Brief Introduction to Finite Element Method

June 23, 2025

0.1 Strong Form vs. Weak Form of PDE

0.1.1 The Strong Form

The Poisson equation in one dimension is given by:

$$-k\frac{\partial^2 u}{\partial x^2} = f(x), \tag{1}$$

where k is a given coefficient, u is the unknown function, and f is a known source function.

0.1.2 The Weak Form

To derive the weak form, we introduce a **test function** w from the function space $H^1(\Omega)$. Multiplying both sides of the equation by w and integrating over the domain Ω gives:

$$-\int_\Omega k\frac{\partial^2 u}{\partial x^2} w\,dx = \int_\Omega fw\,dx \tag{2}$$

Applying integration by parts:

$$\int_a^b h'g = [hg]_a^b - \int_a^b hg'$$

we obtain the weak form:

$$\int_\Omega k\frac{\partial u}{\partial x}\cdot\frac{\partial w}{\partial x}\,dx - \left[w\frac{\partial u}{\partial x}\right]_{x_0}^{x_1} = \int_\Omega fw\,dx \tag{3}$$

By choosing test functions that vanish on the boundary (i.e., $w \in H_0^1(\Omega)$), the boundary term disappears, leading to:

$$\int_\Omega k\frac{\partial u}{\partial x}\cdot\frac{\partial w}{\partial x}\,dx = \int_\Omega fw\,dx, \tag{4}$$

where the term in the square brackets has vanished as the test function w has zero values on the boundary Ω.

0.2 The Finite Element Method

0.2.1 Discretization of the Weak Formulation

The finite element method (FEM) is a powerful approach that generalizes easily to higher dimensions and complex geometries. In the 1D case, we discretize the domain Ω into n nodes (the first node gets index 0, and the last node is $n-1$) and approximate the unknown function u using a finite-dimensional subspace. The function u_h is the notation of the discrete function u:

$$u_h = \sum_{i=0}^{n-1} u_i \cdot \psi_i(x), \tag{5}$$

where u_i are unknown coefficients and ψ_i are piecewise linear basis functions. For ψ_i in this text, simple linear interpolation is used, although higher interpolation schemes are available. The weighting function is called the **shape function** (or **base function**) and is defined as:

$$\psi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}}, & x_{i-1} \leq x \leq x_i, \\ \frac{x_{i+1}-x}{x_{i+1}-x_i}, & x_i \leq x \leq x_{i+1}, \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

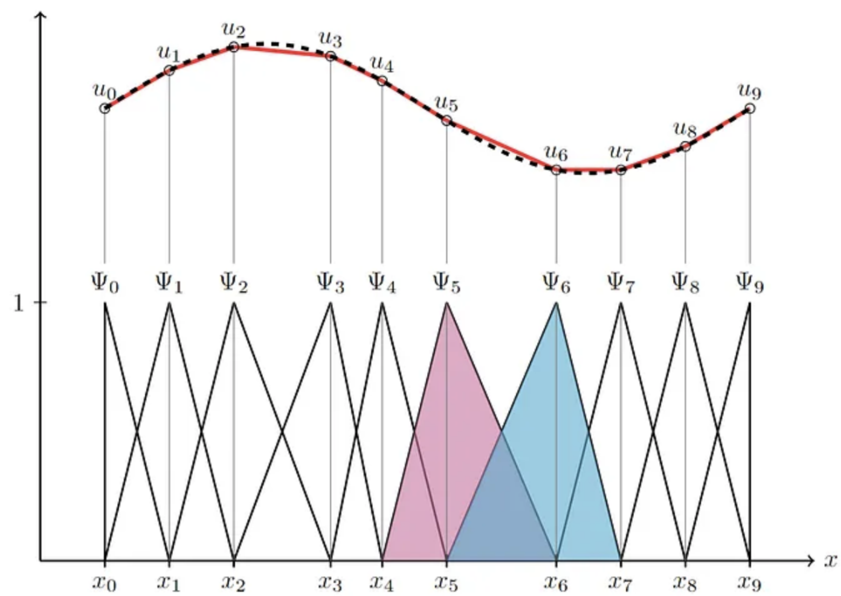

Figure 1: Discretization of the function u using shape functions (from https://medium.com/)

0.2.2 Constructing the Stiffness Matrix and Formulating the Linear System

The Galerkin method suggests that the test function w will be the same as the shape function ψ_i. We insert $u = u_h$ and w into the weak form (4), which leads to the following system:

$$\int_\Omega k \left(\sum_{j=0}^{n-1} u_j \frac{\partial \psi_j}{\partial x} \right) \frac{\partial \psi_i}{\partial x} \, dx = \int_\Omega f(x) \psi_i \, dx$$

The u_j is a scalar and can be pulled out of the derivative. When multiplying the brackets, the equation can also be written as:

$$\sum_{j=0}^{n-1} \underbrace{\left(\int_\Omega k \frac{\partial \psi_j}{\partial x} \frac{\partial \psi_i}{\partial x} \, dx \right)}_{A_{ij}} u_j = \underbrace{\int_\Omega f(x) \psi_i \, dx}_{b_i} \tag{7}$$

This results in the linear system:

$$A\vec{u} = b, \tag{8}$$

where $\vec{u} = [u_0, u_1, \ldots, u_{n-1}]^T$. The matrix A is usually called the **stiffness matrix** and the vector b is called the **force vector**. The entries of A and b are given by:

$$
\begin{aligned}
A_{ij} &= \int_\Omega k \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} \, dx, \\
b_i &= \int_\Omega f \psi_i \, dx.
\end{aligned}
$$

We note that the shape function ψ_i is only non-zero between x_{i-1} and x_{i+1}. Therefore, only the multiplications of two neighboring shape functions from the left, x_{i-1}, and right, x_{i+1}, or itself, x_i, will result in a non-zero value. This leads to the matrix A with the following form:

$$
A_{ij} = \begin{cases}
\frac{k}{x_i - x_{i-1}} + \frac{k}{x_{i+1} - x_i}, & i = j \\
-\frac{k}{x_i - x_{i-1}}, & j = i - 1 \\
-\frac{k}{x_{i+1} - x_i}, & j = i + 1 \\
0, & \text{otherwise}
\end{cases}
$$

for all $i = 1, \ldots, n-1$. For $i = 0, n$, we have:

$$
A_{0,0} = \frac{1}{x_1 - x_0}, \quad A_{n,n} = \frac{1}{x_n - x_{n-1}}.
$$

For a constant source term $f(x) = 1$, we have:

$$
b_i = \frac{x_{i+1} - x_i}{2} + \frac{x_i - x_{i-1}}{2}, \quad i = 1, \ldots, n-1
$$

and

$$
b_0 = \frac{x_1 - x_0}{2}, \quad b_n = \frac{x_n - x_{n-1}}{2}.
$$

Putting everything together in a matrix leads to a matrix $Au = b$ equation system, as shown below. The matrix A is a square matrix with dimensions equal to the number of nodes, and b is a vector with the same length:

$$
\begin{bmatrix}
A_{0,0} & A_{0,1} & 0 & \cdots & \cdots & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \cdots & \cdots & \vdots \\
\cdots & \cdots & -\frac{k}{x_i - x_{i-1}} & \left(\frac{k}{x_i - x_{i-1}} + \frac{k}{x_{i+1} - x_i} \right) & -\frac{k}{x_{i+1} - x_i} & \cdots & \cdots \\
\vdots & \cdots & \cdots & \ddots & \ddots & \ddots & \vdots \\
\cdots & \cdots & \cdots & \cdots & 0 & A_{n-1,n-2} & A_{n-1,n-1}
\end{bmatrix}
\vec{u} =
\begin{bmatrix}
b_0 \\ \vdots \\ \vdots \\ \frac{x_{i+1} - x_{i-1}}{2} \\ \vdots \\ \vdots \\ b_{n-1}
\end{bmatrix}
\tag{9}
$$

Notice that the entries of A and b are the result of integrating the shape functions and therefore depend only on the discretization of the domain. Once the entries for matrix A and vector b are filled out, the hard work is done. There are a few ways to solve a linear system such as $Au = b$. For instance, the direct solution is defined as:

$$
u = A^{-1} b
$$

0.2.3 Boundary Conditions

Here we consider the most commonly used boundary conditions for PDEs: the **Dirichlet** and **Neumann** boundary conditions.

Dirichlet Boundary Conditions

The first-type boundary condition specifies the exact value of u on the boundary:

$$u(x) = g, \quad x \in \partial\Omega$$

For the 1D case, we have $u(x_0) = u_0$ and $u(x_{n-1}) = u_{n-1}$. Therefore, the first and the last equations in (7) (or (8)) are redundant, which eliminates the corresponding unknowns.

Particularly, the equation of the **second** row is modified as:

$$A_{1,0}u_0 + A_{1,1}u_1 + A_{1,2}u_2 = b_1 \Longrightarrow A_{1,1}u_1 + A_{1,2}u_2 = b_1 - A_{1,0}u_0$$

Similarly, we modify the **second-to-last** row as:

$$A_{n-2,n-3}u_{n-3}+A_{n-2,n-2}u_{n-2}+A_{n-2,n-1}u_{n-1} = b_{n-2} \Longrightarrow A_{n-2,n-2}u_{n-2}+A_{n-2,n-1}u_{n-1} = b_{n-2}-A_{n-2,n-3}u_{n-3}$$

Putting everything together in the matrix equation system leads to the following matrix solution:

$$\begin{bmatrix} A_{1,1} & A_{1,2} & 0 & \cdots & \cdots & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \cdots & \cdots & \vdots \\ \cdots & \cdots & -\frac{k}{x_i - x_{i-1}} & \left(\frac{k}{x_i - x_{i-1}} + \frac{k}{x_{i+1} - x_i}\right) & -\frac{k}{x_{i+1} - x_i} & \cdots & \cdots \\ \vdots & \cdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ \cdots & \cdots & \cdots & \cdots & 0 & A_{n-2,n-3} & A_{n-2,n-2} \end{bmatrix} \vec{u} = \begin{bmatrix} b_1 - A_{1,0}u_0 \\ b_2 \\ \vdots \\ \frac{x_{i+1} - x_{i-1}}{2} \\ \vdots \\ b_{n-3} \\ b_{n-2} - A_{n-2,n-1}u_{n-1} \end{bmatrix} \tag{10}$$

where $\vec{u} = (u_1, \ldots, u_{n-2})$.

Neumann Boundary Conditions

Neumann boundary condition specifies the derivative value on the boundary:

$$\frac{\partial u}{\partial \vec{n}} = \nabla u \cdot \vec{n} = g, \quad x \in \partial\Omega$$

where $\vec{n}$ represents the normal vector on the boundary. For the 1D case, $\vec{n}$ degenerates into a constant, which is -1 on the left boundary x_0 and 1 on the right boundary x_{n-1}. Therefore, we have the Neumann boundary conditions $-u'(x_0) = g(x_0)$ and $u'(x_{n-1}) = g(x_{n-1})$.

The Neumann boundary condition modifies the right-hand side vector b by incorporating the boundary term from the weak formulation. It appears in the **squared brackets in the weak formulation** (3). The integration by parts creates an extra term for the boundary.

The boundary term contains the first-order derivative of u. When opening the brackets, the boundary term is written as follows:

$$\left[w\frac{\partial u}{\partial x}\right]_{x_0}^{x_{n-1}} = w(x_{n-1})u'(x_{n-1}) - w(x_0)u'(x_0) = w(x_{n-1})g(x_{n-1}) + w(x_0)g(x_0)$$

Recall from the definition of the test function (which is equal to the shape function) that the value of w at the nodes x_0 and x_{n-1} is one. The matrix equation system can be determined as follows:

$$\begin{bmatrix} A_{0,0} & A_{0,1} & 0 & \cdots & \cdots & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \cdots & \cdots & \vdots \\ \cdots & \cdots & -\frac{k}{x_i - x_{i-1}} & \left(\frac{k}{x_i - x_{i-1}} + \frac{k}{x_{i+1} - x_i}\right) & -\frac{k}{x_{i+1} - x_i} & \cdots & \cdots \\ \vdots & \cdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ \cdots & \cdots & \cdots & \cdots & 0 & A_{n-1,n-2} & A_{n-1,n-1} \end{bmatrix} \vec{u} = \begin{bmatrix} b_0 + g(x_0) \\ b_1 \\ \vdots \\ \frac{x_{i+1} - x_{i-1}}{2} \\ \vdots \\ b_{n-2} \\ b_{n-1} + g(x_{n-1}) \end{bmatrix} \tag{1}$$

The system contains n unknowns as the number of nodes and therefore n equations.