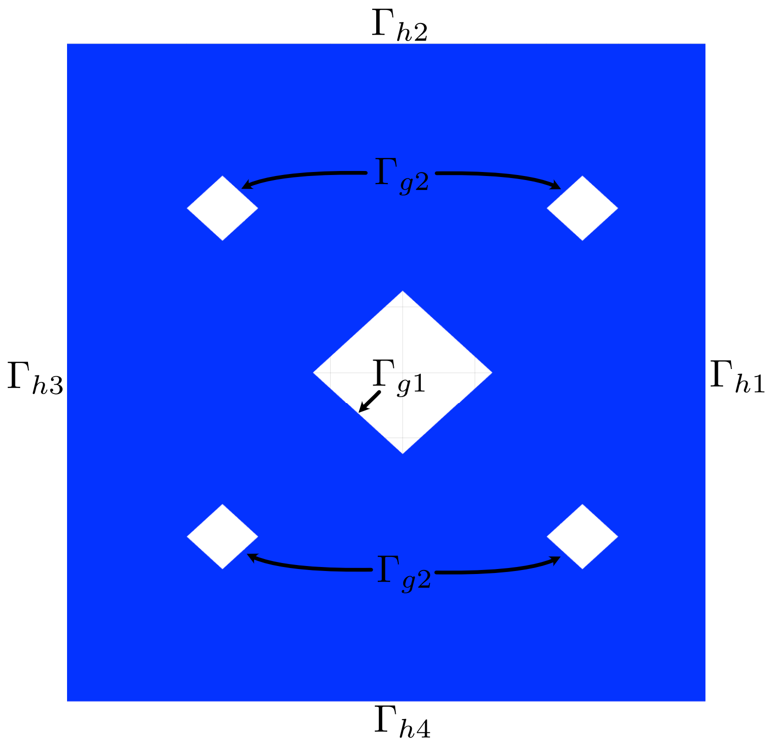
ME 335A
Finite Element Analysis
Instructor: Adrian Lew
Computing Project #2

May 13, 2023

Due Wednesday, May 24, 2023

In this assignment you will be solving a two dimensional heat conduction problem on a square domain $(-1, 1) \times (-1, 1)$ with holes. All lengths are given in feet. In this problem, both interior and exterior boundaries appear. The external boundaries have natural (Neumann) boundary conditions applied, whereas the interior boundaries of the holes have essential (Dirichlet) boundary conditions applied to them. To stick to polygonal geometries, we have made all holes with a square shape. Relative to a coordinate system at the center of the square, the central hole has its four vertices at $(1/4, 0), (0, 1/4), (-1/4, 0), (0, -1/4)$, the four peripheral holes have theirs at $v + \{(1/10, 0), (0, 1/10), (-1/10, 0), (0, -1/10)\}$, where v is any of the vectors with components $(\pm 1/2, \pm 1/2)$. This geometry and the specified boundary conditions represents is inspired by a model engine block, where the center bore is in contact with a high temperature piston. The peripheral holes are channels where the coolant circulates at a lower temperature. The problem domain is denoted Ω. The central bore hole boundary will be denoted Γ_{g1} and the cooling hole boundaries (union of all) will be denoted Γ_{g2}. The external faces of the block are denoted Γ_{h1} through Γ_{h4}, as shown.



Recall that the governing differential equation for this heat conduction problem is $-(K_{ij}u,_j),_i = f$. We will treat the engine block as being thermally isotropic and homogeneous, so that K_{ij} can be replaced with $k\delta_i j$, where k is the constant-valued isotropic thermal conductivity and δ_{ij} is the Kronecker delta.

Now the problem‘ may be stated as: Find the temperature $u \in C^2(\Omega)$ such that:

$$\begin{aligned}
-k\Delta u &= f && \text{in} && \Omega \\
u &= g_1 && \text{on} && \Gamma_{g_1} \\
u &= g_2 && \text{on} && \Gamma_{g_2} \\
k\nabla u \cdot \check{n} &= h_1 && \text{on} && \Gamma_{h1} \\
k\nabla u \cdot \check{n} &= h_2 && \text{on} && \Gamma_{h2} \\
k\nabla u \cdot \check{n} &= h_3 && \text{on} && \Gamma_{h3} \\
k\nabla u \cdot \check{n} &= h_4 && \text{on} && \Gamma_{h4}
\end{aligned}$$

We will assume a constant conductivity $k = 2 \times 10^{-4}$W ft^{-1} °C^{-1}.

Getting Started. Posted on Canvas you will find two files, `CP2.m` and `CP2Mesh.m`. The first file is the incomplete file that you need to fill in, just as in CP-1. The second file is the custom-made function that creates meshes for the geometry of this problem. It has a single function, `CP2Mesh(HMax)`, where `HMax` is the largest acceptable mesh size. The function then triangulates the domain with triangle, and returns the arrays X, LV, and BE as described in the notes, and one more array BN. This last array contain the indices of all vertices that lie on the boundary of the domain: BN(1,i) gives the vertex number in LV of the i-th vertex in BN, while BN(2,i) indicates to what line of the boundary it belongs to according to:

Value	Line
1	Γ_{h1}
2	Γ_{h2}
3	Γ_{h3}
4	Γ_{h4}
5	Γ_{g1}
6	Γ_{g2}

The data for this problem is specified as follows,

$$\begin{aligned} f &= 0.2\mathrm{W}/\mathrm{ft}^3 \\ g_1 &= 2500^\circ C \\ g_2 &= 100^\circ C \\ h_1 &= h_2 = h_3 = h_4 = 0.0 \end{aligned}$$

We added a non-zero heat source f to make the problem slighlty more interesting. All the external (natural) boundaries are homogeneous, which corresponds to thermally insulating those boundaries (no heat flux in or out).

1. (10) The first task is to code the element stiffness matrix and element load vector for a P_1 triangular element, as well as the section that imposes boundary conditions. The assembly is already shown in the notes, so it is already included. The function that computes K^e and F^e is also in the notes, but it'd be a good idea for you to try to code it yourself, for your learning.

   Once you have completed these definitions, run your code. You should now be able to generate the solution to the heat conduction problem described above. For grading purposes, please submit a plot of the solution and the code with your additions.

2. We now want to evaluate the function and its derivative at any point in the domain. To this end, complete the function `P1Functions` to compute the value of the shape functions and their derivatives at any point in an element.

   Once you do, the function serves also as a test of whether a point x is in the element or not. To this end, recall that the shape functions are the barycentric coordinates, and points in the triangle have all their barycentric coordinates in the interval $[0, 1]$. If any of them is not in that interval, x does not belong to the element.

3. Next, complete the function `[u, du]=uValue(xp, X, LV, U)`, which evaluates the solution and its gradient at `xp`, and returns zero arrays if the point is outside the domain (it does not fall in any element).

   A crucial part of this step is to search the element, if any, to which `xp` belongs to. You can use a brute force search here, although the fast way to do it would be through a quadtree.

4. (10) Evaluate your solution and its gradient at (0,0.9), (0.25,0.25) for HMax=0.1. Include the units in the answer.

5. (10) We will now modify the stated boundary value problem to incorporate a nonhomogenous natural boundary condition. The data for this problem is the same as above except now,

$$\begin{aligned} h_1 &= 0.9\mathrm{Wft}^{-2} \\ h_2 &= 0.0 \\ h_3 &= -0.1\mathrm{Wft}^{-2} \\ h_4 &= 0.0. \end{aligned}$$

The heat flux values are constant (or zero) on each face and correspond to insulating the left and right edges of the domain, supplying heat through the top edge, and removing heat from the bottom edge.

Your task is to add the contributions of the Neumann boundary conditions. The code is provided in the notes, but it'd be good idea to try to do it.

Once you have completed these definitions, run your code. You should now be able to generate the solution to the heat conduction problem described above. For grading purposes, please submit a plot of the solution and the code with your additions.

6. We will now assess convergence of the solution in the absence of an exact solution. If the implementation of the pointwise function evaluation is not efficient, it would be very costly to evaluate the L^2 or H^1 norms of the difference of solutions with different meshes. Instead, we will sample the function and its gradient at a few points, to keep the computational cost under control. So, we define the following error measures, trying to mimic the norms (but they are not norms):

$$e_{0,2}(u) = \left[\sum_{i,j=0}^{N} u(X_{ij})^2\right]^{1/2}, \qquad e_{1,2}(u) = \left[\sum_{i,j=0}^{N} \|\nabla u(X_{i,j})\|^2\right]^{1/2}.$$

We will set $X_{ij} = (-1+2i/N, -1+2j/N)$ with $N = 10$. If $X_{ij} \notin \Omega$, just set $u = 0$ and $\|\nabla u\| = 0$.

Compute the solutions of part 1 for HMax$\in \{0.2, 0.1, 0.05, 0.025\}$, and export the values at these points, so that you can later compute the errors. Let u_1, u_2, u_3 and u_4 denote the 4 solutions, respectively.

Since we do not know what the error is, we will evaluate how the numerical solution changes as we refine the mesh. So, if we expect

$$\|u - u_h\| \leq C_u h^r$$

for some order of convergence $r > 0$, from the triangle inquality we know that

$$\|u_{h/2^n} - u_{h/2^{n+1}}\| \leq \|u_{h/2^n} - u\| + \|u_{h/2^{n+1}} - u\| \leq C_u 2 \left(\frac{h}{2^n}\right)^r$$

Hence, by evaluating how the error between consecutive bisections of the mesh decreases, we can evaluate whether the error is already decreasing in the asymptotic form we expect.

We can have a rough idea of the value of C_u and a good idea of the value of r by plotting $\log(\|u_{h/2^n} - u_{h/2^{n+1}}\|)$ as a function of $\log(h/2^n)$, and find the slope for r and the intersection with the y-axis for $\log(2C_u)$. If we have confidence that our method converges, then we could use $C_u h^r$ as an estimate of how large our error is.

Questions:

(a) (15) Plot $\log\left(e_{a,2}(u_i - u_{i+1})\right)$ for $a = 0, 1$ as a function of log(HMax) for i=1,2,3. Compute the slope between the last two pairs of points. What is the observed order of convergence for $a = 0, 1$? Please provide the plots as part of your solution.

(b) (15) Plot the differences between consecutive solutions at all points X_{ij}. What mesh do you need to compute the temperature at *all* points $\{X_{ij}\}$ with less that 30°C of error? Explain, and provide the plots as part of your solution.