БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ

Лабораторная работа №1
«Способы построения и исследования разностных схем»
по дисциплине «Численные методы математической физики»

Вариант 13

Выполнил:
Черепенников Р.М.
студент 3 курс 8 группы

Проверил:
Будник А.М.
Доцент кафедры вычислительной
математики ФПМИ

Минск 2022

Постановка задачи

Дана третья краевая задача для ОДУ второго порядка следующего вида:

$$(k(x)u'(x))' - q(x)u(x) = -f(x),\ 0 \le x \le 1$$

$$k(0)u'(0) = æ_0 u(0) - g_0$$

$$-k(1)u'(1) = æ_1 u(1) - g_1$$

где

$f(x)$	$k(x)$	$q(x)$	$æ_0$	g_0	$æ_1$	g_1
e^x	$sin^2(x) + 1$	$cos(x)$	1	1	1	1

Необходимо:

1. Используя разностные операторы вместо дифференциальных, аппроксимировать поставленную задачу разностной схемой 2-го порядка на минимальном шаблоне. Для повышения порядка аппроксимации граничных условий выделить главный член погрешности и заменить его, используя следующий вид исходного дифференциального уравнения
   $k(x)u''(x) + k'(x)u'(x) - q(x)u(x) = -f(x)$
2. Интегро-интерполяционным методом построить консервативную разностную схему. Для вычисления коэффициентов этой схемы использовать формулу средних прямоугольников.
3. Аппроксимировать исходную задачу вариационно-разностным методом. Для вычисления коэффициентов полученной схемы использовать формулу трапеций.
4. Методом разностной прогонки реализовать полученные в п.п. 1-3 разностные схемы при $h = 0.1$. Провести сравнительный анализ сеточных решений при разных шагах.

1. Аппроксимация исходной задачи разностной схемой

1.1 Теория

Аппроксимируем дифференциальную задачу. Так как в ней содержится вторая производная $u(x)$, то минимальный шаблон имеет вид
$Ш(x) = \{x - h, x, x + h\}$.
Для левого и правого граничных условий минимальный шаблон будет иметь вид $Ш_л(0) = \{0, h\}$, $Ш_п(1) = \{1 - h, 1\}$

Рассмотрим исходное уравнение в виде:

$$k(x)u''(x) + k'(x)u'(x) - q(x)u(x) = -f(x)$$

Его можно аппроксимировать следующим образом:

$$ku_{\bar{x}x} + k'u_{\overset{\circ}{x}} - q(x)u = -f$$

Перейдем к аппроксимации граничных условий. Рассмотрим левое граничное условие

$$-k(1)u'(1) = æ_1 u(1) - g_1$$

Его можно аппроксимировать следующим образом:

$$-k(1)y_{\bar{x}}(1) = æ_1 y(1) - g_1 \text{ на шаблоне } Ш(x) = \{1 - h, 1\}$$

Однако такая аппроксимация будет иметь погрешность $\psi(h) = O(h)$. Для того чтобы схема имела второй порядок точности необходимо повысить порядок аппроксимации граничных условий. Этого можно добиться с помощью выбора коэффициентов $\tilde{æ}_1$ и $\tilde{g}_1$.

Погрешность будет иметь следующий вид:

$$\psi = -k(1)\frac{u(1) - u(1-h)}{h} - \tilde{æ}_1 u(1) + \tilde{g}_1 =$$

$$= -k(1)\frac{u(1) - (u(1) - hu'(1) + \frac{h^2}{2}u''(1) + O(h^3))}{h} - \tilde{æ}_1 u(1) + \tilde{g}_1 =$$

$$= -k(1)[u'(1) - \frac{h}{2}u''(1) + O(h^2)] - \tilde{æ}_1 u(1) + \tilde{g}_1 =$$

$$= [\text{из гр. условий имеем } -k(1)u'(1) = æ_1 u(1) - g_1$$

$$\text{и вида уравнения } k(1)u''(1) = -k'(1)u'(1) + q(1)u(1) - f(1) =$$

$$= \frac{k'(1)}{k(1)}(æ_1 u(1) - g_1) + q(1)u(1) - f(1)] =$$

$$= æ_1 u(1) - g_1 + \frac{h}{2}[\frac{k'(1)}{k(1)}(æ_1 u(1) - g_1) + q(1)u(1) - f(1)] - \tilde{æ}_1 u(1) + \tilde{g}_1 +$$

$$+ O(h^2) = u(1)[æ_1 + \frac{h}{2}\frac{k'(1)}{k(1)}æ_1 + \frac{h}{2}q(1) - \tilde{æ}_1] +$$

$$+ [- g_1 - \frac{h}{2}\frac{k'(1)}{k(1)} g_1 - \frac{h}{2}f(1) + \widetilde{g}_1] + O(h^2)$$

Соответственно, при

$$\widetilde{æ}_1 = æ_1(1 + \frac{h}{2}\frac{k'(1)}{k(1)}) + \frac{h}{2}q(1)$$

$$\widetilde{g}_1 = g_1(1 + \frac{h}{2}\frac{k'(1)}{k(1)}) + \frac{h}{2}f(1)$$

Погрешность $\psi = O(h^2)$

Действуя аналогично для правого граничного условия получим:

$$\widetilde{æ}_0 = æ_0(1 - \frac{h}{2}\frac{k'(0)}{k(0)}) + \frac{h}{2}q(0)$$

$$\widetilde{g}_0 = g_0(1 - \frac{h}{2}\frac{k'(0)}{k(0)}) + \frac{h}{2}f(0)$$

В общем, получаем следующую разностную схему для исходной задачи:

$$k_i\frac{y_{i+1}-2y_i+y_{i+1}}{h^2} + k'_i\frac{y_{i+1}-y_{i-1}}{2h} - q_iy_i = -f_i,\ i = \overline{1,\ N-1}$$

$$k_0\frac{y_1 - y_0}{h} = \widetilde{æ}_0 y_0 - \widetilde{g}_0$$

$$-k_N\frac{y_N - y_{N-1}}{h} = \widetilde{æ}_1 y_N - \widetilde{g}_1$$

1.2 Реализация метода

Для вычисления значений функции в узлах сетки необходимо решить следующую СЛАУ:

$$y_{i-1}(\frac{k_i}{h^2} - \frac{k'_i}{2h}) + y_i(-\frac{2k_i}{h^2} - q_i) + y_{i+1}(\frac{k_i}{h^2} + \frac{k'_i}{2h}) = -f_i,\ i = \overline{1,\ N-1}$$

$$y_0(-\frac{k_0}{h} - \widetilde{æ}_0) + y_1\frac{k_0}{h} = -\widetilde{g}_0$$

$$y_{N-1}\frac{k_N}{h} + y_N(-\frac{k_N}{h} - \widetilde{æ}_1) = -\widetilde{g}_1$$

Код программы:

```
def sweep_method(
    e: np.ndarray,
    d: np.ndarray,
    c: np.ndarray,
    b: np.ndarray
) -> np.ndarray:
    n = d.shape[0]
    for i in range(n):
```

```
        if i==0:
            if abs(d[i]) > abs(e[i]):
                warn('Не выполнено достаточное условие сходимости метода прогонки')
                break
        if i==n-1:
            if abs(d[i]) > abs(c[i]):
                warn('Не выполнено достаточное условие сходимости метода прогонки')
                break
        elif abs(d[i]) > abs(e[i]) + abs(c[i]):
            warn('Не выполнено достаточное условие сходимости метода прогонки')
            break


    x = np.zeros_like(d)
    for k in range(1, n):
        d[k] = d[k] - e[k-1]*c[k]/d[k-1]
        b[k] = b[k] - b[k-1]*c[k]/d[k-1]
    x[n-1] = b[n-1]/d[n-1]
    for k in reversed(range(n-1)):
        x[k] = (b[k] - e[k]*x[k+1])/d[k]

    return x

def k(x):
    return np.sin(x) ** 2 + 1

def dk(x):
    return 2*np.sin(x)*np.cos(x)

def f(x):
    return np.exp(x)

def q(x):
    return np.cos(x)

kappa_0 = 1
g_0 = 1
kappa_1 = 1
g_1 = 1

def solve_approx(k, dk, q, f, kappa_0, g_0, kappa_1, g_1, h=0.1):
    X = np.arange(0, 1 + h/2, h)
    n = X.shape[0]
    Q = q(X)
    F = f(X)
    A = np.array([k(x - 0.5 * h) for x in X])
    KAPPA_0 = kappa_0 * (1 - h/2 * dk(0)/k(0)) + h/2 * q(0)
    KAPPA_1 = kappa_1*(1 + h/2 * dk(1)/ k(1)) + h/2 * q(1)
    G_0 = g_0 * (1 - h/2 * dk(0)/k(0)) + h/2 * q(0)
    G_1 = g_1*(1 + h/2 * dk(1)/ k(1)) + h/2 * f(1)

    diag_el = np.zeros(n)
```

```
diag_el[0] = (-k(0)/h - KAPPA_0)
diag_el[-1] = (-k(1)/h - KAPPA_1)
for i in range(1, n-1):
    diag_el[i] = -(A[i] + A[i+1])/(h**2) - Q[i]

up_diag = np.zeros(n)
up_diag[0] = k(0)/h
for i in range(1, n-1):
    up_diag[i] = A[i+1]/(h**2)

low_diag = np.zeros(n)
low_diag[-1] = k(1) / h
for i in range(1, n-1):
    low_diag[i] = A[i]/(h**2)

neodnorodnost = np.zeros(n)
neodnorodnost[0] = -G_0
neodnorodnost[-1] = -G_1
for i in range(1, n-1):
    neodnorodnost[i] = -F[i]

return X, sweep_method(up_diag, diag_el, low_diag, neodnorodnost)
```

---

2. Интегро-интерполяционный метод

2.1 Теория

Рассматриваемое нами уравнение можно трактовать как уравнение стационарного распределения температуры $u(x)$ в стержне $0 \le x \le 1$ с коэффициентом теплопроводности $k(x)$.

Закон сохранения тепла (уравнение баланса) на отрезке $x^{(1)} \le x \le x^{(2)}$ имеет следующий вид:

$$W(x^{(1)}) - W(x^{(2)}) - \int_{x^{(1)}}^{x^{(2)}} q(x)u(x)dx + \int_{x^{(1)}}^{x^{(2)}} f(x)dx = 0$$

где $W(x) = -k(x)u'(x)$ – тепловой поток.

Введем коэффициенты:

$$a_i = h\left[\int_{x_{i-1}}^{x_i} \frac{1}{k(x)}dx\right]^{-1}, \ \phi_i = \frac{1}{h}\int_{x_{i-0.5}}^{x_{i+0.5}} f(x)dx, \ d_i = \frac{1}{h}\int_{x_{i-0.5}}^{x_{i+0.5}} q(x)dx$$

Тогда получим следующую разностную задачу:

$$a_{i+1}u_{\bar{x},i+1} - a_i u_{\bar{x},i} - hd_i u_i + h\phi_i \approx 0$$

Перейдем к аппроксимации граничных условий. В качестве примера рассмотрим левое граничное условие. Запишем уравнение баланса для отрезка $[1 - \frac{h}{2}, 1]$

$$W_{1-\frac{h}{2}} - W_1 - \int_{1-\frac{h}{2}}^{1} q(x)u(x)dx + \int_{1-\frac{h}{2}}^{1} f(x)dx = 0$$

Из граничных условий имеем $W_1 = -k(1)u'(1) = æ_1 u(1) - g(1)$, из полученного ранее $W_{1-\frac{h}{2}} = -a_N(y_{\bar{x},N})$. А также заменяя в первом интеграле $u(x)$ на интерполяционный многочлен $u(x) \approx P_0(x) = u_N$ на отрезке $[1 - \frac{h}{2}, 1]$ получаем следующую аппроксимацию граничного условия:

$$-a_N y_{\bar{x},N} = (æ_1 + \frac{h}{2}d_N)y_N - (g_1 + \frac{h}{2}\phi_N)$$

$$d_N = \frac{2}{h}\int_{1-\frac{h}{2}}^{1} q(x)dx, \ \phi_N = \frac{2}{h}\int_{1-\frac{h}{2}}^{1} f(x)dx$$

Аналогично для правого граничного условия получаем следующую аппроксимацию:

$$a_1 y_{x,0} = (æ_0 + \frac{h}{2} d_0) y_0 - (g_0 + \frac{h}{2} \phi_0)$$

$$d_0 = \frac{2}{h} \int_0^{\frac{h}{2}} q(x)dx, \ \phi_0 = \frac{2}{h} \int_0^{\frac{h}{2}} f(x)dx$$

В общем, имеем следующую разностную схему (в индексном виде):

$$\frac{1}{h} (a_{i+1} \frac{y_{i+1} - y_i}{h} - a_i \frac{y_i - y_{i-1}}{h}) - d_i y_i = - \phi_i$$

$$a_i = h[\int_{x_{i-1}}^{x_i} \frac{1}{k(x)} dx]^{-1}, \ \phi_i = \frac{1}{h} \int_{x_{i-0.5}}^{x_{i+0.5}} f(x)dx, \ d_i = \frac{1}{h} \int_{x_{i-0.5}}^{x_{i+0.5}} q(x)dx$$

$$a_1 (\frac{y_1 - y_0}{h}) = (æ_0 + \frac{h}{2} d_0) y_0 - (g_0 + \frac{h}{2} \phi_0)$$

$$d_0 = \frac{2}{h} \int_0^{\frac{h}{2}} q(x)dx, \ \phi_0 = \frac{2}{h} \int_0^{\frac{h}{2}} f(x)dx$$

$$- a_N (\frac{y_N - y_{N-1}}{h}) = (æ_1 + \frac{h}{2} d_N) y_N - (g_1 + \frac{h}{2} \phi_N)$$

$$d_N = \frac{2}{h} \int_{1-\frac{h}{2}}^{1} q(x)dx, \ \phi_N = \frac{2}{h} \int_{1-\frac{h}{2}}^{1} f(x)dx$$

2.2 Реализация метода

По условию, для вычисления интегралов необходимо использовать формулу средних прямоугольников, тогда коэффициенты разностной схемы будут иметь следующий вид:

$$a_i = h[\int_{x_{i-1}}^{x_i} \frac{1}{k(x)} dx]^{-1} = h[h \frac{1}{k_{i-0.5}}]^{-1} = k_{i-0.5}$$

$$\phi_i = \frac{1}{h} \int_{x_{i-0.5}}^{x_{i+0.5}} f(x)dx = f_i$$

$$d_i = \frac{1}{h} \int_{x_{i-0.5}}^{x_{i+0.5}} q(x)dx = q_i$$

$$d_0 = \frac{2}{h}\int_0^{\frac{h}{2}} q(x)dx = q(\frac{h}{4}) , \ \phi_0 = \frac{2}{h}\int_0^{\frac{h}{2}} f(x)dx = f(\frac{h}{4})$$

$$d_N = \frac{2}{h}\int_{1-\frac{h}{2}}^{1} q(x)dx = q(1 - \frac{h}{4}), \ \phi_N = \frac{2}{h}\int_{1-\frac{h}{2}}^{1} f(x)dx = f(1 - \frac{h}{4})$$

В результате получаем следующую СЛАУ:

$$y_{i-1}\frac{a_i}{h^2} + y_i(-\frac{a_{i+1} + a_i}{h^2} - d_i) + y_{i+1}\frac{a_{i+1}}{h^2} = -\phi_i$$

$$y_0(-\frac{a_1}{h} - (æ_0 + \frac{h}{2}d_0)) + y_1\frac{a_1}{h} =- g_0 - \frac{h}{2}\phi_0$$

$$y_{N-1}(\frac{a_N}{h}) + y_N(-\frac{a_N}{h} - (æ_1 + \frac{h}{2}d_N)) =- g_1 - \frac{h}{2}\phi_N$$

Код программы:

```
def solve_balance(k, q, f, kappa_0, g_0, kappa_1, g_1, h=0.1):
    X = np.arange(0, 1 + h/2, h)
    n = X.shape[0]
    A = np.array([k(x - 0.5 * h) for x in X])
    PHI = f(X)
    PHI[0] = f(h/4)
    PHI[-1] = f(1 - h/4)
    D = q(X)
    D[0] = q(h/4)
    D[-1] = q(1 - h/4)


    diag_el = np.zeros(n)
    diag_el[0] = -A[1]/h - (kappa_0 + h/2 * D[0])
    diag_el[-1] = -A[-1]/h - (kappa_1 + h/2 * D[-1])
    for i in range(1, n-1):
        diag_el[i] = -(A[i] + A[i+1])/(h**2) - D[i]

    up_diag = np.zeros(n)
    up_diag[0] = A[1]/h
    for i in range(1, n-1):
        up_diag[i] = A[i+1]/(h**2)

    low_diag = np.zeros(n)
    low_diag[-1] = A[-1]/h
    for i in range(1, n-1):
        low_diag[i] = A[i]/(h**2)

    neodnorodnost = np.zeros(n)
    neodnorodnost[0] = -g_0 - h * PHI[0]/2
```

```
neodnorodnost[-1] = -g_1 - h*PHI[-1]/2

for i in range(1, n - 1):
    neodnorodnost[i] = -PHI[i]
return X, sweep_method(up_diag, diag_el, low_diag, neodnorodnost)
```

3. Вариационно-разностный метод

3.1 Теория

Формулы данного метода будут иметь такой же вид как формулы метода баланса со следующими коэффициентами:

$$a_i = \frac{1}{h}\left[\int\limits_{x_{i-1}}^{x_i} k(x)dx - \int\limits_{x_{i-1}}^{x_i} q(x)(x_i - x)(x - x_{i-1})dx\right],\ i = \overline{1,\ N}$$

$$d_i = \frac{1}{h^2}\left[\int\limits_{x_{i-1}}^{x_i} q(x)(x - x_{i-1})dx + \int\limits_{x_i}^{x_{i+1}} q(x)(x_{i+1} - x)dx\right],\ i = \overline{1,\ N-1}$$

$$\phi_i = \frac{1}{h^2}\left[\int\limits_{x_{i-1}}^{x_i} f(x)(x - x_{i-1})dx + \int\limits_{x_i}^{x_{i+1}} f(x)(x_{i+1} - x)dx\right],\ i = \overline{1,\ N-1}$$

$$d_0 = \frac{2}{h^2}\int\limits_0^h q(x)(h - x)dx \qquad d_N = \frac{2}{h^2}\int\limits_{1-h}^1 q(x)(x - 1 + h)dx$$

$$\phi_0 = \frac{2}{h^2}\int\limits_0^h f(x)(h - x)dx \qquad \phi_N = \frac{2}{h^2}\int\limits_{1-h}^1 f(x)(x - 1 + h)dx$$

3.2 Реализация метода

По условию, для вычисления интегралов необходимо использовать формулу трапеций, тогда коэффициенты разностной схемы будут иметь следующий вид:

$$a_i = \frac{k_i + k_{i-1}}{2},\ i = \overline{1,\ N}$$

$$d_i = q_i,\ i = \overline{1,\ N-1}$$

$$\phi_i = f_i,\ i = \overline{1,\ N-1}$$

$$d_0 = q_0,\ d_N = q_N$$

$$\phi_0 = f_0,\ \phi_N = f_N$$

Код программы:

```
def solve_ritz(k, q, f, kappa_0, g_0, kappa_1, g_1, h=0.1):
    X = np.arange(0, 1 + h/2, h)
    n = X.shape[0]
    A = np.array([(k(x) + k(x-h))/2 for x in X])
    PHI = f(X)
    PHI[0] = f(0)
    PHI[-1] = f(1)
    D = q(X)
```

```
D[0] = q(0)
D[-1] = q(1)

diag_el = np.zeros(n)
diag_el[0] = -A[1]/h - (kappa_0 + h/2 * D[0])
diag_el[-1] = -A[-1]/h - (kappa_1 + h/2 * D[-1])
for i in range(1, n-1):
    diag_el[i] = -(A[i] + A[i+1])/(h**2) - D[i]

up_diag = np.zeros(n)
up_diag[0] = A[1]/h
for i in range(1, n-1):
    up_diag[i] = A[i+1]/(h**2)

low_diag = np.zeros(n)
low_diag[-1] = A[-1]/h
for i in range(1, n-1):
    low_diag[i] = A[i]/(h**2)

neodnorodnost = np.zeros(n)
neodnorodnost[0] = -g_0 - h * PHI[0]/2
neodnorodnost[-1] = -g_1 - h*PHI[-1]/2

for i in range(1, n - 1):
    neodnorodnost[i] = -PHI[i]
return X, sweep_method(up_diag, diag_el, low_diag, neodnorodnost)
```

4. Сравнительный анализ полученных решений

В результате получены следующие решения.

Пункт 1:

[1.2379 1.2629 1.2889 1.3143 1.3376 1.3573 1.3721 1.3808 1.3824 1.3759
 1.36]

Интегро-интерполяционный метод:

[1.2382 1.263 1.2889 1.3141 1.3373 1.3569 1.3715 1.3802 1.3817 1.375
 1.3591]

Вариационно-разностный метод:

[1.2387 1.2637 1.2896 1.315 1.3383 1.3581 1.3729 1.3817 1.3834 1.3769
 1.3611]

В качестве точного решения для сравнения результатов, возьмем решение полученное интегро-интерполяционным метод при шаге $h = 10^{-6}$:

[1.2386 1.2634 1.2893 1.3145 1.3377 1.3572 1.3719 1.3805 1.3819 1.3752
 1.3592]

Тогда в каждой точке погрешность методов будет следующая:

Пункт 1:

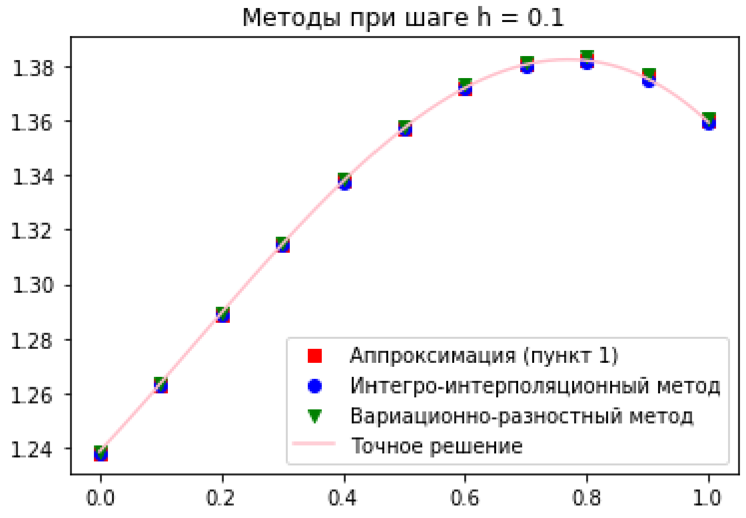
[6.8384e-04 5.3615e-04 3.7700e-04 2.1678e-04 6.2786e-05 8.2317e-05
 2.2003e-04 3.5464e-04 4.9191e-04 6.3811e-04 7.9972e-04]

Интегро-интерполяционный метод:

[0.0004 0.0004 0.0004 0.0004 0.0004 0.0004 0.0003 0.0003 0.0003 0.0002
 0.0001]

Вариационно-разностный метод:

[0.0001 0.0002 0.0004 0.0005 0.0007 0.0008 0.001 0.0012 0.0014 0.0017
 0.0019]



Все методы имеют второй порядок точности о чем свидетельствует погрешность решений (совпадают как минимум два знака после запятой), необходимо отметить и тот факт, что погрешность при решении накапливается. А также уменьшение шага уменьшает погрешность.