Homework 1

January 26th 2021

Deadline: Feb 18th 2021, the answer to the questions will be submitted via Canvas including the code. I should be able to run the code and generate any plot that you used on your answers. Your code will be graded for efficiency, it should run using resources equivalent to a mid-range modern laptop. You can submit your answers in a jupyter notebook, if that is more convenient for you. No late homework will be accepted.

Rules: You are strongly encouraged to discuss the homework with your peers, in particular, piazza is a very good environment for discussing the homework. However, you need to write your own homework and you need disclose you sources. If you work with some of your classmates you need to disclose your collaborators.

Disclaimer: If you are not used to Python several concepts may not be familiar to you. You may want to check the online tutorials shared in the announcements, including the documentation of numpy and scipy (they will be your best friends). Otherwise you can post in piazza, you are encouraged to share code snippets but not a full answer. The same applies if your code is too slow, i.e., I strongly encourage you to post in piazza and interact with your classmates.

A (50 points) We will implement a basic finite element method in 1D.

(a) Implement a Mesh class (following PEP8 class names follow the Camel convention) following the convention in the file. Implement a constructor that takes an np.array containing the subdivisions of the partition.

(b) Implement the V_h class representing the space of continuous piece-wise line polynomials following the structure in the `.py` file. Implement an `eval` function. The input of this function will be the coefficients using the nodal basis $\{\xi_i\}_{i=0}^n$, and an evaluation point x. Then you need to output the associated function evaluated at point x, i.e., $\sum \xi_i \phi_i(x)$.

(c) Implement a Function class implementing an instance of V_h, and implement a `__init__` function. This function should evaluate the function at any point in the interval. You may want to leverage the sorted nature of the points in the mesh. This would allows to "evaluate" a member of the function class, i.e., `u(x)`.

Hint: this is the same functionality than before, so you may want to reuse it, by calling the function you coded above.

(d) Implement the construction of the mass matrix, You want to loop over the intervals and build your matrix using Algorithm 2. You can use Eq. 1.74 to check that your matrix is properly built. In order to obtain a relatively fast construction use a `lil_matrix`.
Hint: see https://docs.scipy.org/doc/scipy/reference/sparse.html.

(e) Implement the assembling of the load vector following Algorithm 3, following the structure in the `.py` file. Use a quadrature of your choice, and explain why you chose that. What are the possible drawbacks of using a quadrature?.

(f) Implement the projection function following the structure in the `.py` file.

(g) Implement a function that computes the stiffness matrix in 1D following algorithm 5. In this case we will use only Dirichlet boundary conditions, so you may simplify the matrix.

(h) Implement the assembly of the source, we are only using Dirichlet boundary condition, thus we only care about the interior degrees of freedom.

(i) Implement a solver for the poisson equation with Dirichlet boundary conditions.

B (10 points) Using your code developed above, showcase the property of the L^2 projection:

$$\|f - P_h f\|^2_{L^2(I)} \leq C \sum_{i=1}^{n} h_i^4 \left\|f''\right\|^2_{L^2(I_i)} \tag{1}$$

Here you have two properties to showcase. The convergence with respect to h_i and the depedence on f''. For the first case you can pick an f, and refine the partition to plot a graph showing the correct scaling. For the second case, you can use a family of function whose second derivate increases (e.g., $sin(2k\pi)$) and plot the error.

Find a counter-example in which such as scaling does not hold. Explain why you are observing this issue.

C (10 points) In this question you will use your code to solve the two-point boundary value

$$\begin{gathered} -u'' = f, \quad x \in I = [0, L] \\ u(0) = u(L) = 0 \end{gathered} \tag{2}$$

(a) Choose an f such that you can showcase that the estimate

$$\left\|(u - u_h)'\right\|^2_{L^2(I)} \leq C \sum_{i=1}^{n} h_i^2 \left\|u''\right\|^2_{L^2(I_i)} \tag{3}$$

holds. (You need to show that this is true by decreasing h, for simplicity you can use equispace grid)

(b) Suppose that $f = e^{-\frac{(x-L/2)^2}{2\sigma}}$, with $\sigma = 0.01$ Using a dyadic refinement based on the element residual, find an adapted grid in which the error is below 10^{-3}. How many intervals do you need to achieve this error? How does it compare with using a regular mesh?

D (10 points) Let u be defined on $I = [0, 1]$ and such that $u(0) = 0$. Prove the so-called Poincaré inequality

$$\|u\|_{L^2(I)} \leq C \left\|u'\right\|_{L^2(I)}$$

E (20 points) Consider the problem

$$\begin{gathered} -u'' + u = f, \quad x \in I = [0, 1] \\ u(0) = u(1) = 0 \end{gathered}$$

(a) Choose a suitable finite element space V_h.

(b) Formulate a finite element method.

(c) Derive the discrete system of equations.

(d) Derive a posteriory estimate for the problem.

(e) (10 Bonus points) Modify your code in the first part to solve this problem.

(f) (10 Bonus points) Choose an f and show that your estimate above holds using a graph of the error while refining the grid.