

Санкт-Петербургский политехнический университет
Высшая школа прикладной математики и вычислительной физики, ФизМех

Направление подготовки
«01.03.02 Прикладная математика и информатика»

Отчет по лабораторной работе № 1
тема "Метод конечных разностей для ОДУ второго порядка"
дисциплина "Конечно-разностные и сеточные методы"

Выполнил студент гр. 5030102/90101

М.К.Турченко

Преподаватель:

Б.С.Григорьев

Санкт-Петербург

2022

Оглавление

Постановка задачи	3
Разностные схемы	4
Численная реализация.....	6
Результаты вычислительного эксперимента	8
Вывод.....	12

Постановка задачи

Необходимо решить ОДУ второго порядка вида

$$-u'' + xu' + (x + 1)u = x(x + 1) \sin(\pi x) + x \sin(\pi x) + \pi x \cos(\pi x) - \pi(2 \cos(\pi x) - \pi x \sin(\pi x)), x \in [0, 1]$$

со следующими граничными условиями:

$$\begin{cases} u'(0) = 0, \\ u(1) = 0 \end{cases}$$

использовав два метода конечных разностей первого и второго порядка аппроксимации производных.

Точное решение: $u^*(x) = x \sin(\pi x)$

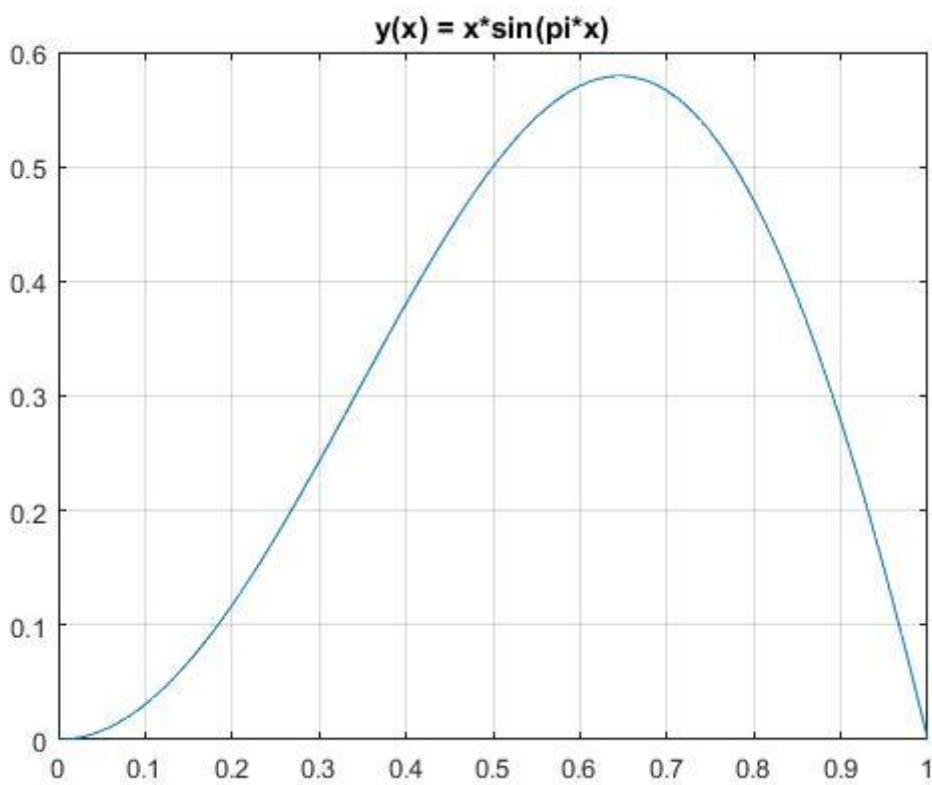


Рис. 1 График точного решения

Для сравнения эффективности методов требуется построить графики зависимостей норм вектора ошибок от величины шага сетки в одних осях, а также привести таблицы значений норм вектора ошибок при разных шагах.

Разностные схемы

Введем следующие обозначения в задаче:

$$p(x) = x,$$

$$q(x) = x + 1,$$

$$f(x) = x(x + 1) \sin(\pi x) + x \sin(\pi x) + \pi x \cos(\pi x) - \pi(2 \cos(\pi x) - \pi x \sin(\pi x))$$

Тогда задача примет вид:

$$-u'' + p(x)u' + q(x)u = f(x), x \in [0, 1]$$

$$\begin{cases} u'(0) = 0 \\ u(1) = 0 \end{cases}$$

Далее введем на отрезке $[0, 1]$ равномерную сетку с шагом h :

$$x_h = \{x_i | x_i = ih, i = 0, 1, \dots, n, h = \frac{1}{n}\}$$

На этой сетке определим сеточные функции

$$p_i = p(x_i), \quad q_i = q(x_i), \quad f_i = f(x_i)$$

Считая $u(x)$ точным решением данного ОДУ, положим $u_i \approx u(x_i)$

Фиксируя $x = x_i$, приходим к равенствам:

$$y''(x_i) + p_i y'(x_i) + q_i y(x_i) = f_i, i = 0, \dots, n$$

В каждом внутреннем узле сетки, т.е. при $i=1, \dots, n-1$, значения производных будем аппроксимировать конечноразностными отношениями.

Для производной второго порядка есть формула второго порядка точности:

$$y'' = \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1}))}{h^2} + O(h^2);$$

А формулы для первой производной будут отличаться в двух конечноразностных схемах:

$$y' = \frac{y(x_{i+1}) - y(x_{i-1}))}{2h} + O(h^2)$$

$$y' = \frac{y(x_{i+1}) - y(x_i)}{h} + O(h)$$

В схеме $O(h)$ для аппроксимации первой производной возьмем правую разностную производную, поскольку граничное условие на правом конце интервала не содержит производную, и никаких проблем с определением граничных условий там не возникнет.

Тогда мы приходим к разностным уравнениям относительно приближенных значений решения:

$$-\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} + p_i \frac{y_{i+1}-y_{i-1}}{2h} + q_i y_i = f_i \text{ для схемы } O(h^2)$$

$$-\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} + p_i \frac{y_{i+1}-y_i}{h} + q_i y_i = f_i \text{ для схемы } O(h)$$

После приведения подобных членов получаем разностные уравнения второго порядка:

$$y_{i+1} \left(-1 + \frac{p_i h}{2} \right) + y_i (2 + q_i h^2) + y_{i-1} \left(-1 - \frac{p_i h}{2} \right) = f_i h^2 \text{ для схемы } O(h^2) \quad (1)$$

$$y_{i+1} (-1 + p_i h) + y_i (2 + q_i h^2 - p_i h) + y_{i-1} (-1) = f_i h^2 \text{ для схемы } O(h)$$

$$i = 1, \dots, n-1$$

Получаем СЛАУ с трехдиагональной матрицей коэффициентов. Однако у нас $n+1$ неизвестная переменная y_0, \dots, y_n , а уравнений мы получили всего $n-1$. Еще два уравнения получим из граничных условий.

Условие на правой границе включает в себя только значение функции, а не значение производной, тогда в обеих схемах будет одинаковое условие на правом конце: $y_n = 0$

Условие на левой границе зависит от производной, тогда его надо аппроксимировать согласно порядку точности схемы:

$$y'(0) = \frac{y_1 - y_0}{h} + O(h)$$

$$y'(0) = \frac{-3y_0 + 4y_1 - y_2}{2h} + O(h^2)$$

В схеме $O(h)$ вместе с нашими граничными условиями получили СЛАУ размера $(n+1) \times (n+1)$ с трехдиагональной матрицей. Для решения такой СЛАУ подходит метод прогонки.

В схеме $O(h^2)$ трехдиагональность системы нарушена из-за граничного условия на левом конце: для трехдиагональности матрицы оно должно зависеть только от y_0 и y_1 . Для этого возьмем разностное уравнение в схеме второго порядка при $i=1$ (1), возьмем граничное условие и исключим в нем y_2 :

$$\begin{cases} y_2 \left(-1 + \frac{p_1 h}{2} \right) + y_1 (2 + q_1 h^2) + y_0 \left(-1 - \frac{p_1 h}{2} \right) = f_1 h^2 \\ 3y_0 + 4y_1 - y_2 = 0 \end{cases}$$

Тогда граничное условие на левом конце примет вид:

$$y_0 \left(-3 + \frac{2}{p_1 h - 2} \left(1 - \frac{p_1 h^2}{2} \right) \right) + y_1 \left(4 + \frac{2}{p_1 h - 2} (2 + q_1 h^2) \right) = \frac{2f_1 h^2}{p_1 h - 2}$$

Таким образом, мы восстановили трехдиагональность матрицы системы в схеме второго порядка, к которой также нужно применить метод прогонки.

Численная реализация

Программа была реализована на языке программирования C++ в среде разработки Microsoft Visual Studio 2019.

`class Equation` — основной класс программы. Содержит все поля и методы, необходимые для построения схемы

Сигнатура метода	Описание метода	Входные параметры	Возвращаемое значение
Equation(int n)	Конструктор класса	Количество разбиений отрезка	-
~Equation()	Деструктор класса	-	-
double p(double x)	Возвращает значение коэффициента перед y'	Аргумент коэффициента	Значение коэффициента в точке
double q(double x)	Возвращает значение коэффициента перед y	Аргумент коэффициента	Значение коэффициента в точке
double f(double x)	Возвращает значение свободного коэффициента	Аргумент коэффициента	Значение коэффициента в точке
double exact(double x)	Возвращает значение точного решения в точке	Аргумент функции	Значение функции в точке
double SecondNormError()	Возвращает значение согласованной второй нормы вектора решения	-	Норма вектора
void FDlin(Equation* eq)	Решает ОДУ 2 порядка МКР первого порядка	Указатель на структуру типа Equation, содержащую все данные	Вектор решений по указателю eq
void FDquad(Equation* eq)	Решает ОДУ 2 порядка МКР второго порядка	Указатель на структуру типа Equation, содержащую все данные	Вектор решений по указателю eq
int main()	Основной метод программы, выполняющий численный расчет	-	0, если программа завершена успешно. Иначе код ошибки

Результаты вычислительного эксперимента

Для сходящейся схемы при малых h будем иметь $\|z_h\| \approx Ch^k$ для сходящейся схемы порядка k . Тогда $\log\|z_h\| \approx C_1 + k\log(h)$, то есть зависимость $\log\|z_h\|$ от $\log(h)$ должна выходить на прямую с угловым коэффициентом k .

Для определения погрешности численного решения двух схем построим графики зависимостей второй нормы ошибки от величины шага в одних осях. По оси абсцисс отложим минус натуральный логарифм шага, а по оси ординат – минус натуральный логарифм нормы вектора ошибки.

Также построим в тех же осях графики прямых, проходящих с угловыми коэффициентами 1 и 2, чтобы показать, что схемы сходятся с соответствующими порядками.

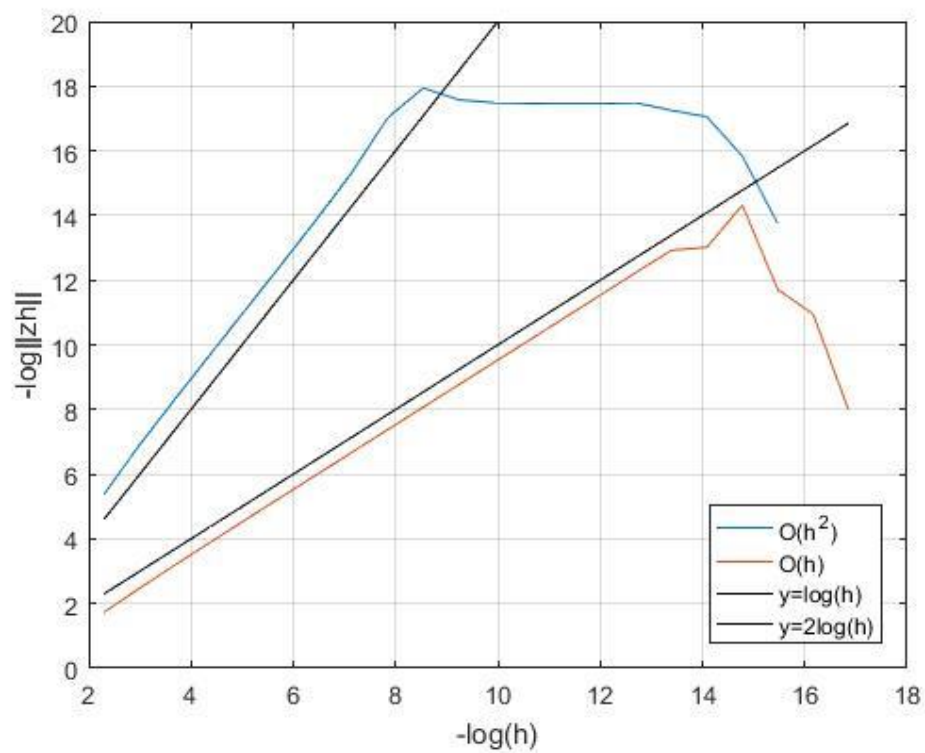


Рис. 2 Зависимость нормы вектора ошибки от длины шага в логарифмических осях

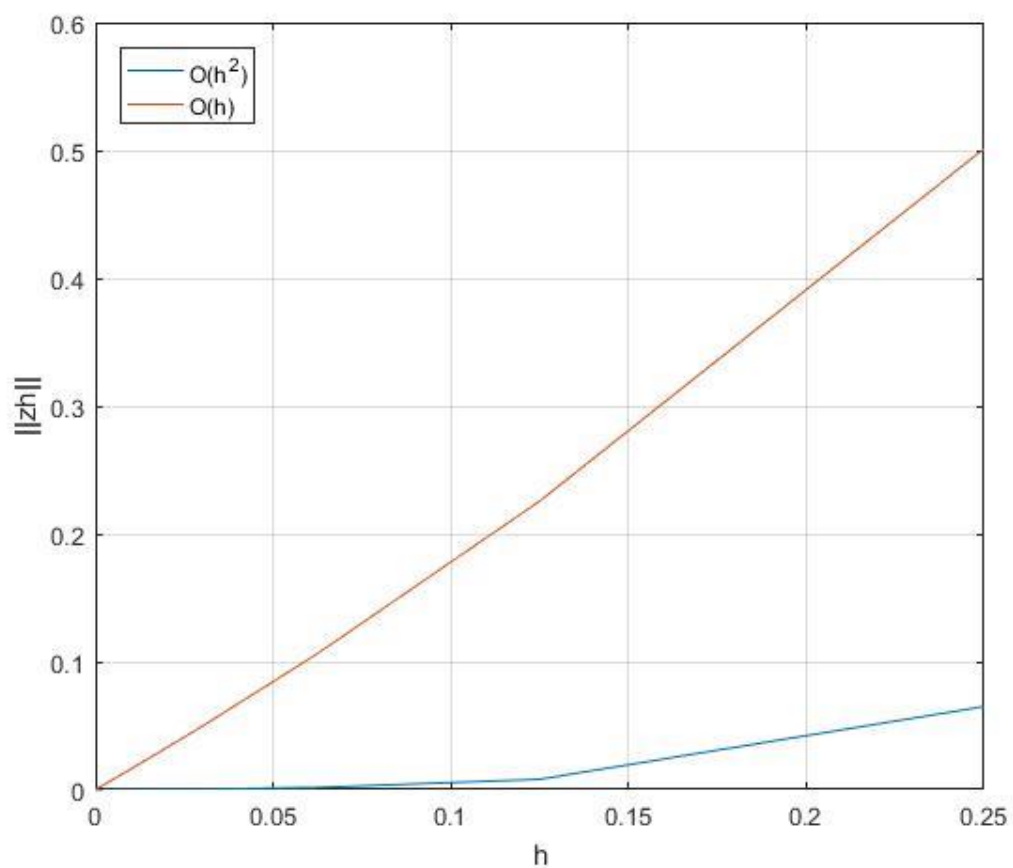


Рис. 3 Зависимость нормы вектора ошибки от длины шага

Из графиков видно, что схемы первого и второго порядка аппроксимации действительно сходятся с порядком h и h^2 соответственно. Однако когда шаг стал очень мал, схемы начали расходиться.

Ошибка решения складывается из погрешности аппроксимации производных конечно-разностной схемой и погрешности представления вещественных чисел в памяти компьютера. Пока погрешность аппроксимации велика, погрешность представления чисел незаметна, и при уменьшении шага схема сходится к точному решению. Однако затем погрешность аппроксимации становится меньше погрешности представления чисел, которую мы убрать не можем. Поэтому начиная с какого-то маленького шага схемы начинают расходиться.

Схема второго порядка разошлась раньше (при величине шага $e^{-9} \approx 1,2 * 10^{-4}$), чем схема первого порядка ($e^{-14} \approx 8,3 * 10^{-7}$)

Тем не менее схема второго порядка достигла большей точности ($e^{-18} \approx 1,5 * 10^{-8}$ норма вектора ошибки), чем схема первого порядка ($e^{-14} \approx 8,3 * 10^{-7}$)

Таблица зависимости $\|z_h\|$ от h :

Таблица 1. Зависимость нормы вектора ошибки от длины шага

Схема $O(h)$			Схема $O(h^2)$	
h	$\ z_h\ $		h	$\ z_h\ $
0.250000	0.501320		0.250000	0.0649468
0.125000	0.225714		0.125000	0.0080587
0.062500	0.106402		0.062500	0.00160122
0.031250	0.051576		0.031250	0.00038152
0.015625	0.025381		0.015625	9.46225e-05
0.007812	0.012589		0.007812	2.36428e-05
0.003906	0.006269		0.003906	5.90101e-06
0.001953	0.003128		0.001953	1.46023e-06
0.000977	0.001562		0.000977	3.49211e-07
0.000488	0.000781		0.000488	7.2437e-08
0.000244	0.000390		0.000244	1.47175e-08
0.000122	0.000195		0.000122	2.1653e-08
0.000061	0.000098		0.000061	2.50268e-08
0.000031	0.000049		0.000031	2.59468e-08
0.000015	0.000024		0.000015	2.61679e-08
0.000008	0.000012		0.000008	2.64257e-08
0.000004	0.000006		0.000004	2.77122e-08
0.000002	0.000003		0.000002	7.86665e-08
0.000001	0.000001		0.000001	1.30305e-07
0.0000005	0.000001		0.0000005	1.54206e-07
0.00000025	0.000001		0.00000025	2.42077e-07
0.000000125	0.000021		0.000000125	7.73372e-07

Вывод

Методы конечных разностей первого и второго порядка успешно решают задачу нахождения решения ОДУ 2 порядка. МКР 1 порядка оказался более устойчивым к уменьшению шага сетки. Однако при одной и той же величине шага МКР 2 порядка сошелся к более точному решению, чем МКР 1 порядка.