Московский государственный университет имени М. В. Ломоносова
Механико-математический факультет
Кафедра Теории Вероятностей

Отчёт

студента 409 группы
Сидоренко Артура Павловича

Численное решение одного уравнения второго порядка

Москва, 2020

1 Постановка задачи

Дана краевая задача

$$-u''(x) + b(x)u(x) = f(x), u(0) = 0, u'(1) = 0. \tag{1}$$

Требуется составить разностную схему для поиска решения задачи на отрезке $[0, 1]$ и численно её апробировать. Предлагается использовать смещённую вправо сетку на $[0, 1]$ из N узлов с шагом $h = 2/(2N - 1)$, $x_k = kh$.

2 Разностная схема

Я буду руководствоваться определениями в [1], стр. 254-259 и [2], стр.27-32.

Предложим следующую разностную схему:

$$-\frac{u_{k+1} - 2u_k + u_{k-1}}{h^2} + b_k u_k = f_k, k = 1, \dots, N-1 \tag{2}$$

$$u_0 = 0, \tag{3}$$

$$u_{N-1} = u_N, \tag{4}$$

где

$$x_k = kh, \tag{5}$$

$$h = \frac{2}{2N-1}, \tag{6}$$

$$f_k = f(x_k), \tag{7}$$

$$b_k = b(x_k). \tag{8}$$

Проверим свойство аппроксимации на решении. Подставим в (2) $u_k = u(x_k)$, где u - решение (1). По формуле Тейлора

$$u(x_{k+1}) = u(x_k) + u'(x_k)h + \frac{h^2}{2}u''(x_k) + \frac{h^3}{3}u'''(x_k) + \frac{h^4}{24}u^{IV}(x_k + \xi), \tag{9}$$

$$u(x_{k+1}) = u(x_k) - u'(x_k)h + \frac{h^2}{2}u''(x_k) - \frac{h^3}{3}u'''(x_k) + \frac{h^4}{24}u^{IV}(x_k - \eta). \tag{10}$$

После вычичлений получим выражение

$$-u''(x_k) - \frac{h^2}{24}(u^{IV}(x_k + \xi) + u^{IV}(x_k + \eta)) + b(x_k)u(x_k) - f(x_k), k = 1, \dots N-1. \tag{11}$$

Свойство, что u - решение, позволяет сократить часть слагаемых, и останется лишь $\frac{h^2}{24}(u^{IV}(x_k+\xi)+u^{IV}(x_k-\eta))$. Эта выкладка даёт аппроксимацию $||L_h(u)_h - f_h||_{C_h} = O(h^2)$. Из этого сразу следует $||L_h(u)_h - f_h||_{L_h^2} = O(h^2)$, где норма L_h^2 порождена скалярным произведением $(u, v) = \sum_1^{N-1} u_k v_k$. Эта выкладка доказывает следующую теорему.

Theorem 2.1. *Схема* (2) – (4) *обеспечивает второй порядок аппроксимации задачи* (1).

Следующий шаг - это проверка устойчивости схемы. Выкладка приведена в разделе 5.

Теорема Филиппова позволяет объединить аппроксимацию и устойчивость для случая линейных дифференциальных операторов и получить вывод о сходимости приближённых решений к настоящему.

Theorem 2.2. *Схема* (2)-(2) *имеет второй порядок сходимости.*

Доказательство. Немедленно следует из теоремы Филиппова, см. [1], стр.259. □

3 Приведение разностной задачи к системе линейных уравнений

Вначале перепишем изначальное условие разностной задачи. Заметим, что u_0 И u_N определяются однозначно по другим компонентам u. Положим $\tilde{u} = (u_k)_{k=1,\ldots,N-1}$. Тогда мы можем переписать (2) – (4) в виде

$$A\tilde{u} + B\tilde{u} = f, \tag{12}$$

$$u_0 = 0, \tag{13}$$

$$u_{N-1} = u_N, \tag{14}$$

где $B = diag(b_1, \ldots, b_N - 1)$,

$$A = \begin{pmatrix} 2/h^2 & -1/h^2 & 0 & \ldots & 0 \\ -1/h^2 & 2/h^2 & -1/h^2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \\ \ldots & \ldots & \ldots & \ldots & \\ \ldots & \ldots & \ldots & -1/h^2 & 1/h^2 \end{pmatrix}. \tag{15}$$

Для краткости положим $A + B = D$.

В следующих двух секциях обсудим два подхода к решению этой системы

4 Метод прогонки

Данный метод является частным случаем метода Гаусса для случая трёхдиагональных матриц. Он обеспечивает время решения $O(N)$. Выпишем основные формулы.

Пусть дана СЛУ $Cy = f$, где $f = (f_0, \dots, f_N)^T$,

$$C = \begin{pmatrix} c_0 & -b_0 & 0 & 0 & \dots & \dots & \dots & \dots & \\ -a_1 & c_1 & -b_1 & 0 & \dots & \dots & \dots & \dots & \\ 0 & -a_2 & c_2 & -b_2 & \dots & \dots & \dots & \dots & \\ & \dots & \dots & \dots & \dots & -a_{N-2} & c_{N-2} & b_{N-2} & 0 \\ & \dots & \dots & \dots & \dots & 0 & -a_{N-1} & c_{N-1} & b_{N-1} \\ & \dots & \dots & \dots & \dots & 0 & 0 & -a_N & c_N \end{pmatrix}. \tag{16}$$

Идея в том, чтобы выразить

$$y_k = \alpha_k y_k + 1 + \beta_k y_k + 1, k = 0, \dots, N \tag{17}$$

После некоторых расчётов (см. [2], стр. 54-55), можно убедится, что

$$\alpha_{k+1} = \frac{b_k}{c_k - \alpha_k a_k}, \tag{18}$$

$$\beta_{k+1} = \frac{f_k + a_k \beta_k}{c_k - \alpha_k a_k}, \tag{19}$$

$$\alpha_0 = \beta_0 = 0. \tag{20}$$

Алгоритм вначале выделяет память под два массива для α и β и высчитывает числа. Далее идёт обратный ход. Начиная с $y_N = \beta_{N+1}$, последовательно ищутся все y_k по формуле (17).

5 Метод Фурье

Другой возможный метод требует предварительного знания о матрице, с которой надо работать. Поэтому мы разберём только случай $b(x) = const$, т.е. $B = bI$. Для матрицы A, опеределённой в (15), можно решить спектральную задачу и получить следующий ответ.

Theorem 5.1. *Собственные числа матрицы A - это*

$$\lambda_j = \frac{4}{h^2} \sin^2(\frac{1}{2}\pi h(j - \frac{1}{2})), j = 1, \dots, N-1 \tag{21}$$

а собственные векторы y^j - это

$$y_k^j = \sqrt{2} \sin(\pi h(j - \frac{1}{2})k). \tag{22}$$

Если ввести скалярное произведение $(u, v) = \sum_1^{N-1} u_k v_k h$, то тогда (22) образуют ортобазис в $\mathbb{R}^{N-1}$.

Вернёмся к системе $(A+B)y = f$. Разложим y по базису собственных векторов: $y = \sum_1^{N-1} c_j y^j$, $c_j = (y, y^j)$. Скалярно дмоножив обе части на y^j, получим $\lambda_j c_j + b c_j = (f, y^j)$, откуда

$$c_j = \frac{(f, y^j)}{\lambda_j + b}. \tag{23}$$

Данный метод требует аналитического исследования конкретной матрицы, так что выгода есть только в конвейерном использовании одной и той же матрицы. Время работы алгоритма составляет $O(N^2)$, что хуже метода прогонки.

Приведём доказательство устойчисвости схемы, применяя технику, показанную выше.

Theorem 5.2. *Схема* (2)-(2) *устойчива.*

Доказательство. Доказывать мы будем для случая, когда b переменно.

Примем $B = diag(b_k)$. Тогда (2) перепишется в виде

$$Ay + By = f. \tag{24}$$

По теореме 5.1 имеем

$$y = \sum_j a_j y^j, f = \sum_j c_j y^j, \tag{25}$$

где y^j - собственные вектора. Заметим, что

$$Ay = \sum_j a_j \lambda_j y^j, By = \sum_j a_j B y^j. \tag{26}$$

Скалярное умножение (24) на y^k даст $\lambda_k a_k + a_k (B y^k, y^k) = c_k$, откуда

$$a_k = \frac{c_k}{\lambda_k + (B y^k, y^k)}. \tag{27}$$

Заметим, что $(By^k, y^k) \geq 0$. Далее,

$$||y||_h^2 = \sum_j a_j^2 \leq \frac{1}{\lambda_{min}^2} c_j^2 \leq \frac{||f||_h^2}{\lambda_{min}^2}. \tag{28}$$

Заметим, что по первому замечательному пределу

$$\lambda_{min} = \frac{4}{h^2} \sin^2(\frac{1}{4}\pi h) \to \frac{\pi^2}{4} > 0, \tag{29}$$

так что имеем, что для всех h верно $||y||_h \leq M||f||_h$, где M не зависит от h. Таким образом, при небольшом изменении правой части решение не сильно меняется.

Теперь исследуем устойчивость по граничным условиям. Положим $y_0 = \epsilon_0$. Тогда при замене $z_n = y_n - \epsilon_0$ в (2) выносится слагаемое вида $b_k\epsilon_0$, которое будет ограничено в норме $||\dots||_h$, так что всё сводится к предыдущему случаю. Для другого граничного условия всё аналогично.

$\square$

6 Численное решение

Все эти методы были реализованы в программе на C++11. Кратко опишем её работу.

Для хранения массивов используется стандартный контейнер std::vector, что позволяет не следить за отчиской памяти во время работы. Данные разностной задачи содержатся в структуре task, данные же для решения системы линейных уравнений – в структуре lin_sys. Функция progonka осуществляет решение СЛУ методом прогонки, функция prepare_progonka переводит разностную задачу на язык СЛУ, а затем запускается progonka. Функция fourier осуществляет решение разностной задачи при помощи метода Фурье. При этом fourier работает только если параметр b есть константа, иаче – выводит исключение std::invalid_argument.

Результатом работы являются выходные файлы output_zero.txt, output_10k.txt и output_var.txt, состоящие из нескольких колонок. Проверка велась на уравнениях с правой частью такой, что точным решением будем $u(x) = xe^x - 2ex$. Выбирались $b = 0$, $b = 10^4$ и переменная $b(x) = \sqrt{x + 0.42}\sin(x + 0.4242)e^x$.

Опишем строение выходных файлов. Столбец Number of segments – это параметр N, L2 norm between Exact and Progonka – это L_h^2 - норма разности между точным решением и решением, полученным методом прогонки, L2 norm between Exact and Fourier – L_h^2-норма разности между точным решением и решением, полученным методом Фурье. Колонка L2 norm between Progonka and Fourier показывает разницу между двумя приближёнными решениями. Этого и предыдущего столбца нет в output_var.txt. Она должна быть равна нулю с точностью до вычислительной погрешности. Последняя колонка Factor(L2 norm / h ** 2) – это частное колонки L2 norm between Exact and Progonka и h^2. Результат в этой колонке выходит на константу с ростом N, что подтверждает факт о втором порядке точности метода.

Список литературы

[1] Н.С. Бахвалов, А.А, Корнев, Е.В. Чижонков, *Численные методы. Задачи и упражнения*, Москва, Дрофа, 2009.

[2] А.А. Корнев, *Лекции по курсу 'Численные методы'*, Москва, Издательство попечительского совета механико-математического факультета МГУ, 2018.