Problem Set 4

Math228B Numerical solutions to differential equations

Håkon Olav Torvik

UC Berkeley

(Dated: March 18, 2022)

Part 1

Considering the bvp

$$u''''(x) = f(x) \equiv 480x - 120, \quad x \in (0,1) \tag{1}$$

$$u(0) = u'(0) = u(1) = u'(1) = 0. \tag{2}$$

It is straight forward to find and verify that the analytical solution to (1)-(2) is

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

a

A weighted average over the interval, for the approximate solution $u_h(x) \in V_h$, and for some function $v(x)$, is written as

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

Simplifying by integrating by parts:

$$\begin{aligned}\int_0^1 u_h''''(x)v(x)dx &= u_h'''(x)v(x)\Big|_0^1 - \int_0^1 u_h'''(x)v'(x)dx \\ &= \int_0^1 u_h''(x)v''(x)dx + \Big[u_h'''(x)v(x) - u_h''(x)v'(x)\Big]_0^1.\end{aligned}$$

By using the Galerkin method, we choose the function $v(x)$ to be from the same function space as $u_h(x)$, such that the boundary conditions (2) are also imposed on $v(x)$. The last term above will then be 0, and we end up the problem being to find $u_h \in V_h$, such that

$$\int_0^1 u_h''(x)v''(x)dx = \int_0^1 f(x)v(x)dx, \quad \forall v \in V_h \tag{3}$$

b

Want to find a basis $\{\varphi_i\}$ that span

$$V_h = \{v \in C^1([0,1]) : v|_K \in \mathcal{P}_3(K) \forall K \in T_h, v(0) = v'(0) = v(1) = v'(1) = 0\}.$$

That is, we want a cubic on each of the 2 elements $K_i \in T_h = \{K_1, K_2\}$, giving 8 degrees of freedom. The 4 boundary conditions represents 4 constrains, while the continuity conditions C^1 are 2 further constraints, such that $\dim(V_h) = 2$.

To ensure orthogonality of the basis, I want basis functions that satisfy $\varphi_i^{(j)}(x) = \delta_{ij}, i,j = 0,1$, where $x = \frac{1}{2}$ is the only point in the domain where I'm free to choose any value of of the basis functions, and $f^{(k)}$ represents the kth derivative. I know that the cubic Hermite interpolating polynomials have the following properties:

$$\begin{aligned}
H_0(0) &= 1 & \tilde{H}_0(0) &= H_1(0) = \tilde{H}_1(0) = 0 \\
\tilde{H}_0'(0) &= 1 & H_0'(0) &= H_1'(0) = \tilde{H}_1'(0) = 0 \\
H_1(1) &= 1 & H_0(1) &= \tilde{H}_0(1) = \tilde{H}_1(1) = 0 \\
\tilde{H}_1'(1) &= 1 & H_0'(1) &= \tilde{H}_0'(1) = H_1'(1) = 0
\end{aligned}$$

Then will $\{\varphi_0, \varphi_1\}$ form a basis for V_h, where

$$\varphi_0 = \begin{cases} H_1(2x) & 0 \le x \le \frac{1}{2} \\ H_0(2x-1) & \frac{1}{2} < x \le 1, \end{cases}$$

$$\varphi_1 = \begin{cases} \tilde{H}_1(2x) & 0 \le x \le \frac{1}{2} \\ \tilde{H}_0(2x-1) & \frac{1}{2} < x \le 1. \end{cases}$$

c

First I write the explicit expressions for the cubic Hermite polynomials and its derivatives for reference later.

$$
\begin{aligned}
&H_0(x) = 2x^3 - 3x^2 + 1 && H_0'(x) = 6x^2 - 6x && H_0''(x) = 12x - 6 \\
&H_1(x) = 3x^2 - 2x^3 && H_1'(x) = 6x - 6x^2 && H_1''(x) = 6 - 12x \\
&\tilde{H}_0(x) = x^3 - 2x^2 + x && \tilde{H}_0'(x) = 3x^2 - 4x + 1 && \tilde{H}_0''(x) = 6x - 4 \\
&\tilde{H}_1(x) = x^3 - x^2 && \tilde{H}_1'(x) = 3x^2 - 2x && \tilde{H}_1''(x) = 6x - 2
\end{aligned}
$$

From (3), the solution u to (1)-(2) can be written

$$
\begin{aligned}
&Au = b, \\
&A_{ij} = \int_0^1 \varphi_i''(x)\varphi_j''(x)dx, \quad b_i = \int_0^1 f(x)\varphi_i(x)dx
\end{aligned}
$$

To simplify integrals, I do the following change of variables.

$$
\begin{aligned}
\int_0^{\frac{1}{2}} f(2x)dx &= \frac{1}{2}\int_0^1 f(x)dx \\
\int_{\frac{1}{2}}^1 f(2x-1)dx &= \frac{1}{2}\int_0^1 f(x)dx
\end{aligned}
$$

Staring to integrate.

$$\begin{aligned}
A_{11} &= \int_0^{\frac{1}{2}} H_1''(2x)^2 dx + \int_{\frac{1}{2}}^1 H_0''(2x-1)^2 dx \\
&= \frac{1}{2}\int_0^1 H_1''(x)^2 dx + \frac{1}{2}\int_0^1 H_0''(x)^2 dx \\
&= 12 \\
A_{22} &= \int_0^{\frac{1}{2}} \tilde{H}_1''(2x)^2 dx + \int_{\frac{1}{2}}^1 \tilde{H}_0''(2x-1)^2 dx \\
&= \frac{1}{2}\int_0^1 \tilde{H}_1''(x)^2 + \frac{1}{2}\int_0^1 \tilde{H}_0''(x)^2 dx \\
&= 4 \\
A_{12} = A_{21} &= \int_0^{\frac{1}{2}} H_1''(2x)\tilde{H}_1''(2x) dx + \int_{\frac{1}{2}}^1 H_0''(2x-1)\tilde{H}_0''(2x-1) dx \\
&= \frac{1}{2}\int_0^1 H_1''(x)\tilde{H}_1''(x) dx + \frac{1}{2}\int_0^1 H_0''(x)\tilde{H}_0''(x) dx \\
&= 0 \\
A &= \begin{pmatrix} 12 & 0 \\ 0 & 4 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
b_1 &= \int_0^{\frac{1}{2}} f(x)H_1(2x) dx + \int_{\frac{1}{2}}^1 f(x)H_0(2x-1) dx \\
&= \frac{1}{2}\int_0^1 f\left(\frac{x}{2}\right) H_1(x) dx + \frac{1}{2}\int_0^1 f\left(\frac{x+1}{2}\right) H_0(x) dx \\
&= 60 \\
b_2 &= \int_0^{\frac{1}{2}} f(x)\tilde{H}_1(2x) dx + \int_{\frac{1}{2}}^1 f(x)\tilde{H}_0(2x-1) dx \\
&= \frac{1}{2}\int_0^1 f\left(\frac{x}{2}\right) \tilde{H}_1(x) dx + \frac{1}{2}\int_0^1 f\left(\frac{x+1}{2}\right) \tilde{H}_0(x) dx \\
&= 8 \\
b &= \begin{pmatrix} 60 \\ 8 \end{pmatrix}
\end{aligned}$$

If I put this together to solve for u, I get that $u = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$. Using these as coefficient for the

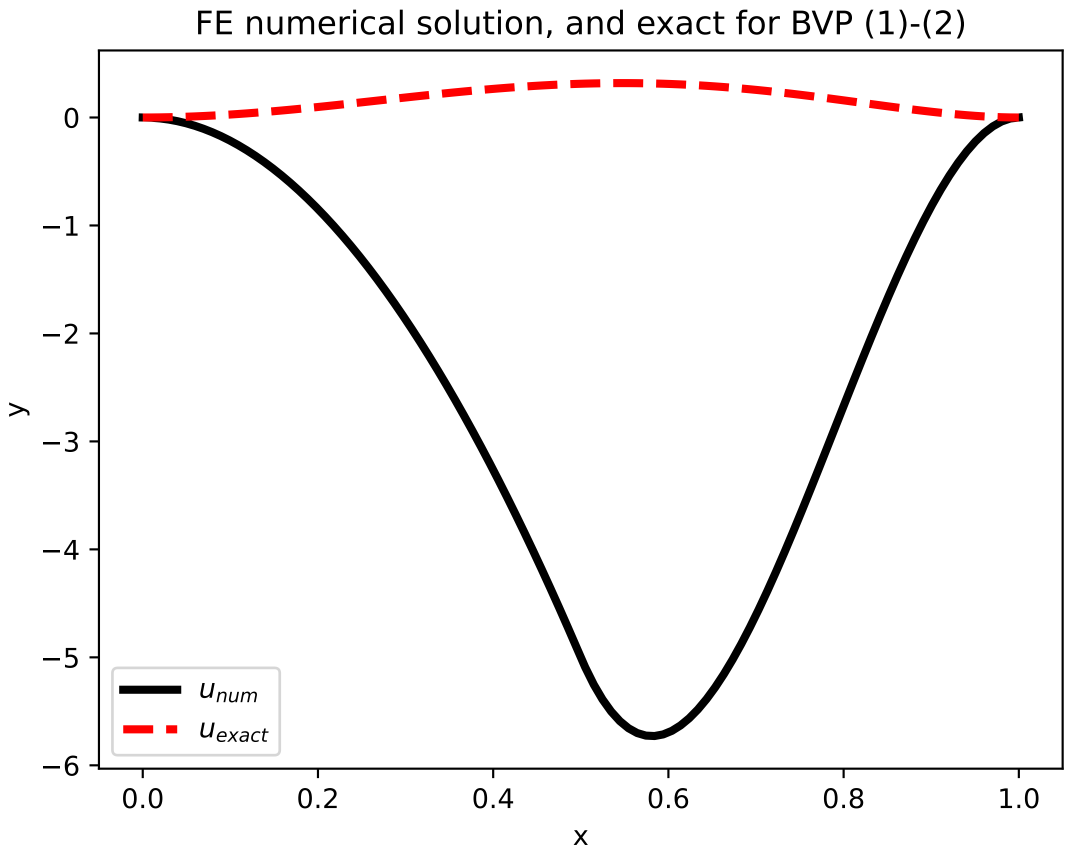

Figure 1: Numerical and exact solution. I obviously have found the wrong coefficients.

basis functions I plot the numerical solution, along with the exact solution, in Figure 1. My solution is obviously wrong, but due to time constraints I wasnt able to correct it. I think what I should have done was to not integrate over the whole domain, rather each element seperately, and get a 4×4 matrix A, such that the solution u has 4 coefficients, the first two being the coefficient for each basis function on K_1, and the second to on K_2. Because of the continuity constraint, these should be the same however, which is why I did not do this first.

Part 2

Using the stamping method as presented in the lecture notes, I write a function `fempoi()` for solving Poissons equation $-\nabla^2 u = 1$ on any unstructured, triangulated mesh. Homogeneous dirichlet conditions are imposed on the edges, while homogeneous Neumann conditions

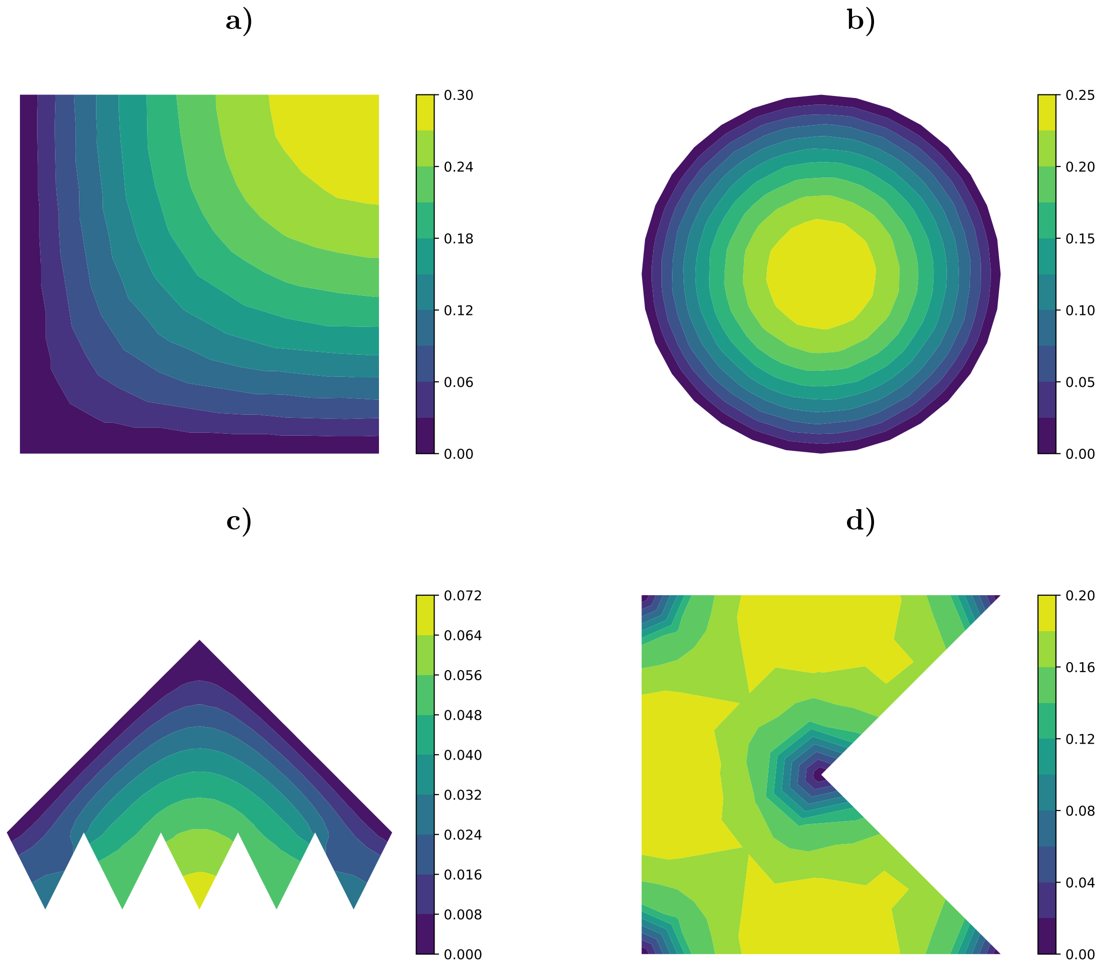

Figure 2: Solution to poissions equation on each respective polygon, with a mix of dirichlet and neumann conditions. **a)**: Square, dirichlet on left and lower boundary. **b)**: circle, dirichlet on entire boundary. **c)**: jagged polygon, dirichlet on top. **d)**: Square pac-man, dirichlet on the 5 corners.

naturally arise elsewhere.

Using this function, I recreate the 3 figures in the project description, in as part **a), b)** and **c)** in Figure 2. They all share a great resembelance with their counterpart. For figure-balance I also include a solution on the square pac-man polygon from PS3, with dirichlet only on the 5 corners and Neumann along the edges, with methematically doesn't really make sense, but looks cool.

None of the meshes are refined. While testing, I found a bug in my meshgenerator from PS3, due to a misunderstainding of when to remove dengenerate triangles. This has been fixed in the code for this project.

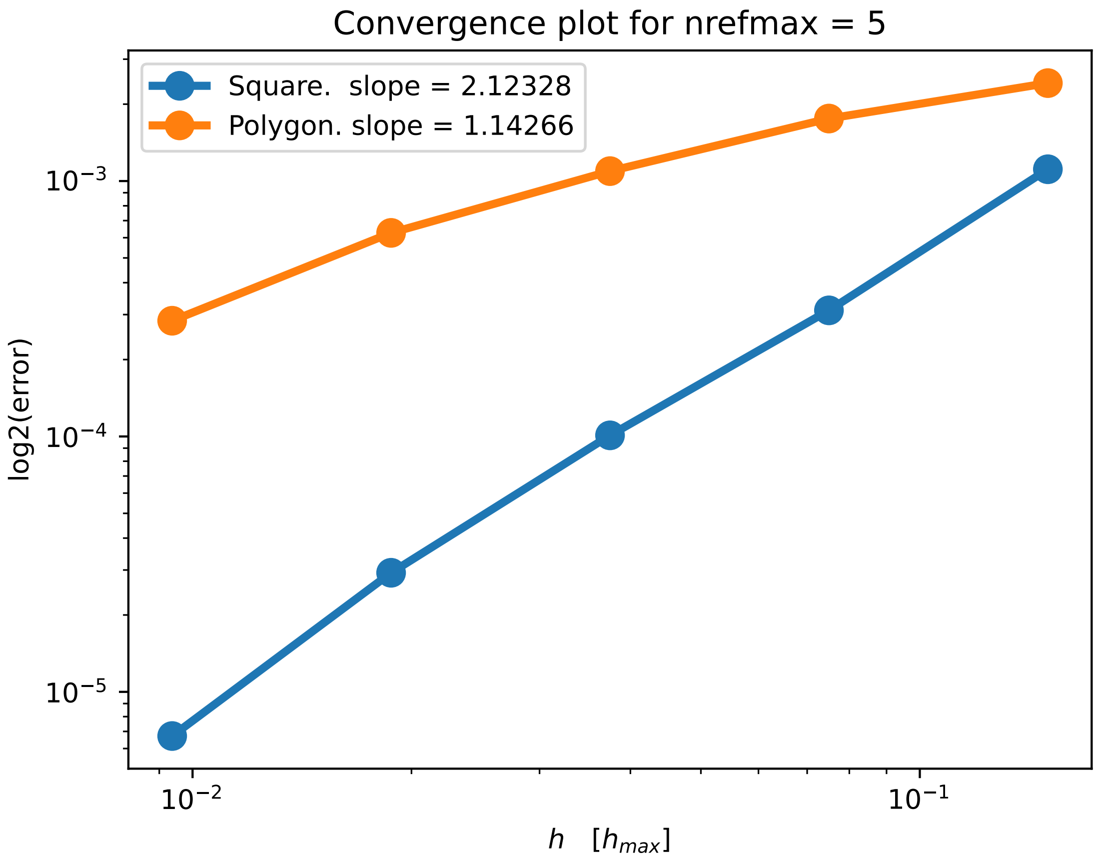

Figure 3: Convergence plot with rates for 2 different polygons, as function of refinement.

Part 3

To determine the convergence rate for the function `fempoi()`, I write a function `poiconv()` which calls `fempoi()` on a more and more refined mesh of the same polygon, and finds the max-norm of the difference between each refinement and the last.

To test this I use a square and a square pac-man (from PS3). The resulting convergence plot can be seen in Figure 3, where the true solution is taken as the 5th refinement of the mesh. From reading off the slope, we see that for the square the error goes as $\mathcal{O}(h^{2.12})$, while for the pac-man it goes as $\mathcal{O}(h^{1.14})$. The very same method is 2nd order accurate for one shape, while only 1st order accurate for another! This is because the constant in the error-bound is dependent on the angles of the triangles, and can be very large. This is the case for the square pac-man, and thus the method converges slower.