Homework 8. Due Wednesday, April 9

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. **(6 pts)** The goal of this problem is to show that the FEM solution of the Poisson equation is the linear interpolant of the exact solution. We will examine only the 1D case. This problem is a composite of problems 2.2a, 2.4, and 5.4 from [2].

   Consider the BVP

$$-u'' = f(x), \quad 0 < x < 1, \quad u(0) = u(1) = 0, \tag{1}$$

   where $f(x)$ is a given function integrable on $[0, 1]$.

   (a) Verify that the solution of Eq. (1) is given by

$$u(x) = \int_0^1 G(x, y)f(y)dy, \tag{2}$$

   where $G(x, y)$ is Green's function defined in $[0, 1]^2$ by

$$G(x, y) = \begin{cases} (1 - x)y, & 0 \le y \le x \le 1, \\ x(1 - y), & 0 \le x \le y \le 1. \end{cases} \tag{3}$$

   (b) Verify that for any function $v(x) \in H_0^1([0, 1])$, the following identity holds:

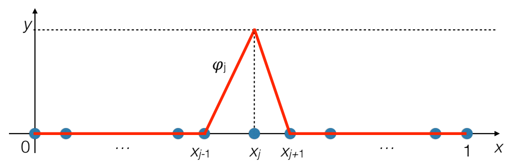
$$\int_0^1 v'(y)\frac{d}{dy}G(x, y)dy = v(x). \tag{4}$$

   Conclude that since the exact solution u of Eq. (1) belongs to $H_0^1([0, 1])$, Eq. (4) holds for u for all $x \in [0, 1]$.

   (c) The linear interpolant of the exact solution $u(x)$, denoted by $I_h u$, is a function that is equal to u at the nodes

$$0 \equiv x_0 < x_1 < \ldots < x_n < x_{n+1} \equiv 1,$$

   and is linear in each interval $[x_j, x_{j+1}]$, $j = 0, 1, \ldots, n$. (We do not assume that the nodes are equispaced.) Write $I_h u$ as a linear combination of the basis functions shown in fig. below.



(d) Show that the set of functions $\psi_j(y) := G(x_j, y)$ is a basis in $S_h(x) = \text{span}\{\phi_1, \dots, \phi_n\}$.

(e) Show that for the linear interpolant $I_h u$,

$$\int_0^1 [I_h u]'(y) \frac{d}{dy} G(x_j, y) dy = u(x_j). \tag{5}$$

(f) Write out the definition of the FEM solution. Prove, using all the facts above that the FEM solution

$$U^{FEM}(x) = \sum_{j=1}^{n} U_j^{FEM} \phi_j(x)$$

coincides with $I_h u$.

2. **(6 pts)** This problem suggests a way to modify a non-self-adjoint elliptic equation to make it amenable for solving using FEM.

(a) Let Ω be a compact domain in 2D. The boundary of Ω can be decomposed as $\partial\Omega = \Gamma_D \cup \Gamma_N$. Consider the following boundary-value problem:

$$\begin{cases} \beta^{-1}\Delta u + b(x) \cdot \nabla u = 0, & x \in \Omega \\ u = u_D & x \in \Gamma_D \\ \frac{\partial u}{\partial n} = g, & x \in \Gamma_N. \end{cases} \tag{6}$$

Let b be a smooth vector field and β be a constant. Suppose it can be decomposed into curl-free and divergence-free components:

$$b = -\nabla V(x) + f(x). \tag{7}$$

The curl-free component is ∇V and the divergence-free component is f: $\nabla \cdot f = 0$. Show that BVP (6) is equivalent to

$$\begin{cases} \nabla \cdot \left(e^{-\beta V(x)} \nabla u\right) + \beta e^{-\beta V(x)} f(x) \cdot \nabla u = 0, & x \in \Omega \\ u = u_D & x \in \Gamma_D \\ \frac{\partial u}{\partial n} = g, & x \in \Gamma_N. \end{cases} \tag{8}$$

(b) Consider the Maier-Stein drift field

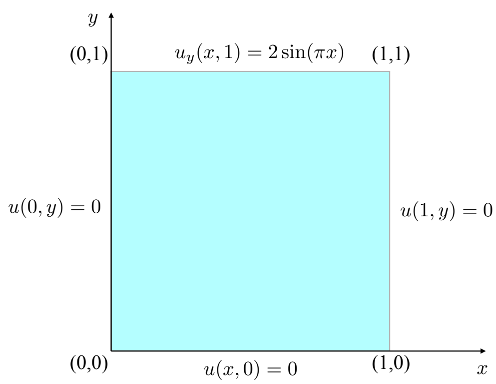
$$b(x,y) = \begin{bmatrix} x - x^3 - 10xy^2 \\ -(1+x^2)y \end{bmatrix}. \tag{9}$$

Decompose $b(x,y)$ into divergence-free and curl-free components. Find the potential function $V(x,y)$.

3. **(10 pts)** Set up and solve Problem 6 from Lagaris, Likas, and Fotiadis (1998), a boundary-value problem for the Poisson equation

$$u_{xx} + u_{yy} = f(x,y) \quad \text{where} \quad f(x,y) = (2-\pi^2 y^2)\sin(\pi x), \quad (x,y) \in [0,1]^2, \tag{10}$$

with mixed boundary conditions:



The exact solution to this problem is given by

$$u_{ex}(x,y) = y^2 \sin(\pi x). \tag{11}$$

To solve this problem, you can mimic the provided codes for Problem 5 from Lagaris et al.:

- `Lagaris5.zip`, a Matlab package written by me from scratch, or
- `Lagaris5.ipynb`, a Python code written by me from scratch, or
- `Lagaris_Margot.ipynb`, a Python written by Margot Yuan (AMSC) with the use of automatic differentiation and built-in Adam optimizer.

Use a neural network with one hidden layer and $n = 10$ neurons of the form:

$$\mathcal{N}(x,y;w) = \sum_{j=0}^{n-1} w_{3j}\sigma\left(w_{0j}x + w_{1j}y + w_{2j}\right), \tag{12}$$

where $\sigma(z) = \tanh(z)$ acts entrywise. The total number of parameters to optimize is $4n$.

The proposed form of the solution $U(x, y; w)$ is given in Lagaris et al. in Eqs. (24)–(25). Take $N_{tr} = 49$ training points forming a uniform 7×7 `meshgrid` in $[0, 1]^2$ (set a 9×9 meshgrid and strip its boundaries). Set up the least squares loss function

$$\mathcal{L}(w) = \frac{1}{2} \sum_{i=0}^{N_{tr}-1} \left| U_{xx}(x_i, y_i; w) + U_{yy}(x_i, y_i; w) - f(x_i, y_i) \right|^2 . \tag{13}$$

Plot the computed solution and the numerical error using level sets or a heatmap. Plot the loss function versus the iteration number using a log scale for the y-axis.

References

[1] Jochen Alberty, Carsten Carstensen and Stefan A. Funken, Remarks around 50 lines of Matlab: short finite element implementation

[2] S. Larsson and V. Thomee, Partial Differential Equations with Numerical Methods, Springer-Verlag Berlin Heidelberg, 2003, 2009 (soft cover)