Метод конечных элементов: теоретические сведения и программная реализация

Москва
2024

1 Метод Ритца поиска приближенных обобщенных решений операторных уравнений

Метод Ритца – вариационный и проекционный метод, с помощью которого можно приближенно находить обобщенные решения операторных уравнений в гильбертовых пространствах. Далее мы еще поясним, что это такое. Этот метод появился на рубеже XIX–XX веков и первоначально применялся инженерами для решения задач механики твердого тела. Похожими методами являются метод Бубнова–Галёркина, Галёркина–Петрова. Далее приводится проекционная форма метода Ритца.

Рассмотрим следующее операторное уравнение:

$$\mathcal{A}u = f, \quad \mathcal{A} : H \to H, \tag{1}$$

где $f \in H$, а $\mathcal{A}$ – некоторый оператор, действующий в гильбертом пространстве H. Он имеет следующие свойства:

- Область его определения $D(\mathcal{A})$ плотна в H, т.е. $\overline{D(\mathcal{A})} = H$;
- $\mathcal{A}$ – симметричный, т.е. $(\mathcal{A}u, v) = (u, \mathcal{A}v) \;\; \forall u, v \in D(\mathcal{A})$;
- $\mathcal{A}$ – положительно определённый, т.е. $\exists \gamma : (\mathcal{A}u, u) \geq \gamma^2 ||u||^2 \;\; \forall u \in D(\mathcal{A})$.

Пользуясь этими свойствами, на $D(\mathcal{A})$ можно ввести новое скалярное произведение $[u, v] = (\mathcal{A}u, v)$ и порождаемую им *энергетическую норму* $[u] = \sqrt{(Au, u)}$, а с её помощью ввести специальное *энергетическое пространство* $H_{\mathcal{A}}$ – пополнение $D(\mathcal{A})$ по энергетической норме.

Введя энергетическое пространство, можно рассматривать так называемую *слабую постановку задачи.* Это понятие очень важно. Дело в том, что введение энергетического пространства, расширяющего область определения оператора $\mathcal{A}$, позволяет нам рассматривать более широкий круг задач, ослабив требования на гладкость в некоторых местах. Вспомните: при построении конечноразностных схем предполагается достаточная гладкость коэффициентов и решения. Теперь же эти требования несколько ослабляются.

Определение 1. $u \in H_{\mathcal{A}}$ *называется обобщённым решением задачи* (1)*, если*

$$[u, v] = (f, v) \;\; \forall v \in H_{\mathcal{A}}.$$

Теорема 1. *Обобщенное решение существует, единственно и ограничено по норме, а именно:* $\forall f \in H \;\; \exists! u \in H_{\mathcal{A}} : \;\; [u] \leq C||f||$.

Не будем останавливаться на ее доказательстве, которое выполняется с помощью теоремы Рисса и свойств энергетической нормы.

Метод Ритца предлагает достаточно ясный способ поиска *приближенного* обобщенного решения. В энергетическом пространстве $H_{\mathcal{A}}$ выбирается система линейно независимых функций $\{\varphi_i\}_{i=1}^N$. Их линейная оболочка обозначается $H_{\mathcal{A}}^N$. Функции $\{\varphi_i\}_{i=1}^N$ должны обладать свойством *предельной плотности* в $H_{\mathcal{A}}$:

$$\forall v \in H_{\mathcal{A}} \inf_{v_N \in H_{\mathcal{A}}^N} [v - v_N] \leq \varepsilon_N(v) \to 0 \quad \text{при} \quad N \to \infty \tag{2}$$

Иными словами, любой элемент из энергетического пространства $H_{\mathcal{A}}$ можно с любой заданной точностью приблизить линейной комбинацией базисных функций из $H_{\mathcal{A}}^N$, если выбрать достаточно большое N.

Тогда и приближенное обобщенное решение u будем искать в $H^N_{\mathcal{A}}$. Вопрос заключается в поиске коэффициентов $\{a_i\}_{i=1}^N$ разложения по базисным функциям. Для этого вычислим невязку, которую дает приближенное решение u_N, и потребуем ее ортогональности по отношению к $H^N_{\mathcal{A}}$:

$$\mathcal{A}u_N - f \perp H^N_{\mathcal{A}}, \tag{3}$$

что равнозначно ортогональности по отношению к каждой из базисных функций

$$\mathcal{A}u_N - f \perp \varphi_1, \dots, \varphi_N. \tag{4}$$

Запишем же эти условия в виде системы уравнений

$$\begin{cases} (\mathcal{A}u_N - f, \varphi_1) = 0, \\ \dots \\ (\mathcal{A}u_N - f, \varphi_N) = 0. \end{cases} \tag{5}$$

Эту систему можно представить в виде

$$\begin{cases} (\mathcal{A}u_N, \varphi_1) = (f, \varphi_1), \\ \dots \\ (\mathcal{A}u_N, \varphi_N) = (f, \varphi_N). \end{cases} \tag{6}$$

или

$$\begin{cases} [u_N, \varphi_1] = (f, \varphi_1), \\ \dots \\ [u_N, \varphi_N] = (f, \varphi_N). \end{cases} \tag{7}$$

Вспомним теперь, что $u_N = \sum_{i=1}^N a_i \varphi_i$, отсюда

$$\begin{cases} a_1[\varphi_1, \varphi_1] + \dots + a_N[\varphi_N, \varphi_1] = (f, \varphi_1), \\ \dots \\ a_1[\varphi_1, \varphi_N] + \dots + a_N[\varphi_N, \varphi_N] = (f, \varphi_N). \end{cases} \tag{8}$$

Видно, что эта система является линейной системой относительно коэффициентов $\{a_i\}$:

$$Aa = b, \tag{9}$$

где $A_{ij} = [\varphi_i, \varphi_j]$, а $b_i = (f, \varphi_i)$. Матрица A называется *матрицей жесткости.*

Лемма 1. $A = A^T > 0$.

Доказательство мы также опустим, он техническое и достаточно простое.

Результатом всех проделанных действий является

Теорема 2. *Пусть u – обобщенное решение (см. опр. 1) задачи* (1)*, а u_N – приближенное обобщенное решение, являющееся линейной комбинацией предельно плотных базисных функций $\{\varphi_i\}$, а коэффициенты разложения определяются из линейной системы* (9)*. Тогда*

1. *Приближенное обобщенное решение u_N существует и единственно, при этом $[u_N] \le C_1||f||$;*
2. $||u - u_N|| \le C_2 \varepsilon_N(u) \to 0$ *при* $N \to \infty$.

Доказательство. По лемме 1 решение системы (9) существует и единственно, значит, таково и приближенное обобщенное решение, определяемое полученными коэффициентами разложения.
$[u_N]^2 = [u_N, u_N] = (f, u_N) \leq ||f|| \cdot ||u_N|| \leq ||f|| \cdot [u_N]/\gamma$, отсюда $C_1 = 1/\gamma$.
Из способа поиска коэффициентов $\{a_i\}$ имеем $[u_N, \varphi_i] = 0 \quad \forall i$. В то же время, поскольку u – обобщенное решение, то согласно его определению 1, $[u, \varphi_i] = 0 \quad \forall i$ тоже. Отсюда $[u - u_N, \varphi_i] = 0 \quad \forall i$, то есть ошибка приближенного решения ортогональна всем базисным функциям, следовательно, и любой их линейной комбинации. Выберем такую комбинацию $v_N = \sum_{i=1}^{N} b_i \varphi_i$ и имеем $[u - u_N, u_N - v_N] = 0$.
Далее запишем $[u - u_N]^2 = [u - u_N, u - u_N] = [u - u_N, u - u_N + v_N - v_N] = [u - u_N, u - v_N] - [u - u_N, u_N - v_N] = [u - u_N, u - v_N] \leq [u - u_N][u - v_N]$.
Отсюда $[u - u_N] \leq [u - v_N] \quad \forall v_N \in H_{\mathcal{A}}^N$.
Возьмем теперь inf по всем $v_N \in H_{\mathcal{A}}^N$ и учтем, что $[u - u_N]$ не изменится, а правую часть можно преобразовать по свойству предельной плотности (2), получим
$[u - u_N] \leq \inf_{v_N \in H_{\mathcal{A}}^N} [u - v_N] \leq \varepsilon_N(u) \to 0 \quad$ при $\quad N \to \infty$; далее вспомним, что $||u - u_N|| \leq [u - u_N]/\gamma$, отсюда $C_2 = 1/\gamma$. □

ИТОГ

Для задачи (1) введено понятие обобщённого решения (опр. 1), предложен способ его поиска в виде линейной комбинации некоторых базисных функций $\{\phi_i\}$. Чтобы получить это приближенное решение, надо найти коэффициенты разложения по базисным функциям, решив линейную систему (9). Такое приближенное решение существует, единственно и сходится к обобщенному при увеличении числа базисных функций.

2 Метод Ритца в применении к краевой задаче для стационарного уравнения диффузии

В качестве примера рассмотрим краевую задачу для стационарного уравнения диффузии с однородными (нулевыми) граничными условиями Дирихле:

$$\begin{cases} -\nabla \cdot (\mathbb{D} \nabla C) = f \quad \text{в} \quad \Omega, \\ C|_{\partial\Omega} = 0. \end{cases} \tag{10}$$

В этой задаче оператор $\mathcal{A}$ имеет вид

$$\mathcal{A} = -\nabla \cdot (\mathbb{D} \nabla \cdot) : L_2(\Omega) \to L_2(\Omega), \tag{11}$$

т.е. $H \equiv L_2(\Omega)$и область определения

$$D(\mathcal{A}) = \{u \in C^2(\overline{\Omega}) : u|_{\partial\Omega} = 0\} \subset H. \tag{12}$$

Можно показать, что $\mathcal{A}$ – симметричный и положительно определенный (используя неравенство Пуанкаре–Стеклова).

Энергетическим пространством $H_{\mathcal{A}}$ является соболевское пространство $W_2^1(\Omega)$, и обобщенное решение ищется именно в нем. Это пространство значительно шире пространства $C^2(\overline{\Omega})$.

3 Кусочные функции и метод конечных элементов

Вопросом, определяющим практическую применимость метода Ритца, является выбор базисных функций $\{\varphi_i\}$. Первичная практика применения метода Ритца распространялась на области простой формы и такие базисные функции, как тригонометрические. В таком случае матрица A в линейной системе (9) является плотной.

В какой-то момент (см., например, работу Р. Куранта «Variational methods for the solution of problems of equilibrium and vibrations» 1943 года) стало понятно, что если в качестве $\{\varphi_i\}$ использовать функции, не равные нулю лишь в некоторой малой области, то в матрице A большая часть элементов будет нулевыми. В качестве таких функций можно использовать кусочно-полиномиальные функции.

В большинстве случаев кусочные функции удовлетворяют следующему соотношению:

$$\varphi_i(x_j) = \delta_{ij}, \tag{13}$$

где x_j – координаты j-го узла, а φ_i – базисная функция, соответствующая j-му узлу.

3.1 Кусочно-линейные функции в одномерном случае

В одномерном случае базисные кусочно-линейные функции имеют вид, приведенный на рисунке 1. В этом случае отрезок разбит узлами, которые на оси Ox обозначены как 0, 1, 2, 3. Приведены базисные функции $\varphi_0, \ldots, \varphi_3$.

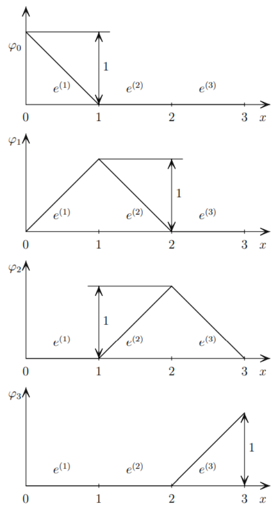

Рис. 1: Кусочно-линейные базисные функции на одномерной сетке

Для таких функций матрица A в системе (9) является трехдиагональной. Трехдиагональная она потому, что для некоторого i число $[\varphi_i, \varphi_j]$ не равно нулю только для $j = i-1,\ i,\ i+1$. Можно проверить, что итоговая трехдиагональная матрица практически совпадает с матрицей, которую дает метод конечных разностей.

3.2 Кусочно-линейные функции на треугольниках

Такой подход можно расширить и на треугольники. Пример базисной функции для одного узла приведен на рисунке 2. Такая функция не равна 0 только в ячейках, которые окружают

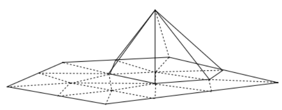

Рис. 2: Кусочно-линейная базисная функция на треугольной сетке

узел, которому она соответствует. В этих треугольниках она ведет себя линейно. Во всех остальных ячейках она нулевая.

Если рассмотреть некоторый треугольник с вершинами (x_1, y_1), (x_2, y_2), (x_3, y_3) и ввести на нем локальную нумерацию узлов, то базисные функции φ_1, φ_2, φ_3 этих узлов на данном треугольнике имеют вид

$$\varphi_1(x, y) = \frac{(x - x_3)(y_2 - y_3) - (x_2 - x_3)(y - y_3)}{(x_1 - x_3)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_3)} \tag{14}$$

$$\varphi_2(x, y) = \frac{(x - x_3)(y_1 - y_3) - (x_1 - x_3)(y - y_3)}{(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3)} \tag{15}$$

$$\varphi_3(x, y) = \frac{(x - x_1)(y_2 - y_1) - (x_2 - x_1)(y - y_1)}{(x_3 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_1)} \tag{16}$$

Выпишем выражения для градиентов этих функций:

$$\nabla\varphi_1(x, y) = \frac{1}{(x_1 - x_3)(y_2 - y_3) - (x_2 - x_3)(y_1 - y_3)} \cdot \begin{bmatrix} (y_2 - y_3) \\ -(x_2 - x_3) \end{bmatrix} \tag{17}$$

$$\nabla\varphi_2(x, y) = \frac{1}{(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3)} \cdot \begin{bmatrix} (y_1 - y_3) \\ -(x_1 - x_3) \end{bmatrix} \tag{18}$$

$$\nabla\varphi_3(x, y) = \frac{1}{(x_3 - x_1)(y_2 - y_1) - (x_2 - x_1)(y_3 - y_1)} \cdot \begin{bmatrix} (y_2 - y_1) \\ -(x_2 - x_1) \end{bmatrix} \tag{19}$$

3.2.1 Особенности сборки матрицы жесткости

Можно догадаться, что значение $[\varphi_i, \varphi_j]$ не равно 0 только в том случае, если узлы i и j являются вершинами одного треугольника.

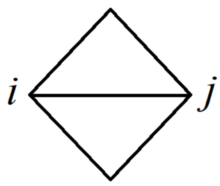

Рис. 3: Два узла i и j, чьи базисные функции одновременно не равны нулю на двух треугольниках

Заполнение матрицы жесткости A (9) можно проводить по-разному. Например, можно заполнять матрицу по строкам. Каждая строка соответствует одному узлу. В ходе заполнения строки для i-го узла нужно вычислять все ненулевые произведения $[\varphi_i, \varphi_j]$. Для соседних узлов i и j их базисные функции не равны нулю одновременно на двух треугольниках. Таким образом, для вычисления $[\varphi_i, \varphi_j]$ нужно будет обратиться к двум разным треугольникам, чтобы произвести интегрирование. Более того, в ходе заполнения строки для i-го узла к каждому из соседних треугольников придется обратиться 2 раза при вычислении функций для разных соседних узлов. Это все приводит к лишним вычислениям.

Поэтому сборку глобальной матрицы жесткости A выполняют по-другому. Запускается цикл по по ячейкам (треугольникам). На каждом треугольнике рассматриваются φ_1, φ_2, φ_3 (используется локальная для треугольника нумерация узлов). Далее для этих трех функций вычисляются все попарные скалярные произведения, записывающиеся в *локальную* матрицу жесткости. После чего локальная матрица встраивается в глобальную.

4 Выполнение задания

По адресу `https://github.com/denis-anuprienko/INMOST_Practicum_2024/src/` расположены заготовки кода по МКЭ. Кратко пройдемся по его устройству:

- Класс `Problem` содержит в себе все необходимое для решения задачи: сетку, тег решения (искомой концентрации), тег тензора диффузии, тег граничных условий, тег источника, тег аналитического решения и тег индекса узла в глобальной системе уравнений, маркер узлов Дирихле (т.е. всех граничных узлов) и число этих узлов

- В функции `Problem::initProblem()` происходит создание тегов; перебор ячеек с проверкой, что это треугольники, и с записью значений тензора диффузии (для всех ячеек они одинаковы, но при рассмотрении неоднородных областей могут отличаться); перебор узлов с записью для них значений аналитического решения и источника, разметка и подсчет узлов Дирихле

- Функция `basis_func()` выдает значение базисной функции φ, соответствующей конкретному узлу конкретной ячейки в заданной точке

- Функция `basis_func_grad()` выдает значение градиента базисной функции φ, соответствующей конкретному узлу конкретной ячейки. Зависимость от x и y отсутствует, так как градиент является константной. Обратите внимание, что значение возвращается в виде `INMOST::rMatrix`, формата данных для хранения плотных матриц из действительных чисел

- Функция `coords_from_barycentric()` вычисляет глобальные координаты по барицентрическим

- Функция `Problem::linear_approx_tri()` выдает квадрат ошибки численного решения, т.е. значение $|u - u_N|^2 \equiv |u - \sum_{i=1}^{N} a_i \varphi_i|^2$

- Функция `Problem::integrate_over_triangle()` возвращает результат интегрирования функции $|u - u_N|^2$ по треугольнику с помощью кубатурной формулы из методички

- Функция `Problem::assembleLocalSystem()` собирает локальные на ячейке матрицу жесткости (плотную, размера 3×3) и вектор правой части

- Функция `Problem::assembleGlobalSystem()` собирает глобальные матрицу жесткости (разреженную, размера $N \times N$, где N – число внутренних узлов сетки) и вектор правой части. Происходит это путем перебора ячеек, сборки для каждой из них локальной системы и затем встраивания в глобальную систему

- Функция `Problem::run()` осуществляет процесс решения: собирает глобальную систему, решает ее и сохраняет решение на сетке