Отчёт по задаче 2.1.

выполнил: Дементьев Ф. А.
409 группа.

Постановка задачи

Для задачи
$$(**)\begin{cases} -u''(x) + b(x)u(x) = f(x); \; x \in [0,1], \; b(x) \geq 0 \\ u(0) = u(1) = 0; \end{cases}$$
на трёхточечном шаблоне построить разностную схему, имеющую второй порядок сходимости в $L_{2h}[0,1]$ - норме.

Для заданной правой части $f(x)$ найдём решение методом Фурье (учитывая $b(x) \equiv b$) и методом прогонки. Сравним результаты при $b(x) \equiv b$.

Метод прогонки.

Краевые условия даны следующие $\overbrace{01}^{\text{сетки}}\overbrace{00}^{\text{края}}$.

Сдвиг сетки даёт $h = \frac{2}{2N-1}$, $x_k = k \cdot h$, $k = 0, \dots, N$.

$f(0) = f(1) = 0;$

для производной второго порядка возьмём схему $\frac{-u_{k+1} + 2u_k - u_{k-1}}{h^2}$, $k = 1, \dots, N-1$.

Тогда эквивалентная разностная схема принимает вид:
$$(*)\begin{cases} -\frac{u_{k+1} - 2u_k + u_{k-1}}{h^2} + b_k y_k = f_k; \; k = 1, \dots, N-1 \\ u_0 = 0 \\ u_N = -u_{N-1} \end{cases}$$

рассматривая матричное представление, края мы получаем первой и последней строками: т.е. для $\underset{n\times n}{A}\cdot\vec{u} = \vec{f}$

$(a_{00} \dots a_{0N})(u_0 \dots u_N)^T = f_0 (=0)$, т.к. $u_0 = 0$, то

имеем $(1\ 0 \dots 0)(u_0 \dots u_N)^T = f_0$.

Аналогично $(a_{N0} \dots a_{NN})\cdot(u_0 \dots u_N)^T = f_N\ (=0)$, т.к. $u_{N-1} = -u_N$

то $(0 \dots 0\ 1\ 1)(u_0 \dots u_N)^T = f_N\ (=0)$

$u_{N-1} + u_N = 0$ – верно; $\Rightarrow$

$$(N+1)\begin{pmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2}+b_1 & -\frac{1}{h^2} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -\frac{1}{h^2} & \frac{2}{h^2}+b_{N-1} & -\frac{1}{h^2} \\ 0 & \dots & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ \vdots \\ u_{k-1} \\ u_k \\ u_{k+1} \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} f_0 \\ \vdots \\ f_k \\ \vdots \\ f_N \end{pmatrix}$$

$(N+1)$ $\qquad 1 = -a_N$

Наша матрица A – трехдиагональная.

т.е. $\begin{cases} -\frac{1}{h^2}u_{k+1} + \left(\frac{2}{h^2}+b_k\right)u_k - \frac{1}{h^2}u_{k-1} = f_k, & k = 1,\dots,N-1 \\ u_0 = 0 \\ u_{N-1} = -u_N \end{cases}$

идея метода: выразить $\boxed{u_i = \alpha_{i+1}u_{i+1} + \beta_{i+1},\ i = N-1,\dots,0}$

Используя формулы $\alpha_{i+1} = \dfrac{b_i}{c_i - a_i\alpha_i}$ и $\beta_{i+1} = \dfrac{f_i + a_i\beta_i}{c_i - a_i\alpha_i}$

для матрицы $A^* = \begin{pmatrix} c_0 & b_0 & 0 & 0 & \dots \\ -a_1 & c_1 & & & \\ 0 & \ddots & \ddots & & \\ & & & & -b_{N-1} \\ 0 & \dots & & -a_N & c_N \end{pmatrix}$; $\quad a_k = \frac{1}{h^2}$, $b_k = \frac{1}{h^2}$, $c_k = \frac{2}{h^2}+b_k$

$b_0 = 0$, $a_N = -1$, $c_N = 1$

и применяя их к нашим матрицам и обозначениям, получаем: $\alpha_{i+1} = \dfrac{\frac{1}{h^2}}{\frac{2}{h^2}+b_i - \frac{1}{h^2}\alpha_i}$, $\beta_{i+1} = \dfrac{f_i + \frac{1}{h^2}\beta_i}{\frac{2}{h^2}+b_i - \frac{1}{h^2}\alpha_i}$,

при $i = 1,\dots,N-1$

т.к. для нашей матрицы $a_N = 1, c_N = 1$, $b_0 = f_0 = f_N = 0$, $c_0 = 1$, то

$$\alpha_1 = \frac{b_0}{c_0} = 0;$$

$$\beta_1 = \frac{f_0}{c_0} = 0;$$

Записали формулы правой прогонки.

Обратным ходом прогонки, вычисляем u_N:

$$u_N = \frac{f_N + a_N \beta_N}{c_N - a_N \alpha_N} = \frac{0 + 1 \cdot \beta_N}{1 - 1 \cdot \alpha_N} = \frac{\beta_N}{1 - \alpha_N}$$

<u>Аппроксимация на решении</u>

(данная часть рассуждений о порядке сходимости имеет отношение к обоим методам, так как производная представлена одинаково)

$\exists\, u(x)$ – решение, тогда $-u''(x_k) + b(x_k) u(x_k) - f(x_k) = 0$

Рассмотрим $\left| (L_h(u)_{u_h} - f_h)|_{x = x_k} \right|$ и оценим:

$$\left| -\frac{u(x_{k+1}) - 2u(x_k) + u(x_{k-1})}{h^2} + b(x_k) u(x_k) - f(x_k) \right| \ominus$$

Разложим в ряд Тейлора в точке $x = x_k$:

$$u(x_{k\pm1}) = u(x_k \pm h) = u(x_k) \pm h \cdot u'(x_k) + u''(x_k)\frac{h^2}{2} \pm u'''(x_k)\frac{h^3}{6} + u^{IV}(\xi_k) h^4 \cdot \frac{1}{24};$$

$$\ominus \left| -\frac{\cancel{2u(x_k)} + 2u''(x_k)\frac{h^2}{2} + 2u^{IV}(\xi_k)\frac{h^4}{24} - \cancel{2u(x_k)}}{h^2} + b(x_k) u(x_k) - f(x_k) \right| =$$

$$= \left| \underbrace{-u''(x_k) + b(x_k) u(x_k) - f(x_k)}_{=0} + \frac{1}{12} u^{IV}(\xi_k) \cdot h^2 \right| = \underline{O}(h^2), \quad k = 1, \dots, N-1$$

на границах $u(x_0) = u(x_N) = 0$;

$\Rightarrow$ Разностная схема (*) аппроксимирует на решении задачу Коши (**) с порядком аппроксимации $\underline{p = 2}$;

Устойчивость:

! (рассуждения об устойчивости применимы к обоим методам)

Устойчивость схемы покажем методом априорных оценок:

в схеме ⊛ k-е уравнение умножим на u_k и возьмём сумму по $k = 1, .., N-1$:

получим: $-\frac{1}{h^2}(u_2 - u_1)u_1 + \frac{1}{h^2}(u_1 - u_0)u_1 - \frac{1}{h^2}(u_3 - u_2)u_2 + $

$+ \frac{1}{h^2}(u_2 - u_1)u_2 - \ldots = -\frac{1}{h^2}\sum_{i=1}^{N-1}(u_{i+1} - u_i)u_i + \frac{1}{h^2}\sum_{i=1}^{N-1}(u_i - u_{i-1})u_i =$

$= \langle$т.к. $u_0 = 0$, сдвинем индекс вниз ~~добавим слагаемое~~$\rangle = \langle$т.к. $u_N = 0$, добавим во втором$\rangle =$

$$= -\frac{1}{h^2}\sum_{i=1}^{N}(u_i - u_{i-1})u_{i-1} + \frac{1}{h^2}\sum_{i=1}^{N}(u_i - u_{i-1})u_i =$$

$$= +\frac{1}{h^2}\sum_{i=1}^{N}\left[+u_{i-1}^2 - u_i u_{i-1} - u_{i-1}u_i + u_i^2\right] = \frac{1}{h^2}\sum_{i=1}^{N}(u_i - u_{i-1})^2$$

$$\Rightarrow \quad \frac{1}{h^2}\sum_{i=1}^{N}(u_i - u_{i-1})^2 + \sum_{i=1}^{N-1}p_i u_i^2 = \sum_{i=1}^{N-1}f_i u_i$$

обозначив $\nabla u_i = u_i - u_{i-1}$, $(u, v] = \sum_{i=1}^{N}u_i v_i$, $(u, v) = \sum_{i=1}^{N-1}u_i v_i$ $\Rightarrow \frac{1}{h^2}\sum_{i=1}^{N}\nabla u_i^2 + (pu, u) = (f, u)$

$$\Rightarrow \quad \frac{1}{h^2}(\nabla u, \nabla u) + (pu, u) = (f, u).$$

Теперь выводим сеточный аналог:

$u_k = \sum_{i=1}^{k}(u_i - u_{i-1})$, значит $\forall k$: $1 \le k \le N-1$

выполнено нер-во: $u_k^2 = \left(\sum_{i=1}^{k}(u_i - u_{i-1})\right)^2 \le \underbrace{\sum_{i=1}^{k}1\cdot 1}_{k}\cdot\sum_{i=1}^{k}(u_i - u_{i-1})^2 \le$

$\le N\cdot\sum_{i=1}^{N}(u_i - u_{i-1})^2$;

Теперь беру сумму по k: $\sum_{k=1}^{N-1}u_k^2 \le \underset{N-1}{k}\cdot\sum_{i=1}^{N}(u_i - u_{i-1})^2 \le$

$\le N^2\sum_{i=1}^{N}(u_i - u_{i-1})^2 \underset{Nh=1}{\le} \frac{1}{h^2}\sum_{i=1}^{N}(u_i - u_{i-1})^2 + \sum_{i=1}^{N-1}p_i u_i^2 =$

$= \sum_{i=1}^{N-1} f_i u_i \le \frac{1}{2}\left(\sum_{i=1}^{N-1} f_i^2 + \sum_{i=1}^{N-1} u_i^2\right)$, т.е. $2\sum_{k=1}^{N-1} u_k^2 \le \sum_{i=1}^{N-1} f_i^2 + \sum_{i=1}^{N-1} u_i^2$

$\Rightarrow \sum_{i=1}^{N-1} u_i^2 \le \sum_{i=1}^{N-1} f_i^2$, и, значит, априорная оценка решения разностной задачи в норме $\|u_h\|_h^2 = h(u_h, u_h)$ пространства $L_{2,h}$, согласованной с непрерывной нормой L_2 будет иметь вид $\|u_h\|_h \le \|f_h\|_h$, $\Rightarrow$ устойчивость!

Значит, при небольших колебаниях правой части (т.е. $f(x)$) и граничных условий, решение меняется тоже на малую величину.

Сходимость.

Обратимся к Th. Филиппова:

Если выполнены условия:

1. Схемы (*) и (**) задаются лин. операторами;
2. Решение (**) $\exists!$;
3. Разностная схема (*) аппроксимирует на решении задачу (**) с порядком p; // ($p=2$)
4. Схема (*) устойчива;

Тогда решение разностной схемы (*) сходится к решению дифф. задачи (**) с порядком не ниже p. // т.е. с порядком ≥ 2.

Значит решение нашей схемы (*) будет сходится к решению задачи (**) с порядком ≥ 2.

Метод Фурье

Аппроксимация на решении, сходимость и устойчивость обоснованы для схемы:

$$(*)\begin{cases} -\dfrac{U_{k+1} - 2U_k + U_{k-1}}{h^2} + b_k U_k = f_k, & k = 1, \dots, N-1 \\ U_0 = 0 & Nh = 1; \quad h = \frac{2}{2N-1} \\ U_N = -U_{N-1} \end{cases}$$

Схема в этом методе не меняется. Различие лишь в том, что $b_k \equiv b \neq b(x)$, но рассуждения от этого силу не теряют. Поэтому, перейдём к описанию метода: для матричного уравнения $A \cdot U = f$, где $U, f \in \mathbb{R}^N$, мы знаем все собственные векторы: $\{\varphi^k = \sqrt{2}\sin(\pi k x)\}_{k=1}^{N-1}$ и все собственные числа: $\{\lambda_k = \frac{2}{h^2} - \frac{2}{h^2}\cos(\pi k x)\}_{k=1}^{N-1}$ и система $\{\varphi^k\}_{k=1}^{N-1}$ образует ортонормированный базис относительно скалярного произведения $(\varphi^{(m)}, \varphi^{(n)}) = \sum_{i=1}^{N-1} \varphi_i^{(m)} \varphi_i^{(n)} h$:

Необходимо разложить $U(x)$ — решение по базису, т.е. будем искать решение задачи в виде $U = \sum_{i=1}^{N-1} c_i \varphi^{(i)}$; если подставить в задачу, то

$$A \cdot \left(\sum_{i=1}^{N-1} c_i \varphi^{(i)}\right) = \sum_{i=1}^{N-1} c_i \lambda^{(i)} \varphi^{(i)} = f$$

при скалярном домножении на $\varphi^{(m)}$, $m = 1, \dots, N-1$

получим $$C_m \lambda^{(m)} = (f, \varphi^{(m)})$$

$$\Rightarrow C_m = \frac{(f, \varphi^{(m)})}{\lambda^{(m)}},$$ а далее можем собрать решение.

В нашем случае матрица A это $A = C + B$;

$$\begin{pmatrix} 1 & 0 & \dots & \dots & 0 & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2}+b & -\frac{1}{h^2} & b & \dots & 0 \\ \vdots & & \ddots & & & 0 \\ 0 & 0 & \dots & \dots & 1 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -\frac{1}{h^2} & \frac{2}{h^2}+b & -\frac{1}{h^2} & 0 \dots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & & \\ 0 & \dots & 0 & -\frac{1}{h^2} & \frac{2}{h^2}+b & -\frac{1}{h^2} \\ 0 & \dots & 0 & 0 & 1 & 1 \end{pmatrix}}_{C} + \underbrace{\begin{pmatrix} 0 & 0 & & 0 \\ 0 & b & & \vdots \\ \vdots & & \ddots & \vdots \\ & & b & 0 \\ 0 & \dots & 0 & 0 \end{pmatrix}}_{B} \Rightarrow$$

собственные числа матрицы $C = \{\lambda_k\}_{k=1}^{N-1}$

=> собственные числа матрицы $A = \{\lambda_k + B\}_{k=1}^{N-1}$

а собственные функции у них совпадают.

Работа программы:

Используются функции:

- double eigen_fun (int k, int i, double h) – k-ая собственная функция
- double eigen_val (int k, double h, double B) – выдаёт k-е собственное значение с поправкой +B.
- void f2C (double *C, double *f, double b, int n) раскладывает векторно f по базису из собственных ф-ий с коэффициентами C[i];
- void C2f (double *C, double *new f, int n) собирает ф-ю по коэффициентам, т.е. строит решение сеточной задачи

для метода Фурье

- void run (double *a, double *b, double *c, double *alfa, double *betta, double *new Y, double *f, int N) сама функция прогонки.
- double solution (double x) – точное решение
- double funB (double x) – B(x) – ~~сето~~
- double fun_f (double x) – правая часть.
- void massiv (double *a, double *b, double *c, double *Bx, double *Y, double *f, int N) – считает все исходные данные в массивы для прогонки, считает сеточные аналоги функциям.
- double errorNorm (double *x, double *y, double *error, int N) считает вектор отклонений $\vec{y}$ от $\vec{x}$ покоординатно и норму $L_{2,h}$.

Численная проверка:

- Метод Фурье отдельно реализован в Fur1.cpp

на данных: $u(x) = x^2 - x$; - точное решение
$f(x) = 3x^2 - 3x - 2$; - правая часть
$b(x) \equiv 3$;

получим результат на выходе значение

$$res1 = \frac{\|error\|_{L_2 h}}{h^2} = \frac{\sqrt{\sum_{i=1}^{N} error^2[i]}}{h^2}$$, error[i] - вектор ошибки

при N = 10	0,150725096139
при N = 100	0,1252055198
при N = 1000	0,12288640663
при N = 10000	0,13310338 1845

- Метод прогонки отдельно представлен в progon1.cpp

аналогичная величина даёт результаты:

при N = 10	0,163022762498
при N = 100	0,14092801 7979
при N = 1000	0,1388250 8813
при N = 10000	0,138705047105
при N = 100000	0,2263227583184

при
$u(x) = x^2 - x$
$f(x) = x^3 - x^2 - 2$
$b(x) = x$

Вывод: погрешность имеет порядок $O(h^2)$.

- Для сравнения результатов, в Task22.cpp

реализованы оба метода, выведены три величины:

для прогонки res2 : $0{,}00177975 = \sqrt{\sum_{i=1}^{N} error^2[i]}$

для Фурье res1 : $0{,}0017797529 = \sqrt{\sum_{i=1}^{N} error^2[i]}$

max отклонение между решениями : 2×10^{-16}

$\max\limits_{0 \le k \le N} |u_k^1 - u_k^2|$

на данных: $u(x) = x^2 - x$,
$f(x) = x^2 - x - 2$;
$b(x) \equiv 1$;