1. The Poisson problem

The Poisson equation is written as

$$\Delta\phi = \nabla \cdot \nabla\phi = -f,$$

where ϕ and f are two scalar functions. We can write it in a mixed formulation if we introduce $\boldsymbol{u} = \nabla\phi$:

$$\begin{aligned} \boldsymbol{u} &= \nabla\phi \ , \\ \nabla \cdot \boldsymbol{u} &= -f \ . \end{aligned} \tag{1}$$

An engineering problem is usually given as: In a given domain Ω, f is known all over the Ω, ϕ is known on the boundary $\partial\Omega$, find ϕ and $\boldsymbol{u}$ that solve the problem (1).

To solve such a problem, we first weaken it by saying that: If ϕ and $\boldsymbol{u}$ that solve

$$\begin{aligned} \int_\Omega \boldsymbol{v}^\mathsf{T} \cdot \boldsymbol{u} - \int_\Omega \boldsymbol{v}^\mathsf{T} \cdot \nabla\phi &= 0 \ , \\ \int_\Omega (\nabla \cdot \boldsymbol{u})\, g &= -\int_\Omega fg \ , \end{aligned} \tag{2}$$

for all $\boldsymbol{v}$ and g, ϕ and $\boldsymbol{u}$ solve the problem (1). If we perform integration by parts to the term $\int_\Omega \boldsymbol{v}^\mathsf{T} \cdot \nabla\phi$, we eventually get a problem given by: Find ϕ and $\boldsymbol{u}$, such that

$$\begin{aligned} \int_\Omega \boldsymbol{v}^\mathsf{T} \cdot \boldsymbol{u} + \int_\Omega (\nabla \cdot \boldsymbol{v})\, \phi &= \int_{\partial\Omega} \phi\, \boldsymbol{v} \cdot \boldsymbol{n} \ , \\ \int_\Omega (\nabla \cdot \boldsymbol{u})\, g &= -\int_\Omega fg \ , \end{aligned} \tag{3}$$

for all $\boldsymbol{v}$ and g. The next question is how to discretize this problem. We will restrict ourselves to $\mathbb{R}$ at this moment.

2. 1D

We consider the domain $\Omega = x \in [-1, 1]$, and

$$f = \alpha^2\pi^2 \sin(\alpha\pi x),$$

where $\alpha \in \{1, 2, 3, 4, \cdots\}$ is a given factor (choose a value as you wish, start with a low one, like $\alpha = 1$). The boundary condition for ϕ is given as $\phi(-1) = \phi(1) = 0$.

In $\mathbb{R}$, a vector, for example $\boldsymbol{u}$, is of no difference with a scalar, like ϕ. So we write (3) as

$$\begin{aligned} \int_\Omega vu + \int_\Omega (\nabla \cdot v)\, \phi &= 0 \ , \\ \int_\Omega (\nabla \cdot v)\, g &= -\int_\Omega fg \ . \end{aligned} \tag{4}$$

We let ϕ be discretized as

$$\phi^h = \sum_{i=1}^{N} \phi_i e_i(x). \tag{5}$$

Let u be discretized as

$$u^h = \sum_{i=0}^{N} u_i h_i(x). \tag{6}$$

If we choose the test functions g to be $e_i(x)$, $i = 1, 2, \cdots, N$, and choose the test functions v to be $h_i(x)$, $i = 0, 1, 2, \cdots, N$. we can get the following discrete system:

$$\begin{Bmatrix} \mathbb{M}^u & (\mathbb{M}^\phi \mathbb{E})^\mathsf{T} \\ \mathbb{M}^\phi \mathbb{E} & \end{Bmatrix} \begin{Bmatrix} \vec{u} \\ \vec{\phi} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -\vec{f} \end{Bmatrix}, \tag{7}$$

where $\vec{u} = \{u_0, u_1, \cdots, u_N\}^\mathsf{T}$ and $\vec{\phi} = \{\phi_1, \phi_2, \cdots, \phi_N\}^\mathsf{T}$ are the vectors of the expansion coefficients (degrees of freedom), and

$$\vec{f_i} = \int_{-1}^{1} f e_i(x) \,\mathrm{d}x,$$

and

$$\mathbb{M}^u_{i,j} = \int_{-1}^{1} h_i(x) h_j(x) \,\mathrm{d}x$$

is the mass matrix of shape $(N+1, N+1)$ regarding basis functions h, and

$$\mathbb{M}^\phi_{i,j} = \int_{-1}^{1} e_i(x) e_j(x) \,\mathrm{d}x$$

is the mass matrix of shape (N, N) regarding basis functions e, and

$$\mathbb{E} = \begin{Bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{Bmatrix}$$

is the incidence matrix of shape $(N, N+1)$ which basically does the derivative.

If we solve the system (7), for example, using 'numpy.linalg.solve', we can obtain $\vec{u}$ and $\vec{\phi}$. Therefore we can get the solutions ϕ^h and u^h, see (5) and (6).

Now, you can play around, by changing N and α, to see how the solutions look like.