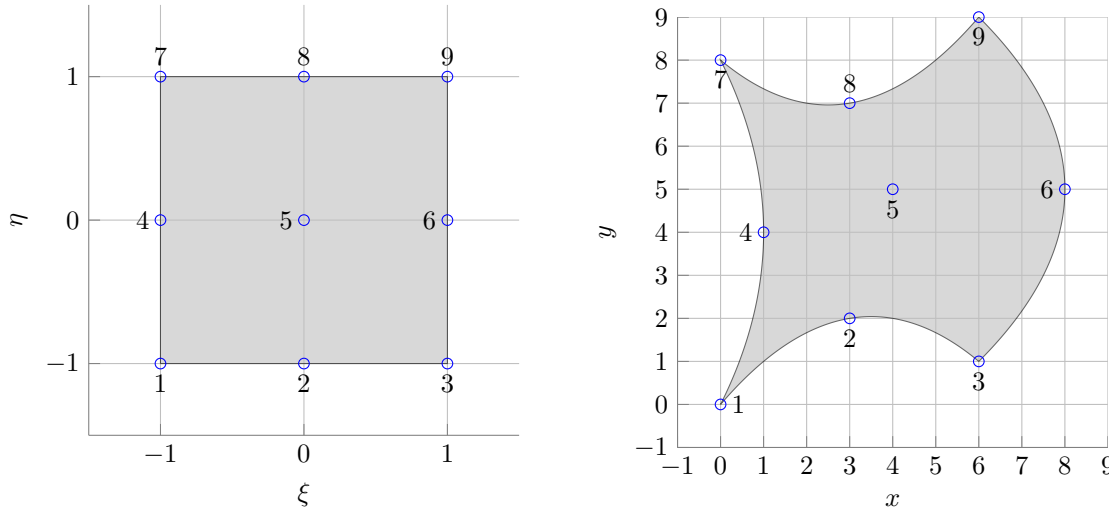
$$\frac{d\phi_i^e}{dx} = \frac{d\psi_i}{d\xi} \frac{d\xi^e}{dx} = \frac{d\psi_i}{d\xi} \left(\frac{dx^e}{d\xi} \right)^{-1}.$$

How many quadrature nodes are needed to exactly integrate the stiffness matrix and force vector in the isoparametric setting? What is special about $p = 1$?

Problem 2: (20 points) The one-dimensional four-node isoparametric element has nodes in the parent domain located at $-1, -1/3, 1/3, 1$.

- (a) Write out the isoparametric shape functions $\psi_i(\xi)$.
- (b) Write out the isoparametric mapping, assuming an element with physical nodes located at $0, 1, 1.5, 2$.
- (c) Compute the isoparametric Jacobian $g_e = \partial x^e / \partial \xi$ and plot it over $\xi \in (-1, 1)$. Is the Jacobian positive everywhere?
- (d) Repeat (b) and (c) assuming an element with physical nodes at $0, 1.5, 1, 2$, i.e., the second and third node have “crossed over” each other. What happens to the Jacobian?

Problem 3: (30 points) Consider the nine-node quadrilateral element (Lagrangian) element below.



Nine-node Lagrangian quadrilateral element in the reference domain ($\xi - \eta$ space) (left) and the physical domain ($x - y$ space) (right).

- What is the Jacobian of the isoparametric mapping and its determinant? You may write it in terms of $\{\psi_i\}_{i=1,\dots,9}$, the basis functions for the bi-quadratic quadrilateral element in the parent domain (rather than explicitly in terms of ξ, η). The nodal coordinates of the element can be read directly from the figure.
- Plot the determinant of the Jacobian of the isoparametric mapping over the reference (parent) element. You can use the functions you wrote in Problems 1 and 2 of Homework 3 (or the solutions provided online) to create the basis functions for the bi-quadratic element. To generate a 3D plot, you want to use the `meshgrid` command to generate a grid of points and `surf` to plot the determinant on the grid.
- Compute the area of the element, $A = \int_{\Omega^e} da = \int_{\Omega^e} g(\xi, \eta) dA$ where $g(\xi, \eta)$ is the determinant of the Jacobian of the mapping, using Gaussian quadrature (use the function `quadhcube_gaussleg` provided in the Homework 4 code distribution to generate the quadrature points and weights). What is the minimum number of quadrature points required in each dimension to compute the integral exactly? Verify this: approximate the area with an increasing number of quadrature nodes in each dimension and check that the area converges once the minimum number of quadrature nodes for exact integration is exceeded.

Problem 4: (30 points) Due to the *local* nature of the finite element basis functions, the global stiffness matrix is always *sparse*, i.e., many of its entries are zero. Significant computational savings (both in terms of memory usage and computing time) are available if the stiffness matrix is stored in a format that takes advantage of this knowledge, i.e., a sparse matrix format.

- Describe which entries of the stiffness matrix are non-zeros strictly in terms of the `ldof2gdof` matrix.
- Consider a problem with 7 degree of freedoms and the following `ldof2gdof` matrix

$$\text{ldof2gdof} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \\ 3 & 5 & 7 \end{bmatrix}.$$

How many entries are there in the stiffness matrix? How many of them are non-zero (be careful to avoid double counting)? List the row-column pairs of the stiffness matrix that contain a non-zero entry.

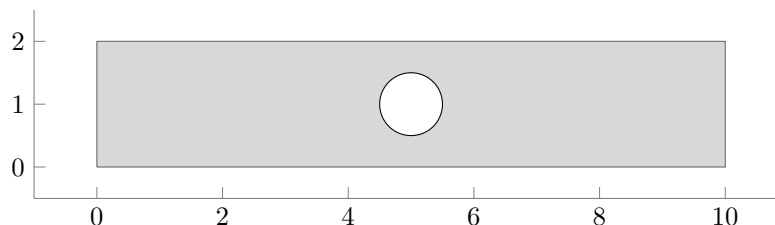
- (c) Use the functions provided in the Homework 4 code distribution (`create_unif_mesh_hcube.m` and `create_map_ldof_to_gdof.m`) to create the `ldof2gdof` matrix for a sequence of meshes in 1, 2, and 3 dimensions:
- (i) One dimension: polynomial order $p = 2$, number of elements = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024.
 - (ii) Two dimensions: polynomial order $p = 2$, number of elements = 1, 2, 4, 8, 16, 32, 64, 128, 256.
 - (iii) Three dimensions: polynomial order $p = 2$, number of elements = 1, 2, 4, 8, 16, 32.

For each sequence of meshes, plot the ratio between the number of non-zeros in the stiffness matrix and the total number of entries in the matrix. You do not need to worry about counting the number of non-zeros in the stiffness matrix associated with each mesh. Instead, use the function `create_sparsity_struct` provided in the Homework 4 code distribution to form the sparsity structure of each matrix (the number of rows in the output `cooidx` is the number of non-zeros in the stiffness matrix). Assume one degree of freedom per node when constructing the `ldof2gdof` matrix. Warning: Do not actually form the dense stiffness matrix for the larger mesh.

- (d) For the mesh with $32 \times 32 \times 32$ tri-quadratic elements, how much memory is saved by using a coordinate sparsity structure that stores the triplet (row number, column number, matrix value) for each non-zero entry of the matrix over a dense matrix?
- (e) Plot the sparsity structure (use the `spy` command in MATLAB) for the 1D mesh with 5 quadratic element elements, the 2D mesh with 5×5 bi-quadratic elements, and the 3D mesh with $5 \times 5 \times 5$ tri-quadratic elements.

Problem 5: (40 points) From S. Govindjee, UC Berkeley. In this problem you will examine the behavior of linear vs quadratic elements in trying to estimate the stress around a hole in a plate.

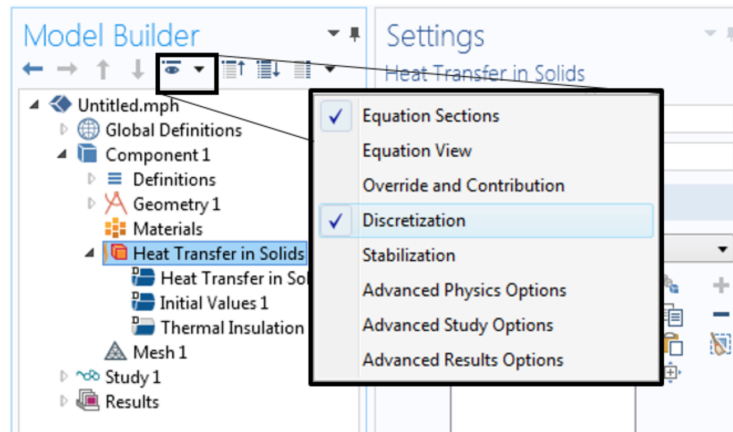
- 1) Start COMSOL: 2D analysis, structural mechanics, stationary analysis.
- 2) Under *Global Definition* and *Parameters*
 - Right-click *Parameters* and define a new parameter named *mesh_size* (under *Name*). We will be varying the *Expression* value; set it to 0.05 for now.
- 3) Create the geometry as follows:
 - Rectangle: $0 \leq x \leq 10\text{m}$ and $0 \leq y \leq 2\text{m}$
 - Circle: center (5,1)m and $r = 0.5\text{m}$
 - Take the difference of the rectangle and the circle (using *Booleans and Partitions, Difference* under the *Geometry* tab) to create the domain shown in the figure. If you have trouble selecting the appropriate object when creating the difference, use the middle button to cycle through objects while hovering the mouse over them.



- Add a point at $(x, y) = (0, 1)$ for later use

Our objective is to compute the maximum von Mises stress $\max \sigma_{mises}$ throughout the domain using different mesh densities and element types.

- 4) Within *Solid Mechanics* set the *2D Approximation* to *Plane Stress* and the *Thickness* to 1 mm. Also set the *Discretization* to *Linear*. If you do not see *Discretization* in the *Solid Mechanics* settings, you can add it through the *Model Builder* window.



- 5) Set the elastic properties as: Young's modulus = 70 GPa and Poisson's ratio = 0.33.
- 6) Add a *Prescribed Displacement* and a *Prescribed Load*. On the left edge, fix the displacement in the x -direction to be zero. At the point (0, 1), fix the displacement in both directions. On the right edge, set the load type to *Total Force* with a value of 1 kN in the x -direction.
- 7) Define a *User-controlled mesh* with *Maximum element size* set to be *mesh_size* (your parameter). This forces the mesh to have elements roughly the size of the parameter value.
- 8) Right-click *Study* and add a *Parametric Sweep*. Left-click *Parametric Sweep* and add *mesh_size* as a parameter name using the plus sign and in the *Parameter values* field enter:
1 0.8 0.6 0.4 0.2 0.1 0.09 0.08 0.07 0.06 0.05.
- 9) If you now solve (*Compute* under *Study*), COMSOL will solve the problem using the sequence of meshes you defined.
- 10) The maximum von Mises stress happens at the top and bottom of the hole. To extract the needed information, use the *Derived Values* option with a *Point Evaluation*. To get the correct variable in *Point Evaluation* change *Expression* to *solid.mises* by typing it in or using the menus. Make sure the *Data set* is *Study1/Parametric Solutions 1*. If you now *Evaluate All* (*Results* tab), it will create a table of values for you for each mesh density.
- 11) Repeat this study with quadratic elements. Go back to *Solid Mechanics* and change the *Discretization* to *Quadratic*. Also go to the *Mesh* tab and change the size to $2 \times \text{mesh_size}$. This will keep the number of nodes close to that of the linear case (more fair comparison). Go to *Study* and *Compute* a new solution. Re-evaluate your *Point Evaluation*. This will add a new column to your results table.
- 12) Export the table with both columns (linear and quadratic elements) and create a plot of mesh size vs. maximum von Mises stress (use a semi-logx plot). Don't forget that your quadratic elements have elements roughly twice the size of your linear elements (even though it won't be reflected in your table). Comment on the meaning of what you observe. Also export a plot of the contours of the von Mises stress over the plate.