

Math 228B: Project 4 Writeup

Hongli Zhao

April 23, 2020

1 2D Poisson Problem with Homogeneous Neumann Boundary Conditions

$$\begin{cases} -\nabla^2 u(x, y) = 1, & \text{on } \Omega \\ n \cdot \nabla u = 0, & \text{on } \Gamma_1 \\ u = 0, & \text{on } \Gamma_2 \end{cases}$$

and the boundary is $\Gamma = \Gamma_1 \cup \Gamma_2$.

In the general case where we solve:

$$\begin{cases} -\nabla^2 u(x, y) = f \\ n \cdot \nabla u = g \end{cases}$$

First by multiplying both sides of the PDE by a test function $v \in V_h$ and integrating by parts yields:

$$\int_{\Omega} -\nabla^2 u_h v dx = \int_{\Omega} f v dx$$

then applying divergence theorem and using the Neumann condition gives:

$$\int_{\Omega} \nabla u_n \cdot \nabla v dx = \int_{\Omega} f v dx + \oint_{\Gamma} g v ds$$

Using $u_h(x) = \sum_{j=1}^n \xi_j \phi_j$ and let $v = \phi_i$:

$$\int_{\Omega} \left[\sum_{j=1}^n \xi_j \nabla \phi_j \right] \cdot \nabla \phi_i dx = \int_{\Omega} f \phi_i dx + \oint_{\Gamma} g \phi_i ds$$

Switching the order of integration and summation, we form the system:

$$A\xi = b$$

where:

$$A_{ij} = \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j dx$$

$$b_i = \int_{\Omega} f \phi_i dx + \oint_{\Gamma} g \phi_i ds$$

In our problem, $f = 1$ for all $\omega \in \Omega$, $g = 0$ for Neumann condition on the boundary Γ . Restricting the function to each triangle, the element stiffness matrix and load vector are given by:

$$a_{ij}^K = \int_K \nabla \phi_i \nabla \phi_j dx$$

$$b_i^K = \int_K f \phi_i dx + \int_{K \cap \Gamma} g \phi_i dx = \int_K \phi_i dx$$

Here we derive how to compute the entries on a general triangle, using piecewise linear elements ϕ_j . Let K be described by nodes N_1, N_2, N_3 whose coordinates are (x_i, y_i) for $i = 1, 2, 3$. Then using the fact that $\phi_i(N_j) = \delta_{ij}$, we fit a linear function by solving:

$$\begin{cases} \phi_i(N_i) = 1 \\ \phi_i(N_j) = 0 \\ \phi_i(N_k) = 0 \end{cases}$$

Let $\phi_i = ax + by + c$ be the general form, we plug in the coordinates and solve a system:

$$\begin{cases} ax_i + by_i + c = 1 \\ ax_j + by_j + c = 0 \\ ax_k + by_k + c = 0 \end{cases}$$

yielding:

$$\begin{cases} a = \hat{a} = -\frac{y_k - y_j}{x_k - x_j} b \\ b = \hat{b} = \frac{1}{y_i - y_j} + \frac{(x_i - x_j)(y_k - y_j)}{(y_i - y_j)(x_k - x_j)} \\ c = \hat{c} = 1 - x_i a - y_i b \end{cases}$$

Knowing explicitly what $\hat{a}, \hat{b}, \hat{c}$ are, we can get explicitly the piecewise linear functions:

$$\phi_i(x, y) = a_i x + b_i y + c_i$$

Then:

$$\nabla \phi_i = a_i + b_i$$

For the local stiffness matrix, we can compute the entries explicitly:

$$a_{ij}^K = \int_K \nabla \phi_i \nabla \phi_j dx = [(a_i \cdot a_j) + (b_i \cdot b_j)] \cdot (\text{Area of } K)$$

where a_i, a_j, b_i, b_j are coefficients of the basis functions. Given that we know the coordinates for each triangle K , the area can easily be computed by using the system matrix:

$$B = \begin{pmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{pmatrix}$$

then we have $\det B = 2 \cdot (\text{Area of } K)$, or $(\text{Area of } K) = \frac{1}{2} \det B$.

For the load vector:

$$b_i = \int_K \phi_i dx$$

this corresponds to the volume under our piecewise linear "plane", over the triangular area K , this corresponds to $\frac{1}{3}$ of the volume of the prism with unit height, or:

$$b_i = \frac{1}{3} \cdot (\text{Area of } K \cdot 1) = \frac{1}{6} \det B$$

Now that we have explicitly the entries of A and load vector b , we can assemble the global stiffness matrix using the stamping method.

After we have computed the global system vector ξ for the Neumann problem, we can then impose the Dirichlet data by directly replacing the ξ 's for nodes that are on the Dirichlet boundary.

Since the nodal basis function can unique determine our numerical solution, the computed solutions at each nodal points are:

$$u_h(N_i) = \xi_i$$

where ξ_i are the entries that we computed from the stiffness matrix.

1.1 result presentation

Running the script:

```
\ fempoi.st.m
```

provides 3 outputs as follows:

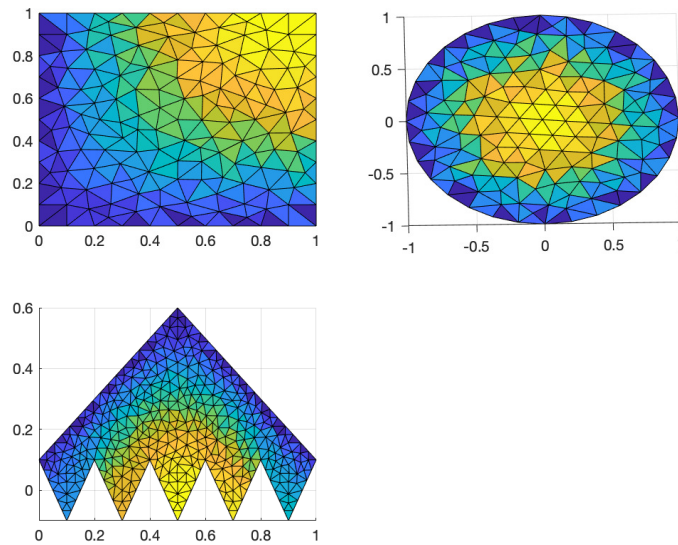


Figure 1: 2D color surface plot of results

Achieved in fempoi.m by solving with homogeneous Neumann condition first and replacing with Dirichlet conditions for mixed boundary problems.

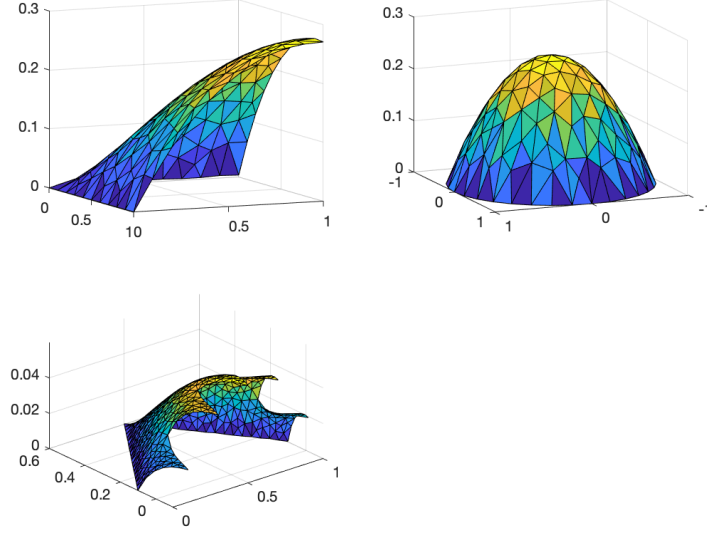


Figure 2: 3D color surface plot of results

2 Errors for All-Dirichlet Problem

Using the

`fempoi.m`

routine defined from part 1, we solve two Poisson problems using all-Dirichlet conditions on a square mesh and a polygon mesh. Using the finest level of refinement as an exact solution, we take max norm at each step and plot the log-log error behaviors. The error plot is presented as follows; we see that our convergence rate is roughly better than second order.

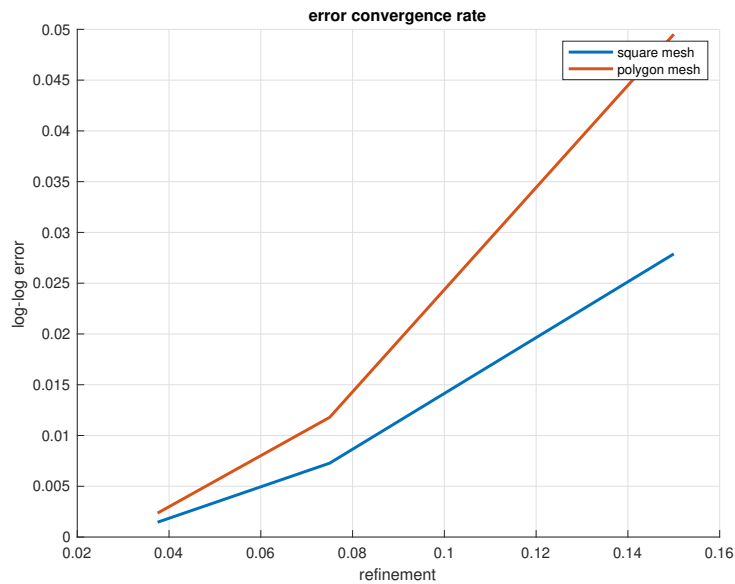


Figure 3: Error behaviors for two meshes, Poisson Problem 2D

Obtained through running the script:

```
poiconvtst.m
```

, which also provides the error and rate information as follows:

```
>> poiconvtst
% solving square mesh problem
errors =

    0.0279
    0.0073
    0.0015

displaying the convergence rate ...
=====

rate =

    2.3164
```

```

% solving polygon mesh problem
errors =

    0.0495
    0.0118
    0.0024

displaying the convergence rate ...
=====

rate =

    2.3219

>>

```

3 Fourth Order Boundary Value Problem

3.1 Galerkin Formulation

We would like to find $u_h \in V_h$ such that:

$$\int_0^1 u_h''(x)v''(x)dx = \int_0^1 f(x)v(x)dx$$

The Galerkin formulation is derived by using an appropriate functional space for $v \in V_h$ (keeping the boundary conditions $v(0) = v(1) = v'(0) = v'(1) = 0$) and applying integration by parts two times.

Starting from the PDE:

$$u''''(x) = f(x)$$

Multiply both sides by test function $v(x) \in V_h$, and integrate across the domain:

$$u''''(x)v(x) = f(x)v(x)$$

$$\int_0^1 u''''(x)v(x)dx = \int_0^1 f(x)v(x)dx$$

Let $u = v(x)$, $dv = u'''(x)dx$, we can integrate the left hand side of the PDE by parts.

$$\int_0^1 u'''(x)v(x)dx = v(x)u'''(x)\Big|_0^1 - \int_0^1 u'''(x)v'(x)dx$$

Since $v(0) = v(1) = 0$, we have:

$$\int_0^1 u'''(x)v(x)dx = - \int_0^1 u'''(x)v'(x)dx$$

Now we need to integrate the new expression by parts, let $u = v'(x)$, $dv = u'''(x)dx$, we apply integration by parts again:

$$-(\int_0^1 u'''(x)v'(x)dx) = -(v'(x)u'''(x)\Big|_0^1 - \int_0^1 u'''(x)v''(x)dx)$$

but $v'(0) = v'(1) = 0$ by our formulation, then we have finally the LHS:

$$\int_0^1 u'''(x)v(x)dx = -(-\int_0^1 u'''(x)v''(x)dx) = \int_0^1 u'''(x)v''(x)dx$$

which gives us the Galerkin formulation:

Find $u = u_h$ such that:

$$(u_h'', v'') = (f, v)$$

holds for all $v \in V_h$, or explicitly

$$\int_0^1 u_h''(x)v''(x)dx = \int_0^1 f(x)v(x)dx$$

3.2 Hermite Bases

In each element K_1, K_2 , we would like to have a degree 3 polynomial, and the overall basis function ϕ_1 and $\phi_2 \in V_h$ should match the values at 0 and 1 as well as their derivatives. Additionally, we also need the polynomial ϕ_i to have continuity for both itself and its first order derivative.

Since in each $K_i, i = 1, 2$, we would need a degree-3 polynomial, we would have 4 degrees of freedom in that element, and 8 degrees of freedom for ϕ_i

overall. Since we require $\phi(0) = \phi'(0) = \phi(1) = \phi'(1) = 0$, and continuity at both $\phi(\frac{1}{2}), \phi'(\frac{1}{2})$, we have that:

$$\dim V_h = 8 - 4 - 2 = 2$$

which means our functional basis has 2 degrees of freedom.

Define $\phi_1(x) \in V_h$ with the following conditions:

$$\begin{cases} \phi_1(0) = \phi'_1(0) = 0 \\ \phi_1(\frac{1}{2}) = 0 \\ \phi'_1(\frac{1}{2}) = 1 \\ \phi_1(1) = \phi'_1(1) = 0 \end{cases}$$

Let $\phi_1(x) = a_1x^3 + b_1x^2 + c_1x + d_1$ in K_1 and $\phi_1(x) = a_2x^3 + b_2x^2 + c_2x + d_2$ in K_2 , and we have the system for its coefficients:

$$\begin{cases} d_1 = 0 \\ c_1 = 0 \\ \frac{1}{8}a_1 + \frac{1}{4}b_1 + \frac{1}{2}c_1 + d_1 = 0 \\ \frac{3}{4}a_1 + b_1 + c_1 = 1 \end{cases}$$

Giving us the coefficients:

$$\begin{cases} a_1 = 4 \\ b_1 = -2 \\ c_1 = 0 \\ d_1 = 0 \end{cases}$$

Similarly in K_2 :

$$\begin{cases} \frac{1}{8}a_2 + \frac{1}{4}b_2 + \frac{1}{2}c_2 + d_2 = 0 \\ \frac{3}{4}a_2 + b_2 + c_2 = 1 \\ a_2 + b_2 + c_2 + d_2 = 0 \\ 3a_2 + 2b_2 + c_2 = 0 \end{cases}$$

$$\begin{cases} a_2 = 4 \\ b_2 = -10 \\ c_2 = 8 \\ d_2 = -2 \end{cases}$$

Then we have:

$$\phi_1(x) = \begin{cases} 4x^3 - 2x^2, & x \in K_1 \\ 4x^3 - 10x^2 + 8x - 2, & x \in K_2 \end{cases}$$

Similarly, we can define $\phi_2(x) \in V_h$ with the following conditions, altering the behavior at $\phi_2(\frac{1}{2})$ as long as it is different from ϕ_1 .

$$\begin{cases} \phi_2(0) = \phi_2'(0) = 0 \\ \phi_2(\frac{1}{2}) = 1 \\ \phi_2'(\frac{1}{2}) = 1 \\ \phi_2(1) = \phi_2'(1) = 0 \end{cases}$$

Let $\phi_2(x) = a_1'x^3 + b_1'x^2 + c_1'x + d_1'$ in K_1 and $\phi_2(x) = a_2'x^3 + b_2'x^2 + c_2'x + d_2'$ in K_2 , and we have the system for its coefficients:

$$\begin{cases} d_1' = 0 \\ c_1' = 0 \\ \frac{1}{8}a_1' + \frac{1}{4}b_1' + \frac{1}{2}c_1' + d_1' = 1 \\ \frac{3}{4}a_1' + b_1' + c_1' = 1 \end{cases}$$

Giving us the coefficients:

$$\begin{cases} a_1' = -12 \\ b_1' = 10 \\ c_1' = 0 \\ d_1' = 0 \end{cases}$$

In K_2 :

$$\begin{cases} \frac{1}{8}a_2' + \frac{1}{4}b_2' + \frac{1}{2}c_2' + d_2' = 1 \\ \frac{3}{4}a_2' + b_2' + c_2' = 1 \\ a_2' + b_2' + c_2' + d_2' = 0 \\ 3a_2' + 2b_2' + c_2' = 0 \end{cases}$$

$$\begin{cases} a_2' = 20 \\ b_2' = -46 \\ c_2' = 32 \\ d_2' = -6 \end{cases}$$

Then we have:

$$\phi_2(x) = \begin{cases} -12x^3 + 10x^2, & x \in K_1 \\ 20x^3 - 46x^2 + 32x - 6, & x \in K_2 \end{cases}$$

Therefore $\{\phi_1, \phi_2\}$ can serve as one basis for V_h .

3.3 Numerical solution

The exact solution is given by:

$$u^{(4)}(x) = 480x - 120$$

$$u'''(x) = 240x^2 - 120x + c_1$$

$$u''(x) = 80x^3 - 60x^2 + c_1x + c_2$$

$$u'(x) = 20x^4 - 20x^3 + \frac{1}{2}c_1x^2 + c_2x + c_3$$

$$u(x) = 4x^5 - 5x^4 + \frac{1}{6}c_1x^3 + \frac{1}{2}c_2x^2 + c_3x + c_4$$

Then plugging in the boundary conditions would give us:

$$u(0) = u(1) = u'(0) = u'(1) = 0$$

$$\begin{cases} c_4 = 0 \\ 4 - 5 + \frac{1}{6}c_1 + \frac{1}{2}c_2 + c_3 + c_4 = 0 \\ c_3 = 0 \\ 20 - 20 + \frac{1}{2}c_1 + c_2 + c_3 = 0 \end{cases}$$

solve, and we have:

$$\begin{cases} c_1 = -12 \\ c_2 = 6 \\ c_3 = c_4 = 0 \end{cases}$$

thus the final exact solution is given by:

$$u_{exact}(x) = 4x^5 - 5x^4 - 2x^3 + 3x^2$$

we can verify:

$$u'''' = 4 \cdot (5 \cdot 4 \cdot 3 \cdot 2)x - 5! = 480x - 120$$

Using the basis defined above, we test the numerical solution. First we begin by assembling the stiffness matrix and load vector.

Let $u_h = \xi_1\phi_1 + \xi_2\phi_2$. We would like to solve for the ξ vector.

Using the variational formulation, let $v(x) = \phi_j(x)$:

$$\begin{aligned} LHS &= \int_0^1 u_h''(x)v''(x)dx = \int_0^1 \sum_{i=1}^2 \xi_i \phi_i''(x) \phi_j''(x) dx \\ &= \sum_{i=1}^2 \xi_i \int_0^1 \phi_i''(x) \phi_j''(x) dx = \sum_{i=1}^2 \xi_i a_{ij} \end{aligned}$$

where $a_{ij} = \int \phi_i'' \phi_j'' dx$.

The load vector is:

$$RHS = \int_0^1 f(x)v(x)dx = \int_0^1 f(x)\phi_j(x)dx = b_j$$

Then concisely:

$$\begin{aligned} \sum_{i=1}^2 \xi_i a_{ij} &= b_j \\ A\xi &= b \end{aligned}$$

where $A \in \mathbb{R}^{2 \times 2}$.

Due to our functional space being small, we can explicitly calculate the stiffness matrix and load vector, which we do so in the numerical implementation. The main routine is presented in:

`prob3_beam.m`

which uses MATLAB symbolic computation to explicitly compute the functions composed of the functional basis $\phi_i \in V_h$, and then calculate our desired integrals.

The result is presented below:

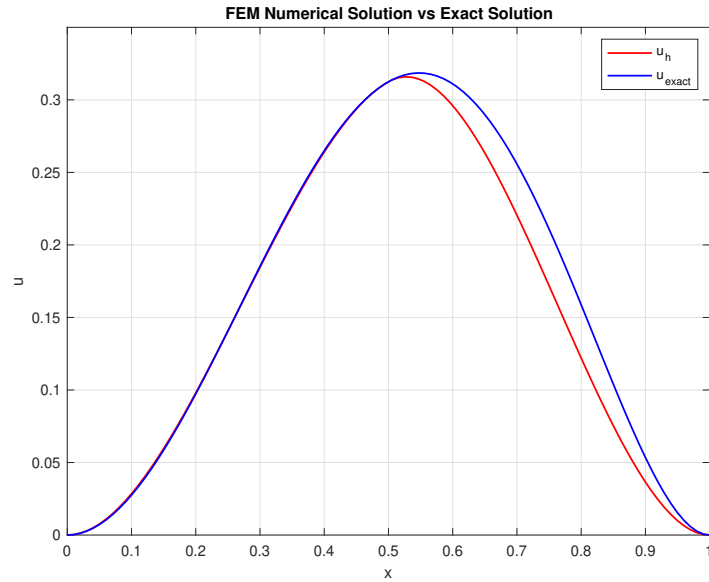


Figure 4: Numerical Solution versus Exact Solution for $u(x)$

We see that our FEM formulation with this V_h is able to capture the exact solution relatively well except for the second half. If we:

- (1) analytically integrate all integrals (to avoid numerical errors) and
 - (2) increase the number of elements,
- we should expect better convergence.