Math 6008 Numerical PDEs–Lecture 14

Finite element method and Fourier spectral method

Instructor: Lei Li, INS, Shanghai Jiao Tong University;
Email: leili2010@sjtu.edu.cn

1 Second order elliptic equations in 2D

For related material, you may read section 1.4 in the book of Johnson.

We consider the Poisson equation

$$
\begin{aligned}
&-\Delta u = f, \ \ \Omega, \\
&\frac{\partial u}{\partial n} + \alpha u = g, \ \ \partial\Omega.
\end{aligned}
$$

We may derive the weak formulation as before. In fact, we multiply a test function v and integrate by parts and then figure out the the correct spaces etc.

The suitable space is

$$
H^1 = \{v : \int_\Omega v^2 + |\nabla v|^2 dx < \infty, \int_{\partial\Omega} v^2 ds < \infty\}.
$$

[In fact, the last condition $\int_{\partial\Omega} v^2\, ds < \infty$ can be derived by $\int_\Omega |\nabla v|^2 dx < \infty$.] The bilinear form for this problem is

$$
D(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx + \int_{\partial\Omega} \alpha u v \, ds,
$$

and the linear form is

$$
F(v) = \int_\Omega f v \, dx + \int_{\partial\Omega} g v \, ds.
$$

The Galerkin's weak formulation is

$$
\text{Find } u \in H^1 \text{ such that}
$$

$$
D(u, v) = F(v), \quad \forall v \in H^1.
$$

1.1 Setup of the finite element problem

Similarly as 1D problems, we need a triangulation of the domain to construct the finite dimensional space V_h.

Triangulation

There are several choices of the shapes of the elements. The most frequently used shape is the triangle. For the triangles, one needs to determine the locations of the vertices. The triangles cannot be too slim. There are some techniques to adjust the vertices.

After the triangulation is obtained, we may use some data structures to store them.

Data structure for triangulation.

N is the number of elements; M is the number of nodes.

- T which is of size $3 \times N$.
- Z which is of size $2 \times M$

The n-th column of T is the vector of positions of the three vertices in Z. The i-th column of Z is the coordinates of the i-th node. For example, if the first column of T is $[1, 4, 5]^T$, then the vertices of the first triangle are the first, the fourth and the fifth nodes in Z.

Note that the ordering is important as it affects the bandwidth of the stiff matrix.

The subspace and basis functions

For convenience, we assume that Ω has piecewise straight boundary curves so that the union of the triangles in the triangulation will be Ω. (Otherwise, the constructed V_h below is not strictly a subspace of $H^1(\Omega)$ but careful treatment can still prove the convergence.)

Again, we consider piecewise polynomials. As before, for the derivatives to be integrable, we need the functions to be continuous, or $C(\bar{\Omega})$. We again consider the simplest case: the functions are linear on each element.

$$U_h = \{v_h \in C(\bar{\Omega}) : v_h \text{ is linear on each element }\}.$$

Clearly, $U_h \subset H^1$ and one can verify that this is a finite dimensional space. Each function in U_h is clearly determined by its values on the vertices.

Consider an elment e with vertices $P_i(x_i, y_i), P_j(x_j, y_j), P_k(x_k, y_k)$. The function is linear on e so

$$u(x, y) = ax + by + c.$$

The values at the three vertices are u_i, u_j, u_k, given

$$\begin{cases} ax_i + by_i + c = u_i, \\ ax_j + by_j + c = u_j, \\ ax_k + by_k + c = u_k. \end{cases}$$

Then,

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{pmatrix}^{-1} \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix}.$$

By the formula of inverse,

$$\begin{pmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{pmatrix}^{-1} = \frac{1}{\Delta} \begin{pmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \\ c_i & c_j & c_k \end{pmatrix}$$

where

$$\Delta = \det \begin{pmatrix} x_i & y_i & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{pmatrix} = 2\Delta_e,$$

where Δ_e is the area of the triangle determined by the three vertices. To see this, it is clear that Δ is the volume of the parallelepiped determined by the three vectors $(x_i, y_i, 1)$,

$$\Delta = \vec{r_i} \cdot (\vec{r_j} \times \vec{r_k}) = \vec{r_i} \cdot [(\vec{r_j} - \vec{r_i}) \times (\vec{r_k} - \vec{r_i})].$$

The volume of the parallelepipied of the latter is the base area times the height. The height is 1 and the base area is the area of the parallelogram, or twice of the area of the triangle. The a_i, a_j etc are the algebraic cofactors (代数余子式) of the element in the matrix. For example, b_i corresponds to the cofactor A_{12}.

Hence,

$$a = \frac{1}{\Delta}(a_i u_i + a_j u_j + a_k u_k), b = \frac{1}{\Delta}(b_i u_i + b_j u_j + b_k u_k), c = \frac{1}{\Delta}(c_i u_i + c_j u_j + c_k u_k).$$

Consequently,

$$u(x, y) = u_i \frac{1}{\Delta}(a_i x + b_i y + c_i) + u_j \frac{1}{\Delta}(a_j x + b_j y + c_j) + u_k \frac{1}{\Delta}(a_k x + b_k y + c_k).$$

Note that a_i, b_i, c_i corresponds to Laplace expansion along the first row of the coefficient of the matrix. Hence,

$$a = \frac{1}{\Delta}(a_i u_i + a_j u_j + a_k u_k) = \frac{1}{\Delta} \det \begin{pmatrix} u_i & y_i & 1 \\ u_j & y_j & 1 \\ u_k & y_k & 1 \end{pmatrix}.$$

This is the **Cramer's rule**. The formulas for b and c can be derived similarly.

The expression multiplied with u_i is

$$N_i(x, y) = \frac{1}{\Delta}(a_i x + b_i y + c_i) = \frac{1}{\Delta} \det \begin{pmatrix} x & y & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{pmatrix}.$$

Clearlly, we are just replacing the corresponding vertices with the general point (x, y) to get the formula.

Then,

$$u(x,y) = u_i N_i(x,y) + u_j N_j(x,y) + u_k N_k(x,y),$$

where, for example,

$$N_i(x,y) = \frac{1}{\Delta} \det \begin{pmatrix} x & y & 1 \\ x_j & y_j & 1 \\ x_k & y_k & 1 \end{pmatrix}.$$

These functions are called the shape functions on e. It can be verified that the linear functions can be recovered by the exactly:

$$\begin{aligned} &N_i + N_j + N_k = 0, \\ &x_i N_i + x_j N_j + x_k N_k = x, \\ &y_i N_i + y_j N_j + y_k N_k = y. \end{aligned}$$

In fact, since linear functions can be determined by the values at the vertices, these formulas are straightforward to see.

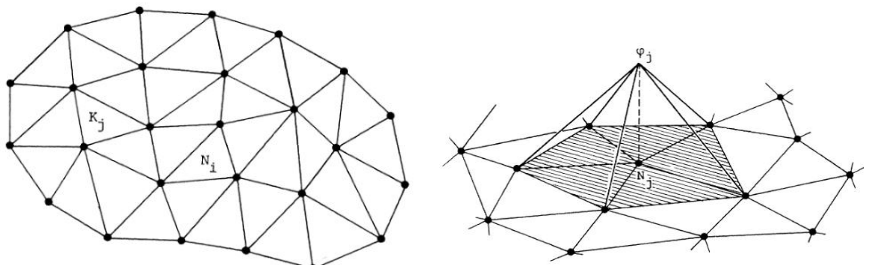

图 1: Illustration of the triangulation and the basis function

Now, we consider the basis functions of the space U_h. As before, $\phi_i(x,y)$ should be continuous and $\phi_i(P_j) = \delta_{ij}$. Hence, ϕ_i is a hat function and on each element near the vertex P_i, its formula can be given by suitable N listed as above.

The general formula for $u_h \in U_h$ is

$$u_h(x,y) = \sum_{i=1}^{N_p} u_i \phi_i(x,y).$$

The Galerkin FEM

Note that there is no boundary condition involved. Hence,

$$U_h \subset H^1.$$

U_h above can be used as the subspace of H^1 directly. The space U_h is a Hilbert space.

The Galerkin's FEM is

$$\text{Find } u_h \in U_h, \text{ such that } D(u_h, v) = F(v), \text{ for all } v \in U_h.$$

1.2 The procedure for formulating the stiff matrix and load vector

Again, to construct the stiff matrix and the load vector, one is going to loop over the elements. The procedure conceptually can be divided into the following step:

- For each e_i, compute the element stiffness matrix (单元刚度矩阵) and the element load vector (单元荷载向量)
- Assembling procedure. The element stiffness matices and load vectors will assembled into the total stiffness matrix and the total load vector.
- Treating the constraints and the boundary conditions.

Read the book for more details, and we skip the details here.

A comment for programming

You may read section 1.8 in Johnson's book for further details.

Even though the stiff matrix and load vector can be constructed using the above procedure, we do not actually compute the element stiff matrices one by one and then compute the enlarged matrices and add to the total stiff matrix for assembling. In practical coding, we actually add the corresponding entries in the element stiff matrices directly into the total stiff matrix.

We may do the following.

Generate the stiffness matrix and load vector

```
- For $n = 1 : N$
    For $\alpha = 1 : 3$
        $p = T(\alpha, n)$
        For $\beta = 1 : 3$
            $q = T(\beta, n)$.
            $a_{pq}^n = \int_{e_n} \nabla\varphi_p \cdot \nabla\varphi_q dx + \int_{\partial e_n \cap \partial\Omega} \alpha(s)\varphi_p\varphi_q ds$.
            $A(p, q) \leftarrow A(p, q) + a_{pq}^n$
        End $\beta$
        $b^p \leftarrow b^p + \int_{e_n} f\varphi_p dx + \int_{\partial e_n \cap \partial\Omega} g\varphi_p ds$
    End $\alpha$
  End $n$
```

2 Other topics for elements

Here, we perform some brief discussion for more advanced topics in FEM.

2.1 Rectangular and isoparametric elements

For 2D problems, the rectuangular elements are frequently used in applications. By changing of variables, we consider the special case $e = [0, 1] \times [0, 1]$ here.

Let us consider the basic case, where we need the functions to satisfy $\int (u^2 + |\nabla u|^2) \, dx < \infty$. As before, if the function is a continuous piecewise polynomial, then it is in this class.

There are four vertices in the element. If we consider fitting the values, there should be then four degrees of freedom for each element. A linear function is not enough as it only has three parameters. Here, one may use the bilinear functions

$$u_h(x, y) = a + bx + cy + dxy.$$

Then, there are four parameters. Hence, using the values at the vertices can determine the parameters uniquely. Moreover, for fixed x, it is linear in y; for fixed y, it is linear in x. Hence, for two adjacent rectangles, they will share a common edge. On this edge, the functions from both elements are linear. They share the same values on the two endpoints. Since the function is linear, the two functions are the same on this edge. Hence, the patched solution would be continuous.

The shape function can be

$$N(\xi, \eta) = \frac{1}{4}(1 \pm \xi)(1 \pm \eta).$$

The isoparametric elements (等参数单元) can be viewed as generalization of the rectangular elements for general domains. You may read the book for more information.

2.2 Higher order polynomials

If the differential equations have higher order derivatives or you want higher accuracy, then you need to use higher order polynomials on each element.

In the triangulation, one has a set of simplexes K_i. These K_i's do not overlap. Consider constructing the basis functions for V_h. There are some common ways people do:

1. Let the basis functions be polynomials on each K_j.

2. For a fixed basis function, the values and derivatives of the polynomials on the traingles should match somehow on the vertices and edges so that the basis function φ_i is in the space H we consider.

3. Each basis function should be nonzero in a small number of K_i's.

4. The polynomials on the whole domain for all degree $d \le m$ (for some m) can be represented accurately using these basis so that the method has certain accuracy.

The detailed explanation deserves a separate course and we skip them. You may read section 3 of Chapter 4 in the book. Moreover, if you want more theoretical analysis and the general rules for constructing the finite element spaces, you can read the book by Johnson or other material.

3 Analysis (skipped)

For the analysis of the errors (including a prior and posterior estimates), one needs some tools from functional analysis (like Lax-Milgram theorem). We refer you to the book by Johnson.

Fourier spectral methods

Recall the DFT (离散傅里叶变换) for a sequence $v = (v_0, \cdots, v_{N-1})$ is given by

$$\hat{v}_k = \sum_{n=0}^{N-1} e^{-ikx_n} v_n, \quad k \in \mathbb{Z},$$

where $x_n = \frac{2\pi}{N} n$. (You can regard v as a function on $[0, 2\pi)$ and x_n is a sample point on the interval.) The inverse DFT is given by

$$v_n = \frac{1}{N} \sum_{k=0}^{N-1} e^{ikx_n} \hat{v}_k = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} e^{ikx_n} \hat{v}_k.$$

We know that they can be computed using FFT in a complexity $O(N \log N)$.

4 Fourier spectral differentiation

For general x, we can define a function

$$f(x) = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} e^{ikx} \hat{v}_k,$$

and then we use $f^{(m)}(x_j)$ to approximate the derivatives $v^{(m)}(x_j)$. The above idea generates the following **Fourier differentiation**:

- Given $v = (v_1, \dots, v_N)$, compute $\hat{v}$.
- Define $\hat{w}_k = (ik)^m \hat{v}_k$, for $k = -N/2+1, \dots, N/2$ (or $k = 0, \dots, N/2, -N/2+1, \dots, -1$ in Matlab.)
- Compute the inverse DFT (inverse FFT) and get w_k, the real part of which is the approximation of the derivative.

Remark 1. *However, the $k = N/2$ mode is a little bit strange. Assume v is a real array. Then,*

$$\hat{v}_{N/2} = \overline{\hat{v}}_{-N/2} = \overline{\hat{v}}_{N/2}.$$

This means $\hat{v}_{N/2}$ is real. This mode will contribute derivative

$$\hat{v}_{N/2} \frac{iN}{2} e^{i\frac{N}{2}x_j}$$

Clearly, this part is imaginary at the given grid points. Taking real part will get rid of this.

Exercise: *Will it matter if we use $f(x) = \frac{1}{N}\sum_{k=0}^{N-1} e^{ikx}\hat{v}_k$ to compute the derivatives? If it matters, which one is better? Choose a periodic smooth function and code up to check out. Can you explain this? (Hint: Use the Aliasing formula in the next lecture.)*

Remark 2. *As a corollary of the Paserval equality, without taking the real part, the discrete integration by parts by Fourier differentiation holds.*