Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Курсовая Работа

Разработка программного обеспечения для воспроизведения на ЭВМ двумерных стационарных моделей
по дисциплине
«Математические модели систем
с распределёнными параметрами»

Выполнил: Ферапонтов М.В.
Группа: гр. 3530904/00104

Проверил: Воскобойников С. П.

Санкт-Петербург
2023

Содержание

1 Вступление **2**
- 1.1 Постановка задачи . . . 2

2 Разностная схема **3**
- 2.1 Внутренние точки . . . 3
- 2.2 На левой границе . . . 4
- 2.3 На правой границе . . . 5
- 2.4 На нижней границе . . . 6
- 2.5 На верхней границе . . . 7
- 2.6 Левый-нижний угол . . . 7
- 2.7 Левый-верхний угол . . . 9
- 2.8 Правый-верхний угол . . . 9
- 2.9 Правый-нижний угол . . . 9

3 Невязка разностной схемы **11**
- 3.1 Невязка в цилиндрической системе координат . . . 11
- 3.2 Невязка во внутренних точках . . . 11
- 3.3 Невязка на левой границе . . . 14
- 3.4 Невязка на правой границе . . . 16

4 Запись СЛАУ **19**
- 4.1 Запись для внутренних точек . . . 19
- 4.2 Запись для левой границы . . . 20
- 4.3 Запись для правой границы . . . 21
- 4.4 Запись для нижней границы . . . 21
- 4.5 Запись для верхней границы . . . 22
- 4.6 Запись для левой нижней граничной точки . . . 23
- 4.7 Запись для правой нижней граничной точки . . . 23

5 Метод сопряжённых градиентов **27**
- 5.1 Явный метод . . . 27
- 5.2 Неявный метод . . . 27

6 Тестирование **29**

7 Заключение **33**

Код программы **34**

1 Вступление

1.1 Постановка задачи

Вариант N7. Используя интегро-интерполяционный метод, разработать подпрограмму для моделирования распределения температуры в цилиндре, описываемого математической моделью

$$-\left[\frac{1}{r}\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right)+\frac{\partial}{\partial z}\left(k_2(r,z)\frac{\partial u}{\partial z}\right)\right]=f(r,z) \tag{1}$$

$$0\le c_{11}\le k_1(r,z)\le c_{12},\quad 0\le c_{11}\le k_2(r,z)\le c_{22},\quad 0\le r\le R,\ 0\le z\le L$$

С граничными условиями:

$u|_{r=0}$ – ограничено $\qquad -k_1\left.\frac{\partial u}{\partial r}\right|_{r=R}=\chi_2\, u|_{r=R}-\varphi_2(z)$

$k_2\left.\frac{\partial u}{\partial z}\right|_{z=0}=\chi_3\, u|_{z=0}-\varphi_3(r) \qquad u|_{z=L}=\varphi_4(r)$

$\chi_2\ge 0 \qquad \chi_3\ge 0$

Матрица алгебраической системы должна храниться в упакованной форме - сжатый разреженный строчный вид. Пример:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A	13	7	1	14	8	2	15	3	16	9	4	17	10	5	18	6	19	11	20	12	21
IC	1	2	4	2	3	5	3	6	4	5	7	5	6	8	6	9	7	8	8	9	9

	1	2	3	4	5	6	7	8	9	10
IR	1	4	7	9	12	15	17	19	21	22

2 Разностная схема

Введем основную сетку:

$$\begin{array}{ll} N_r - \text{число разбиений на}[0, R] & N_z - \text{число разбиений на}[0, L] \\ r_0 < r_1 < \cdots < r_N & z_0 < z_1 < \cdots < z_N \\ r_0 = 0, \quad r_N = R & z_0 = 0, \quad z_N = L \\ h_r = \dfrac{R-0}{N_r} & h_z = \dfrac{L-0}{N_z} \end{array}$$

Введем дополнительную сетку:

$$r_{i-\frac{1}{2}} = \frac{r_i + r_{i-1}}{2} \quad i = 1, \cdots, N_r \qquad\qquad z_{j-\frac{1}{2}} = \frac{z_j + z_{j-1}}{2} \quad j = 1, \cdots, N_z$$

$$\hbar_i = \begin{cases} \frac{h_r}{2}, \ i = 0 \\ \\ h_r, \ i = 1, 2, \dots, N_r - 1 \\ \\ \frac{h_r}{2}, \ i = N_r \end{cases} \qquad\qquad \hbar_j = \begin{cases} \frac{h_z}{2}, \ j = 0 \\ \\ h_z, \ j = 1, 2, \dots, N_z - 1 \\ \\ \frac{h_z}{2}, \ j = N_z \end{cases}$$

Преобразуем наше начальное уравнение домножив на r

$$-\left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right] = rf(r,z)$$

2.1 Внутренние точки

Проинтегрируем уравнение (1) в области $[r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}] \times [z_{j-\frac{1}{2}}, z_{j+\frac{1}{2}}]$ для $i = 1, 2, \dots, N_r - 1$ и $j = 1, 2, \dots, N_z - 1$.

$$-\int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} \left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right] drdz = \int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rf(r,z)drdz$$

Получим:

$$-\left[\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_{i+\frac{1}{2}}} dz - \int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_{i-\frac{1}{2}}} dz \right.$$
$$\left. + \int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_{j+\frac{1}{2}}} dr - \int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_{j-\frac{1}{2}}} dr\right] = \int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rf(r,z)drdz$$

Воспользуемся формулами численного дифференцирования:

$$k_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_{i-\frac{1}{2}}} \approx k_1(r_{i-\frac{1}{2}}, z)\frac{v_{i,j}-v_{i-1,j}}{h_r}$$

$$k_2(r,z)\frac{\partial u}{\partial r}\bigg|_{z=z_{j-\frac{1}{2}}} \approx k_2(r, z_{j-\frac{1}{2}})\frac{v_{i,j}-v_{i,j-1}}{h_z}$$

Также воспользуемся формулой средних прямоугольников:

$$\int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} r\varphi(r,z)dr = \hbar_i r_i \varphi_i$$

$$\int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}}\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} r\varphi(r,z)drdz = \hbar_i\hbar_j r_i \varphi_{i,j}$$

В итоге получим уравнение разностной схемы:

$$-\Bigg[\hbar_j r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j)\frac{v_{i+1,j}-v_{i,j}}{h_r} - \hbar_j r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j)\frac{v_{i,j}-v_{i-1,j}}{h_r}$$
$$+\hbar_i r_i k_2(r_i, z_{j+\frac{1}{2}})\frac{v_{i,j+1}-v_{i,j}}{h_z} - \hbar_i r_i k_2(r_i, z_{j-\frac{1}{2}})\frac{v_{i,j}-v_{i,j-1}}{h_z}\Bigg] = \hbar_i\hbar_j r_i f_{i,j}$$

Так как выбранная оснавная сетка является равномерной, то $\hbar_i = h_r$ и $\hbar_j = h_z$, для $i = 1, 2, \ldots, N_r - 1$ и $j = 1, 2, \ldots, N_z - 1$.

$$-\Bigg[h_z r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j)\frac{v_{i+1,j}-v_{i,j}}{h_r} - h_z r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j)\frac{v_{i,j}-v_{i-1,j}}{h_r}$$
$$+h_r r_i k_2(r_i, z_{j+\frac{1}{2}})\frac{v_{i,j+1}-v_{i,j}}{h_z} - h_r r_i k_2(r_i, z_{j-\frac{1}{2}})\frac{v_{i,j}-v_{i,j-1}}{h_z}\Bigg] = h_r h_z r_i f_{i,j}$$

2.2 На левой границе

Проинтегрируем уравнение (1) в области $[r_i, r_{i+\frac{1}{2}}] \times [z_{j-\frac{1}{2}}, z_{j+\frac{1}{2}}]$ для $i = 0$ и $j = 1, 2, \ldots, N_z - 1$.

$$-\int\limits_{r_i}^{r_{i+\frac{1}{2}}}\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} \left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right] drdz = \int\limits_{r_i}^{r_{i+\frac{1}{2}}}\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rf(r,z)drdz$$

Получаем:

$$-\left[\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\Big|_{r=r_{i+\frac{1}{2}}} dz - \int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\Big|_{r=r_i} dz \right.$$
$$\left. + \int\limits_{r_i}^{r_{i+\frac{1}{2}}} rk_2(r,z)\frac{\partial u}{\partial z}\Big|_{z=z_{j+\frac{1}{2}}} dr - \int\limits_{r_i}^{r_{i+\frac{1}{2}}} rk_2(r,z)\frac{\partial u}{\partial z}\Big|_{z=z_{j-\frac{1}{2}}} dr\right] = \int\limits_{r_i}^{r_{i+\frac{1}{2}}}\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rf(r,z)drdz$$

Имеем граничное условие:

$$u\big|_{r=0} - \text{ограничено, т. е } \frac{\partial u}{\partial r}\Big|_{r=0} = 0$$

$$\int\limits_{r_i}^{r_{i+\frac{1}{2}}} rfdr \approx f_i \int\limits_{r_i}^{r_{i+\frac{1}{2}}} rdr = f_i\frac{r^2_{i+\frac{1}{2}}}{2} = \frac{h_r}{2}f_i\frac{r_{i+\frac{1}{2}}}{2}, \quad i=0, \quad r_i = 0, r_{i+\frac{1}{2}} = \frac{h_r}{2}$$

Получаем уравнение разностной схемы:

$$-\left[\hbar_j r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j)\frac{v_{i+1,j}-v_{i,j}}{h_{i+1}} - 0\right.$$
$$\left.+\hbar_i\frac{r_{i+\frac{1}{2}}}{2}k_2(r_i, z_{j+\frac{1}{2}})\frac{v_{i,j+1}-v_{i,j}}{h_{j+1}} - \hbar_i\frac{r_{i+\frac{1}{2}}}{2}k_2(r_i, z_{j-\frac{1}{2}})\frac{v_{i,j}-v_{i,j-1}}{h_j}\right] = \hbar_i\hbar_j\frac{r_{i+\frac{1}{2}}}{2}f_{i,j}$$

Так как выбранная оснавная сетка является равномерной, то $\hbar_i = \frac{h_r}{2}$ и $\hbar_j = h_z$, для $i = 0$ и $j = 1, 2, ..., N_z - 1$.

$$-\left[h_z r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_j)\frac{v_{1,j}-v_{0,j}}{h_r} - 0\right.$$
$$\left.+\frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}k_2(r_0, z_{j+\frac{1}{2}})\frac{v_{0,j+1}-v_{0,j}}{h_z} - \frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}k_2(r_0, z_{j-\frac{1}{2}})\frac{v_{0,j}-v_{0,j-1}}{h_z}\right] = \frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}h_z f_{0,j}$$

2.3 На правой границе

Проинтегрируем уравнение (1) в области $[r_{i-\frac{1}{2}}, r_i] \times [z_{j-\frac{1}{2}}, z_{j+\frac{1}{2}}]$ для $i = N_r$ и $j = 1, 2, ..., N_z - 1$.

$$-\int\limits_{r_{i-\frac{1}{2}}}^{r_i}\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} \left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right] drdz = \int\limits_{r_{i-\frac{1}{2}}}^{r_i}\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rf(r,z)drdz$$

Получаем:

$$
-\left[\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\Big|_{r=r_i} dz - \int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\Big|_{r=r_{i-\frac{1}{2}}} dz \right.
$$

$$
\left. + \int\limits_{r_{i-\frac{1}{2}}}^{r_i} rk_2(r,z)\frac{\partial u}{\partial z}\Big|_{z=z_{j+\frac{1}{2}}} dr - \int\limits_{r_{i-\frac{1}{2}}}^{r_i} rk_2(r,z)\frac{\partial u}{\partial z}\Big|_{z=z_{j-\frac{1}{2}}} dr\right] = \int\limits_{r_{i-\frac{1}{2}}}^{r_i}\int\limits_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rf(r,z)drdz
$$

Имеем граничное условие:

$$
-k_1\frac{\partial u}{\partial r}\Big|_{r=R} = \chi_2\ u|_{r=R} - \varphi_2(z)
$$

Получаем уравнение разностной схемы:

$$
-\left[-\hbar_j r_i(\chi_2 v_i - \varphi_2(z)) - \hbar_j r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j)\frac{v_{i,j}-v_{i-1,j}}{h_r}\right.
$$

$$
\left.+\hbar_i r_i k_2(r_i, z_{j+\frac{1}{2}})\frac{v_{i,j+1}-v_{i,j}}{h_z} - \hbar_i r_i k_2(r_i, z_{j-\frac{1}{2}})\frac{v_{i,j}-v_{i,j-1}}{h_z}\right] = \hbar_i\hbar_j r_i f_{i,j}
$$

Так как выбранная оснавная сетка является равномерной, то $\hbar_i = \frac{h_r}{2}$ и $\hbar_j = h_z$, для $i = N_r$ и $j = 1, 2, \dots, N_z - 1$.

$$
-\left[-h_z r_{N_r}(\chi_2 v_{N_r} - \varphi_2(z)) - h_z r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_j)\frac{v_{N_r,j}-v_{N_r-1,j}}{h_r}\right.
$$

$$
\left.+\frac{h_r}{2} r_{N_r} k_2(r_{N_r}, z_{j+\frac{1}{2}})\frac{v_{N_r,j+1}-v_{N_r,j}}{h_z} - \frac{h_r}{2} r_{N_r} k_2(r_{N_r}, z_{j-\frac{1}{2}})\frac{v_{N_r,j}-v_{N_r,j-1}}{h_z}\right] = \frac{h_r}{2} r_{N_r} h_z f_{N_r,j}
$$

2.4 На нижней границе

Проинтегрируем уравнение (1) в области $[r_{i-\frac{1}{2}}, r_{i+\frac{1}{2}}]\times[z_j, z_{j+\frac{1}{2}}]$ для $i = 1, 2, \dots, N_r - 1$ и $j = 0$.

$$
-\int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}}\int\limits_{z_j}^{z_{j+\frac{1}{2}}}\left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right] drdz = \int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}}\int\limits_{z_j}^{z_{j+\frac{1}{2}}} rf(r,z)drdz
$$

Получаем:

$$-\left[\int\limits_{z_j}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_{i+\frac{1}{2}}} dz - \int\limits_{z_j}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_{i-\frac{1}{2}}} dz\right.$$
$$\left.+\int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_{j+\frac{1}{2}}} dr - \int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_j} dr\right] = \int\limits_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}}\int\limits_{z_j}^{z_{j+\frac{1}{2}}} rf(r,z)drdz$$

Имеем граничное условие:

$$k_2\frac{\partial u}{\partial z}\bigg|_{z=0} = \chi_3\ u|_{z=0} - \varphi_3(r)$$

Получаем уравнение разностной схемы:

$$-\left[\hbar_j r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j)\frac{v_{i+1,j}-v_{i,j}}{h_r} - \hbar_j r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j)\frac{v_{i,j}-v_{i-1,j}}{h_r}\right.$$
$$\left.+\hbar_i r_i k_2(r_i, z_{j+\frac{1}{2}})\frac{v_{i,j+1}-v_{i,j}}{h_z} - \hbar_i r_i(\chi_3 v_i - \varphi_3(r))\right] = \hbar_i\hbar_j r_i f_{i,j}$$

Так как выбранная оснавная сетка является равномерной, то $\hbar_i = h_r$ и $\hbar_j = \frac{h_z}{2}$, для $i = 1, 2, \dots, N_r - 1$ и $j = 0$.

$$-\left[\frac{h_z}{2} r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_0)\frac{v_{i+1,0}-v_{i,0}}{h_r} - \frac{h_z}{2} r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_0)\frac{v_{i,0}-v_{i-1,0}}{h_r}\right.$$
$$\left.+h_r r_i k_2(r_i, z_{\frac{1}{2}})\frac{v_{i,1}-v_{i,0}}{h_z} - h_r r_i(\chi_3 v_{i,0} - \varphi_3(r))\right] = h_r\frac{h_z}{2} r_i f_{i,0}$$

2.5 На верхней границе

В качестве уравнения разностной схемы на верхней границе сетки, а то есть для $i = 1, 2, \dots, N_r - 1$ и $j = N_z$, возьмём известное граничное условие:

$$u|_{z=L} = \varphi_r(r)$$

$$v_{i,N_z} = \varphi(r_i)$$

2.6 Левый-нижний угол

Проинтегрируем уравнение (1) в области $[r_i, r_{i+\frac{1}{2}}] \times [z_j, z_{j+\frac{1}{2}}]$ для $i = 0$ и $j = 0$.

$$-\int\limits_{r_i}^{r_{i+\frac{1}{2}}}\int\limits_{z_j}^{z_{j+\frac{1}{2}}}\left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right)+\frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right]drdz=\int\limits_{r_i}^{r_{i+\frac{1}{2}}}\int\limits_{z_j}^{z_{j+\frac{1}{2}}}rf(r,z)drdz$$

Получаем:

$$-\left[\int\limits_{z_j}^{z_{j+\frac{1}{2}}}rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_{i+\frac{1}{2}}}dz-\int\limits_{z_j}^{z_{j+\frac{1}{2}}}rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_i}dz\right.$$
$$\left.+\int\limits_{r_i}^{r_{i+\frac{1}{2}}}rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_{j+\frac{1}{2}}}dr-\int\limits_{r_i}^{r_{i+\frac{1}{2}}}rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_j}dr\right]=\int\limits_{r_i}^{r_{i+\frac{1}{2}}}\int\limits_{z_j}^{z_{j+\frac{1}{2}}}rf(r,z)drdz$$

Имеем граничное условие:

$$u|_{r=0}\text{ – ограничено, т. е }\frac{\partial u}{\partial r}\bigg|_{r=0}=0$$

$$\int\limits_{r_i}^{r_{i+\frac{1}{2}}}rfdr\approx f_i\int\limits_{r_i}^{r_{i+\frac{1}{2}}}rdr=f_i\frac{r^2_{i+\frac{1}{2}}}{2}=h_rf_i\frac{r_{i+\frac{1}{2}}}{2},\quad i=0,\quad r_i=0,r_{i+\frac{1}{2}}=\frac{h_r}{2}$$

Также:

$$k_2\frac{\partial u}{\partial z}\bigg|_{z=0}=\chi_3\ u|_{z=0}-\varphi_3(r)$$

Получаем уравнение разностной схемы:

$$-\left[\hbar_jr_{i+\frac{1}{2}}k_1(r_{i+\frac{1}{2}},z_j)\frac{v_{i+1,j}-v_{i,j}}{h_r}-0\right.$$
$$\left.+\hbar_i\frac{r_{i+\frac{1}{2}}}{2}k_2(r_i,z_{j+\frac{1}{2}})\frac{v_{i,j+1}-v_{i,j}}{h_z}-\hbar_i\frac{r_{i+\frac{1}{2}}}{2}(\chi_3\ u|_{z=0}-\varphi_3(r_0))\right]=\hbar_i\hbar_j\frac{r_{i+\frac{1}{2}}}{2}f_{i,j}$$

Так как выбранная оснавная сетка является равномерной, то $\hbar_i=\frac{h_r}{2}$ и $\hbar_j=\frac{h_z}{2}$, для $i=0$ и $j=0$.

$$-\left[\frac{h_z}{2}r_{\frac{1}{2}}k_1(r_{\frac{1}{2}},z_0)\frac{v_{1,0}-v_{0,0}}{h_r}-0\right.$$
$$\left.+\frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}k_2(r_0,z_{\frac{1}{2}})\frac{v_{0,1}-v_{0,0}}{h_z}-\frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}(\chi_3v_{0,0}-\varphi_3(r_0))\right]=\frac{h_r}{2}\frac{h_z}{2}\frac{r_{\frac{1}{2}}}{2}f_{0,0}$$

2.7 Левый-верхний угол

В качестве уравнения разностной схемы в левой-верхней граничной точке сетки, а то есть для $i = 0$ и $j = N_z$, возьмём известное граничное условие:

$$u|_{z=L} = \varphi_r(r)$$

$$v_{0,N_z} = \varphi(r_0)$$

2.8 Правый-верхний угол

В качестве уравнения разностной схемы в правой-верхней граничной точке сетки, а то етсь для $i = N_r$ и $j = N_z$, возьмём известное граничное условие:

$$u|_{z=L} = \varphi_r(r)$$

$$v_{N_r,N_z} = \varphi(r_{N_r})$$

2.9 Правый-нижний угол

Проинтегрируем уравнение (1) в области $[r_{i-\frac{1}{2}}, r_i] \times [z_j, z_{j+\frac{1}{2}}]$ для $i = N_r$ и $j = 0$.

$$-\int\limits_{r_{i-\frac{1}{2}}}^{r_i} \int\limits_{z_j}^{z_{j+\frac{1}{2}}} \left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right] = \int\limits_{r_{i-\frac{1}{2}}}^{r_i} \int\limits_{z_j}^{z_{j+\frac{1}{2}}} rf(r,z)$$

Получаем:

$$-\left[\int\limits_{z_j}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_i} dz - \int\limits_{z_j}^{z_{j+\frac{1}{2}}} rk_1(r,z)\frac{\partial u}{\partial r}\bigg|_{r=r_{i-\frac{1}{2}}} dz \right.$$
$$\left. + \int\limits_{r_{i-\frac{1}{2}}}^{r_i} rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_{j+\frac{1}{2}}} dr - \int\limits_{r_{i-\frac{1}{2}}}^{r_i} rk_2(r,z)\frac{\partial u}{\partial z}\bigg|_{z=z_j} dr\right] = \int\limits_{r_{i-\frac{1}{2}}}^{r_i} \int\limits_{z_j}^{z_{j+\frac{1}{2}}} rf(r,z)drdz$$

Имеем граничные условия:

$$-k_1\frac{\partial u}{\partial r}\bigg|_{r=R} = \chi_2\, u|_{r=R} - \varphi_2(z)$$
$$k_2\frac{\partial u}{\partial z}\bigg|_{z=0} = \chi_3\, u|_{z=0} - \varphi_3(r)$$

Получаем уравнение разностной схемы:

$$-\left[-\hbar_j r_i(\chi_2 \left.u\right|_{r=R} - \varphi_2(z)) - \hbar_j r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j)\frac{v_{i,j} - v_{i-1,j}}{h_r}\right.$$
$$\left.+\hbar_i r_i k_2(r_i, z_{j+\frac{1}{2}})\frac{v_{i,j+1} - v_{i,j}}{h_z} - \hbar_i r_i(\chi_3 v_i - \varphi_3(r))\right] = \hbar_i \hbar_j r_i f_{i,j}$$

Так как выбранная оснавная сетка является равномерной, то $\hbar_i = \frac{h_r}{2}$ и $\hbar_j = \frac{h_z}{2}$, для $i = N_r$ и $j = 0$.

$$-\left[-\frac{h_z}{2} r_{N_r}(\chi_2 v_{N_r,0} - \varphi_2(z)) - \frac{h_z}{2} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_0)\frac{v_{N_r,0} - v_{N_r-1,0}}{h_r}\right.$$
$$\left.+\frac{h_r}{2} r_{N_r} k_2(r_{N_r}, z_{\frac{1}{2}})\frac{v_{N_r,1} - v_{N_r,0}}{h_z} - \frac{h_r}{2} r_{N_r}(\chi_3 v_{N_r,0} - \varphi_3(r))\right] = \frac{h_r}{2}\frac{h_z}{2} r_{N_r} f_{N_r,0}$$

3 Невязка разностной схемы

3.1 Невязка в цилиндрической системе координат

Возьмём наше уже преобразованное уравнение

$$-\left[\frac{\partial}{\partial r}\left(rk_1(r,z)\frac{\partial u}{\partial r}\right)+\frac{\partial}{\partial z}\left(rk_2(r,z)\frac{\partial u}{\partial z}\right)\right]=rf(r,z)$$

Сделаем следующие замены:

$$\tilde{k}_1(r,z)=rk_1(r,z)$$

$$\tilde{k}_2(r,z)=rk_2(r,z)$$

$$\tilde{f}(r,z)=rf(r,z)$$

Получаем:

$$-\left[\frac{\partial}{\partial r}\left(\tilde{k}_1(r,z)\frac{\partial u}{\partial r}\right)+\frac{\partial}{\partial z}\left(\tilde{k}_2(r,z)\frac{\partial u}{\partial z}\right)\right]=\tilde{f}(r,z)$$

3.2 Невязка во внутренних точках

Запишем для уравнения разностной сетки во внутренних точках невязку:

$$\begin{aligned}-\Bigg[&h_z\tilde{k}_1(r_{i+\frac{1}{2}},z_j)\frac{v_{i+1,j}-v_{i,j}}{h_r}-h_z\tilde{k}_1(r_{i-\frac{1}{2}},z_j)\frac{v_{i,j}-v_{i-1,j}}{h_r}\\&+h_r\tilde{k}_2(r_i,z_{j+\frac{1}{2}})\frac{v_{i,j+1}-v_{i,j}}{h_z}-h_r\tilde{k}_2(r_i,z_{j-\frac{1}{2}})\frac{v_{i,j}-v_{i,j-1}}{h_z}\Bigg]=h_rh_z\tilde{f}_{i,j}\end{aligned}$$

$$\begin{aligned}\xi_{i,j}=h_rh_z\tilde{f}_{i,j}+\Bigg[&h_z\tilde{k}_1(r_{i+\frac{1}{2}},z_j)\frac{u_{i+1,j}-u_{i,j}}{h_r}-h_z\tilde{k}_1(r_{i-\frac{1}{2}},z_j)\frac{u_{i,j}-u_{i-1,j}}{h_r}\\&+h_r\tilde{k}_2(r_i,z_{j+\frac{1}{2}})\frac{u_{i,j+1}-u_{i,j}}{h_z}-h_r\tilde{k}_2(r_i,z_{j-\frac{1}{2}})\frac{u_{i,j}-u_{i,j-1}}{h_z}\Bigg]\end{aligned}$$

Напишем разложение Тейлора для невязки:

$$u_{i,j-1}=u(r_i,z_j-h_z)=\left[u-h_z\frac{\partial u}{\partial z}+\frac{h_z^2}{2}\frac{\partial^2 u}{\partial z^2}-\frac{h_z^3}{6}\frac{\partial^3 u}{\partial z^3}+\frac{h_z^4}{24}\frac{\partial^4 u}{\partial z^4}\right]_{i,j}+\mathcal{O}(h_y^5)$$

$$\frac{u_{i,j}-u_{i,j-1}}{h_z}=\left[\frac{\partial u}{\partial z}-\frac{h_z}{2}\frac{\partial^2 u}{\partial z^2}+\frac{h_z^2}{6}\frac{\partial^3 u}{\partial z^3}-\frac{h_z^3}{24}\frac{\partial^4 u}{\partial z^4}\right]_{i,j}+\mathcal{O}(h_y^4)$$

$$\tilde{k}_2(r_i, z_{j-\frac{1}{2}}) = r_i k_2(r_i, z_j - \frac{h_z}{2}) = \left[\tilde{k}_2 - \frac{h_z}{2}\frac{\partial \tilde{k}_2}{\partial z} + \frac{h_z^2}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2} - \frac{h_z^3}{48}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\right]_{i,j} + \mathcal{O}(h^4)$$

$$\begin{aligned}
\tilde{k}_2(r_i, z_{j-\frac{1}{2}})\frac{u_{i,j} - u_{i,j-1}}{h_z} &= \left[\tilde{k}_2 \frac{\partial u}{\partial z}\right]_{i,j} - h_z \left[\frac{1}{2}\tilde{k}_2 \frac{\partial^2 u}{\partial z^2} + \frac{1}{2}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial u}{\partial z}\right]_{i,j} + \\
&+ h_z^2 \left[\frac{1}{6}\tilde{k}_2 \frac{\partial^3 u}{\partial z^3} + \frac{1}{4}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^2 u}{\partial z^2} + \frac{1}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial u}{\partial z}\right]_{i,j} - \\
&- h_z^3 \left[\frac{1}{24}\tilde{k}_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{12}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{16}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{48}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right]_{i,j} + \\
&+ \mathcal{O}(h^4)
\end{aligned}$$

Для $\tilde{k}_2(r_i, z_{j+\frac{1}{2}})$ можно получить невязку аналогичным способом:

$$\begin{aligned}
\tilde{k}_2(r_i, z_{j+\frac{1}{2}})\frac{u_{i,j+1} - u_{i,j}}{h_z} &= \left[\tilde{k}_2 \frac{\partial u}{\partial z}\right]_{i,j} + h_z \left[\frac{1}{2}\tilde{k}_2 \frac{\partial^2 u}{\partial z^2} + \frac{1}{2}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial u}{\partial z}\right]_{i,j} + \\
&+ h_z^2 \left[\frac{1}{6}\tilde{k}_2 \frac{\partial^3 u}{\partial z^3} + \frac{1}{4}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^2 u}{\partial z^2} + \frac{1}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial u}{\partial z}\right]_{i,j} + \\
&+ h_z^3 \left[\frac{1}{24}\tilde{k}_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{12}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{16}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{48}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right]_{i,j} + \\
&+ \mathcal{O}(h^4)
\end{aligned}$$

$$u_{i-1,j} = u(r_i - h_r, j) = \left[u - h_r \frac{\partial u}{\partial r} + \frac{h_r^2}{2}\frac{\partial^2 u}{\partial r^2} - \frac{h_r^3}{6}\frac{\partial^3 u}{\partial r^3} + \frac{h_r^4}{24}\frac{\partial^4 u}{\partial r^4}\right]_{i,j} + \mathcal{O}(h^5)$$

$$\frac{u_{i,j} - u_{i-1,j}}{h_r} = \left[\frac{\partial u}{\partial r} - \frac{h_r}{2}\frac{\partial^2 u}{\partial r^2} + \frac{h_r^2}{6}\frac{\partial^3 u}{\partial r^3} - \frac{h_r^3}{24}\frac{\partial^4 u}{\partial r^4}\right]_{i,j} + \mathcal{O}(h^4)$$

$$\tilde{k}_1(r_{i-\frac{1}{2}}, z_j) = \tilde{k}_1(r_i - \frac{h_r}{2}, z_j) = \left[\tilde{k}_1 - \frac{h_r}{2}\frac{\partial \tilde{k}_1}{\partial r} + \frac{h_r^2}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2} - \frac{h_r^3}{48}\frac{\partial^3 \tilde{k}_1}{\partial r^3}\right]_{i,j} + \mathcal{O}(h^4)$$

$$\tilde{k}_1(r_{i-\frac{1}{2}}, z_j)\frac{u_{i,j}-u_{i-1,j}}{h_r} = \left[\tilde{k}_1\frac{\partial u}{\partial r}\right]_{i,j} - h_r\left[\frac{1}{2}\tilde{k}_1\frac{\partial^2 u}{\partial r^2} + \frac{1}{2}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial u}{\partial r}\right]_{i,j} +$$
$$+ h_r^2\left[\frac{1}{6}\tilde{k}_1\frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{i,j} -$$
$$- h_r^3\left[\frac{1}{24}\tilde{k}_1\frac{\partial^4 u}{\partial r^4} - \frac{1}{12}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{16}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{48}\frac{\partial^3 \tilde{k}_1}{\partial r^3}\frac{\partial u}{\partial r}\right]_{i,j} +$$
$$+ \mathcal{O}(h^4)$$

Для $\tilde{k}_1(r_{i=\frac{1}{2}}, z_j)\frac{u_{i+1,j}-u_{i,j}}{h_r}$ можно получить невязку аналогично:

$$\tilde{k}_1(r_{i+\frac{1}{2}}, z_j)\frac{u_{i+1,j}-u_{i,j}}{h_r} = \left[\tilde{k}_1\frac{\partial u}{\partial r}\right]_{i,j} + h_r\left[\frac{1}{2}\tilde{k}_1\frac{\partial^2 u}{\partial r^2} + \frac{1}{2}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial u}{\partial r}\right]_{i,j} +$$
$$+ h_r^2\left[\frac{1}{6}\tilde{k}_1\frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{i,j} +$$
$$+ h_r^3\left[\frac{1}{24}\tilde{k}_1\frac{\partial^4 u}{\partial r^4} - \frac{1}{12}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{16}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{48}\frac{\partial^3 \tilde{k}_1}{\partial r^3}\frac{\partial u}{\partial r}\right]_{i,j} +$$
$$+ \mathcal{O}(h^4)$$

Тогда:

$$h_z\tilde{k}_1(r_{i+\frac{1}{2}}, z_j)\frac{u_{i+1,j}-u_{i,j}}{h_r} - h_z\tilde{k}_1(r_{i-\frac{1}{2}}, z_j)\frac{u_{i,j}-u_{i-1,j}}{h_r} = h_r h_z\left(\left[\tilde{k}_1\frac{\partial^2 u}{\partial r^2} + \frac{\partial \tilde{k}_1}{\partial r}\frac{\partial u}{\partial r}\right]_{i,j} +\right.$$
$$\left. + h_r^2\left[\frac{1}{12}\tilde{k}_1\frac{\partial^4 u}{\partial r^4} - \frac{1}{6}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_1}{\partial r^3}\frac{\partial u}{\partial r}\right] + \mathcal{O}(h_r^4)\right)$$

$$h_x\tilde{k}_2(r_i, z_{j+\frac{1}{2}})\frac{u_{i,j+1}-u_{i,j}}{h_z} - h_x\tilde{k}_2(r_i, z_{j-\frac{1}{2}})\frac{u_{i,j}-u_{i,j-1}}{h_z} = h_r h_z\left(\left[\tilde{k}_2\frac{\partial^2 u}{\partial z^2} + \frac{\partial \tilde{k}_2}{\partial z}\frac{\partial u}{\partial z}\right]_{i,j} +\right.$$
$$\left. + h_z^2\left[\frac{1}{12}\tilde{k}_2\frac{\partial^4 u}{\partial z^4} - \frac{1}{6}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right]_{i,j} + \mathcal{O}(h_z^4)\right)$$

Будем искать невязку в следующем виде:

$$\tilde{\xi}_{i,j} = \frac{\xi_{i,j}}{h_r h_z}$$

Тогда:

$$\tilde{\xi}_{i,j} = \tilde{f}_{i,j} + \left[\tilde{k}_1 \frac{\partial^2 u}{\partial r^2} + \frac{\partial \tilde{k}_1}{\partial r}\frac{\partial u}{\partial r}\right]_{i,j} + \left[\tilde{k}_2 \frac{\partial^2 u}{\partial z^2} + \frac{\partial \tilde{k}_2}{\partial z}\frac{\partial u}{\partial z}\right]_{i,j} +$$
$$+ h_r^2 \left[\frac{1}{12}\tilde{k}_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{6}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_1}{\partial r^3}\frac{\partial u}{\partial r}\right] + \mathcal{O}(h_r^3) +$$
$$+ h_z^2 \left[\frac{1}{12}\tilde{k}_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{6}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right]_{i,j} + \mathcal{O}(h_z^3)$$

Можно заметить, что:

$$\left[\tilde{k}_1 \frac{\partial^2 u}{\partial r^2} + \frac{\partial \tilde{k}_1}{\partial r}\frac{\partial u}{\partial r}\right]_{i,j} = \frac{\partial}{\partial r}\left(\tilde{k}_1 \frac{\partial u}{\partial r}\right)$$

$$\left[\tilde{k}_2 \frac{\partial^2 u}{\partial z^2} + \frac{\partial \tilde{k}_2}{\partial z}\frac{\partial u}{\partial z}\right]_{i,j} = \frac{\partial}{\partial z}\left(\tilde{k}_2 \frac{\partial u}{\partial z}\right)$$

Тем самым:

$$r_i f_{i,j} + \frac{\partial}{\partial r}\left(k_1 \frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(k_2 \frac{\partial u}{\partial z}\right) = 0$$

Получаем:

$$\tilde{\xi}_{i,j} = h_r^2 \left[\frac{1}{12}\tilde{k}_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{6}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_1}{\partial r^3}\frac{\partial u}{\partial r}\right]_{i,j} + \mathcal{O}(h_r^3) +$$
$$+ h_z^2 \left[\frac{1}{12}\tilde{k}_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{6}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right]_{i,j} + \mathcal{O}(h_z^3)$$

Уберём замену и вернёмся к старым обозначеням:

$$\tilde{\xi}_{i,j} = h_r^2 \left[\frac{1}{12} r k_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{6}\frac{\partial r k_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{8}\frac{\partial^2 r k_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{24}\frac{\partial^3 r k_1}{\partial r^3}\frac{\partial u}{\partial r}\right]_{i,j} + \mathcal{O}(h_r^3) +$$
$$+ h_z^2 \left[\frac{1}{12} r k_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{6}\frac{\partial r k_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 r k_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 r k_2}{\partial z^3}\frac{\partial u}{\partial z}\right]_{i,j} + \mathcal{O}(h_z^3)$$

Порядок аппроксимации $p_r = 2 - 0 = 2$, $p_z = 2 - 0 = 2$.

3.3 Невязка на левой границе

Запишем для уравнения разностной сетки на левой границе невязку:

$$-\left[h_z r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_j)\frac{v_{1,j}-v_{0,j}}{h_r} - 0\right.$$
$$\left.+\frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}k_2(r_0, z_{j+\frac{1}{2}})\frac{v_{0,j+1}-v_{0,j}}{h_z} - \frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}k_2(r_0, z_{j-\frac{1}{2}})\frac{v_{0,j}-v_{0,j-1}}{h_z}\right] = \frac{h_r}{2}\frac{r_{\frac{1}{2}}}{2}h_z f_{0,j}$$

$$\xi_{0,j} = \frac{h_r}{2}h_z f_{0,j} + \left[2h_z k_1(r_{\frac{1}{2}}, z_j)\frac{u_{1,j}-u_{0,j}}{h_r} - 0\right.$$
$$\left.+\frac{h_r}{2}k_2(r_0, z_{j+\frac{1}{2}})\frac{u_{0,j+1}-u_{0,j}}{h_z} - \frac{h_r}{2}k_2(r_0, z_{j-\frac{1}{2}})\frac{u_{0,j}-u_{0,j-1}}{h_z}\right]$$

Напишем разложение Тейлора для невязки:

$$k_1(r_{\frac{1}{2}}, z_j)\frac{u_{1,j}-u_{0,j}}{h_r} = \left[k_1\frac{\partial u}{\partial r}\right]_{i,j} + \frac{h_r}{2}\left[\frac{\partial}{\partial r}\left(k_1\frac{\partial u}{\partial r}\right)\right]_{i,j} +$$
$$+ h_r^2\left[\frac{1}{6}k_1\frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial k_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 k_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{i,j} + \mathcal{O}(h_r^3)$$

$$h_r k_2(r_0, z_{j+\frac{1}{2}})\frac{u_{0,j+1}-u_{0,j}}{h_z} - h_r k_2(r_0, z_{j-\frac{1}{2}})\frac{u_{0,j}-u_{0,j-1}}{h_z} = h_r\left[h_z\left(\frac{\partial}{\partial z}\left(k_2\frac{\partial u}{\partial z}\right)\right)_{i,j} +\right.$$
$$\left.+h_z^3\left(\frac{1}{12}k_2\frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial k_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 k_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 k_2}{\partial z^3}\frac{\partial u}{\partial z}\right)_{0,j} + \mathcal{O}(h_z^4)\right]$$

$$\xi_{0,j} = \frac{h_r}{2}h_z f_{0,j} + \left[2h_z\left(\left[k_1\frac{\partial u}{\partial r}\right]_{i,j} + \frac{h_r}{2}\left[\frac{\partial}{\partial r}\left(k_1\frac{\partial u}{\partial r}\right)\right]_{i,j} +\right.\right.$$
$$\left.+h_r^2\left[\frac{1}{6}k_1\frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial k_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 k_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{i,j} + \mathcal{O}(h_r^3)\right) - 0+$$
$$\left.\frac{h_r}{2}\left(h_z\left[\frac{\partial}{\partial z}\left(k_2\frac{\partial u}{\partial z}\right)\right]_{i,j} + h_z^3\left[\frac{1}{12}k_2\frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial k_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 k_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 k_2}{\partial z^3}\frac{\partial u}{\partial z}\right]_{0,j} + \mathcal{O}(h_z^4)\right)\right]$$

Будем искать невязку в следующем виде:

$$\tilde{\xi}_{0,j} = \frac{\xi_{0,j}}{2h_z}$$

$$\tilde{\xi}_{0,j} = \frac{h_r}{4}\left[f + 2\frac{\partial}{\partial r}\left(k_1\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(k_2\frac{\partial u}{\partial z}\right)\right]_{0,j} + \left[k_1\frac{\partial u}{\partial r}\right]_{0,j} +$$
$$+ h_r^2\left[\frac{1}{6}k_1\frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial k_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 k_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{i,j} + \mathcal{O}(h_r^3)+$$
$$+ \frac{h_r}{4}\left[h_z^2\left(\frac{1}{12}k_2\frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial k_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 k_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 k_2}{\partial z^3}\frac{\partial u}{\partial z}\right)_{0,j} + \mathcal{O}(h_z^3)\right]$$

Можно заметить, что:

$$\left.\frac{\partial u}{\partial r}\right|_{r=0} = 0$$

$$\left[f + 2\frac{\partial}{\partial r}\left(k_1\frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(k_2\frac{\partial u}{\partial z}\right)\right]_{0,j} = 0$$

Получаем:

$$\tilde{\xi}_{0,j} = h_r^2\left[\frac{1}{6}k_1\frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial k_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 k_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{i,j} + \mathcal{O}(h_r^3) +$$

$$+\frac{h_r}{4}\left[h_z^2\left(\frac{1}{12}k_2\frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial k_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 k_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 k_2}{\partial z^3}\frac{\partial u}{\partial z}\right)_{0,j} + \mathcal{O}(h_z^3)\right]$$

Порядок аппроксимации $p_r = 2 - 0 = 2, p_z = 2 - 0 = 2$.

3.4 Невязка на правой границе

Запишем для уравнения разностной сетки на правой границе невязку:

$$-\left[-h_z r_{N_r}(\chi_2 v_{N_r} - \varphi_2(z)) - h_z\tilde{k}_1(r_{N_r-\frac{1}{2}}, z_j)\frac{v_{N_r,j} - v_{N_r-1,j}}{h_r}\right.$$

$$\left.+\frac{h_r}{2}\tilde{k}_2(r_{N_r}, z_{j+\frac{1}{2}})\frac{v_{N_r,j+1} - v_{N_r,j}}{h_z} - \frac{h_r}{2}\tilde{k}_2(r_{N_r}, z_{j-\frac{1}{2}})\frac{v_{N_r,j} - v_{N_r,j-1}}{h_z}\right] = \frac{h_r}{2}h_z\tilde{f}_{N_r,j}$$

$$\xi_{N_r,j} = \frac{h_r}{2}h_z\tilde{f}_{N_r,j} + \left[-h_z r_{N_r}(\chi_2 u_{N_r} - \varphi_2(z)) - h_z\tilde{k}_1(r_{N_r-\frac{1}{2}}, z_j)\frac{u_{N_r,j} - u_{N_r-1,j}}{h_r}\right.$$

$$\left.+\frac{h_r}{2}\tilde{k}_2(r_{N_r}, z_{j+\frac{1}{2}})\frac{u_{N_r,j+1} - u_{N_r,j}}{h_z} - \frac{h_r}{2}\tilde{k}_2(r_{N_r}, z_{j-\frac{1}{2}})\frac{u_{N_r,j} - u_{N_r,j-1}}{h_z}\right]$$

Напишем разложение Тейлора для невязки:

$$\tilde{k}_1(r_{N_r-\frac{1}{2}}, z_j)\frac{u_{N_r,j} - u_{N_r-1,j}}{h_r} = \left[\tilde{k}_1\frac{\partial u}{\partial r}\right]_{N_r,j} - \frac{h_2}{2}\left[\frac{\partial}{\partial r}\left(\tilde{k}_1\frac{\partial u}{\partial r}\right)\right]_{N_r,j} +$$

$$+ h_r^2\left[\frac{1}{6}\tilde{k}_1\frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial\tilde{k}_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2\tilde{k}_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{N_r,j} -$$

$$- h_r^3\left[\frac{1}{24}\tilde{k}_1\frac{\partial^4 u}{\partial r^4} + \frac{1}{12}\frac{\partial\tilde{k}_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{16}\frac{\partial^2\tilde{k}_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{48}\frac{\partial^3\tilde{k}_1}{\partial r^3}\frac{\partial u}{\partial r}\right]_{N_r,j} + \mathcal{O}(h_r^4)$$

$$\frac{h_r}{2}\tilde{k}_2(r_{N_r}, z_{j+\frac{1}{2}})\frac{u_{N_r,j+1} - u_{N_r,j}}{h_z} - \frac{h_r}{2}\tilde{k}_2(r_{N_r}, z_{j-\frac{1}{2}})\frac{u_{N_r,j} - u_{N_r,j-1}}{h_z} =$$

$$\frac{h_r}{2}\left[h_z\left(\frac{\partial}{\partial z}\left(\tilde{k}_2\frac{\partial u}{\partial z}\right)\right)_{N_r,j} + h_z^3\left(\frac{1}{12}\tilde{k}_2\frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial\tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2\tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3\tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right)_{N_r,j}\right] +$$

$$+\mathcal{O}(h_z^4)$$

$$\xi_{N_r,j} = \frac{h_r}{2} h_z \tilde{f}_{N_r,j} - h_z r_{N_r}(\chi_2 u_{N_r} - \varphi_2(z)) - h_z \left[\left[\tilde{k}_1 \frac{\partial u}{\partial r}\right]_{N_r,j} - \frac{h_2}{2} \left[\frac{\partial}{\partial r}\left(\tilde{k}_1 \frac{\partial u}{\partial r}\right)\right]_{N_r,j} + \right.$$

$$+ h_r^2 \left[\frac{1}{6}\tilde{k}_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{N_r,j} -$$

$$\left. - h_r^3 \left[\frac{1}{24}\tilde{k}_1 \frac{\partial^4 u}{\partial r^4} + \frac{1}{12}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^3 u}{\partial r^3} + \frac{1}{16}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial^2 u}{\partial r^2} + \frac{1}{48}\frac{\partial^3 \tilde{k}_1}{\partial r^3}\frac{\partial u}{\partial r}\right]_{N_r,j} + \mathcal{O}(h_r^4)\right] +$$

$$+ \frac{h_r}{2}\left[h_z \left(\frac{\partial}{\partial z}\left(\tilde{k}_2 \frac{\partial u}{\partial z}\right)\right)_{N_r,j} + h_z^3 \left(\frac{1}{12}\tilde{k}_2 \frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right)_{N_r,j} + \right.$$

$$\left. + \mathcal{O}(h_z^4)\right]$$

Будем искать невязку в следующем виде:

$$\tilde{\xi}_{N_r,j} = \frac{\xi_{N_r,j}}{h_z}$$

$$\tilde{\xi}_{N_r,j} = \frac{h_r}{2}\left[\tilde{f} + \frac{\partial}{\partial r}\left(\tilde{k}_1 \frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(\tilde{k}_2 \frac{\partial u}{\partial z}\right)\right]_{N_r,j} - \left[\tilde{k}_1 \frac{\partial u}{\partial r} + r(\chi_2 u - \varphi(z))\right]_{N_r,j} +$$

$$+ h_r^2 \left[\frac{1}{6}\tilde{k}_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial \tilde{k}_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 \tilde{k}_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{N_r,j} + \mathcal{O}(h_r^3) +$$

$$+ \frac{h_r}{2}\left[h_z^2 \left(\frac{1}{12}\tilde{k}_2 \frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial \tilde{k}_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 \tilde{k}_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 \tilde{k}_2}{\partial z^3}\frac{\partial u}{\partial z}\right)_{N_r,j} + \mathcal{O}(h_z^3)\right]$$

Уберём замену и вернёмся к старым обозначениям:

$$\tilde{\xi}_{N_r,j} = \frac{h_r}{2}\left[rf + \frac{\partial}{\partial r}\left(rk_1 \frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(rk_2 \frac{\partial u}{\partial z}\right)\right]_{N_r,j} - \left[rk_1 \frac{\partial u}{\partial r} + r(\chi_2 u - \varphi(z))\right]_{N_r,j} +$$

$$+ h_r^2 \left[\frac{1}{6} rk_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4}\frac{\partial rk_1}{\partial r}\frac{\partial^2 u}{\partial r^2} + \frac{1}{8}\frac{\partial^2 rk_1}{\partial r^2}\frac{\partial u}{\partial r}\right]_{N_r,j} + \mathcal{O}(h_r^3) +$$

$$+ \frac{h_r}{2}\left[h_z^2 \left(\frac{1}{12} rk_2 \frac{\partial^4 u}{\partial z^4} + \frac{1}{6}\frac{\partial rk_2}{\partial z}\frac{\partial^3 u}{\partial z^3} + \frac{1}{8}\frac{\partial^2 rk_2}{\partial z^2}\frac{\partial^2 u}{\partial z^2} + \frac{1}{24}\frac{\partial^3 rk_2}{\partial z^3}\frac{\partial u}{\partial z}\right)_{N_r,j} + \mathcal{O}(h_z^3)\right]$$

Заметим, что:

$$\left[rk1 \frac{\partial u}{\partial r} + r(\chi_2 u - \varphi(z))\right]_{N_r,j} = 0$$

$$\left[rf + \frac{\partial}{\partial r}\left(\tilde{k}_1 \frac{\partial u}{\partial r}\right) + \frac{\partial}{\partial z}\left(\tilde{k}_2 \frac{\partial u}{\partial z}\right)\right]_{N_r,j} = 0$$

Порядок аппроксимации $p_r = 2 - 0 = 2, p_z = 2 - 0 = 2$.

Можно сделать вывод, что полученная разностная схема имеет второй порядок аппроксимации.

4 Запись СЛАУ

Перейдём к одноиндексной записи

$$m = j(N_r + 1) + i$$

Индексы изменяются в следующих границах:

$$0 \le i \le N_r$$

$$0 \le j \le N_z$$

Тогда имеем:

$$0 \le m < (N_r + 1)(N_y + 1)$$

4.1 Запись для внутренних точек

Перепишем наше уравнение с использованием нового индеса для $i \in (0, N_r)$ и $j \in (0, N_z)$:

$$-\left[h_z r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j)\frac{v_{i+1,j} - v_{i,j}}{h_r} - h_z r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j)\frac{v_{i,j} - v_{i-1,j}}{h_r}\right.$$
$$\left.+h_r r_i k_2(r_i, z_{j+\frac{1}{2}})\frac{v_{i,j+1} - v_{i,j}}{h_z} - h_r r_i k_2(r_i, z_{j-\frac{1}{2}})\frac{v_{i,j} - v_{i,j-1}}{h_z}\right] = h_r h_z r_i f_{i,j}$$

$$-\frac{h_r}{h_z} r_i k_2(r_i, z_{j-\frac{1}{2}}) v_{i,j-1} - \frac{h_z}{h_r} r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j) v_{i-1,j} +$$
$$+\left[\frac{h_z}{h_r} r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j) + \frac{h_z}{h_r} r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j) + \frac{h_r}{h_z} r_i k_2(r_i, z_{j+\frac{1}{2}}) + \frac{h_r}{h_z} r_i k_2(r_i, z_{j-\frac{1}{2}})\right] v_{i,j}$$
$$-\frac{h_z}{h_r} r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j) v_{i+1,j} - \frac{h_r}{h_z} r_i k_2(r_i, z_{j+1}) v_{i,j+\frac{1}{2}} = h_r h_z r_i f_{i,j}$$

Введём новые обозначения:

$$a_m w_{m-L} + b_m w_{m-1} + c_m w_m + d_m w_{m+1} + e_m w_{m+L} = g_m$$

где коэффициенты равны:

$$a_m = -\frac{h_r}{h_z} r_i k_2(r_i, z_{j-\frac{1}{2}})$$

$$b_m = -\frac{h_z}{h_r} r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j)$$

$$c_m = \frac{h_z}{h_r} r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j) + \frac{h_z}{h_r} r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j) + \frac{h_r}{h_z} r_i k_2(r_i, z_{j+\frac{1}{2}}) + \frac{h_r}{h_z} r_i k_2(r_i, z_{j-\frac{1}{2}})$$

$$d_m = -\frac{h_z}{h_r} r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_j)$$

$$e_m = -\frac{h_r}{h_z} r_i k_2(r_i, z_{j+\frac{1}{2}})$$

$$g_m = h_r h_z r_i f_{i,j}$$

4.2 Запись для левой границы

Перепишем наше уравнение с использованием нового индекса для $i = 0$ и $j \in (0, N_z)$:

$$-\Bigg[h_z r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_j) \frac{v_{1,j} - v_{0,j}}{h_r} - 0$$
$$+ \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j+\frac{1}{2}}) \frac{v_{0,j+1} - v_{0,j}}{h_z} - \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j-\frac{1}{2}}) \frac{v_{0,j} - v_{0,j-1}}{h_z}\Bigg] = \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} h_z f_{0,j}$$

$$-\frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j-\frac{1}{2}}) v_{0,j-1} +$$
$$+ \left[\frac{h_z}{h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_j) + \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j+\frac{1}{2}}) + \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j-\frac{1}{2}})\right] v_{0,j}$$
$$-\frac{h_z}{h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_j) v_{1,j} - \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j+\frac{1}{2}}) v_{0,j+1} = \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} h_z f_{0,j}$$

Введём новые обозначения:

$$a_m w_{m-L} + c_m w_m + d_m w_{m+1} + e_m w_{m+L} = g_m$$

где коэффициенты равны:

$$a_m = -\frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j-\frac{1}{2}})$$

$$c_m = \frac{h_z}{h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_j) + \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j+\frac{1}{2}}) + \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j-\frac{1}{2}})$$

$$d_m = -\frac{h_z}{h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_j)$$

$$e_m = -\frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{j+\frac{1}{2}})$$

$$g_m = \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} h_z f_{0,j}$$

4.3 Запись для правой границы

Перепишем наше уравнение с использованием нового индекса для $i = N_r$ и $j \in (0, N_z)$:

$$-\left[-h_z r_{N_r}(\chi_2 v_{N_r,j} - \varphi_2(z)) - h_z r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_j)\frac{v_{N_r,j} - v_{N_r-1,j}}{h_r}\right.$$
$$\left.+\frac{h_r}{2} r_{N_r} k_2(r_{N_r}, z_{j+\frac{1}{2}})\frac{v_{N_r,j+1} - v_{N_r,j}}{h_z} - \frac{h_r}{2} r_{N_r} k_2(r_{N_r}, z_{j-\frac{1}{2}})\frac{v_{N_r,j} - v_{N_r,j-1}}{h_z}\right] = \frac{h_r}{2} r_{N_r} h_z f_{N_r,j}$$

$$-\frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{j-\frac{1}{2}}) v_{N_r,j-1} - \frac{h_z}{h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_j) v_{N_r-1,j}$$
$$+\left[h_z r_{N_r}\chi_2 + \frac{h_z}{h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_j) + \frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r, z_{j+\frac{1}{2}}}) + \frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{j-\frac{1}{2}})\right] v_{N_r,j}$$
$$-\frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{j+\frac{1}{2}}) v_{N_r,j+1} = \frac{h_r}{2} r_{N_r} h_z f_{N_r,j} + h_z r_{N_r}\varphi_2(z)$$

Введём новые обозначения:

$$a_m w_{m-L} + b_m w_{m-1} + c_m w_m + e_m w_{m+L} = g_m$$

где коэффициенты:

$$a_m = -\frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{j-\frac{1}{2}})$$

$$b_m = -\frac{h_z}{h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_j)$$

$$c_m = h_z r_{N_r}\chi_2 + \frac{h_z}{h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_j) + \frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r, z_{j+\frac{1}{2}}}) + \frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{j-\frac{1}{2}})$$

$$e_m = -\frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{j+\frac{1}{2}})$$

$$g_m = \frac{h_r}{2} r_{N_r} h_z f_{N_r,j} + h_z r_{N_r}\varphi_2(z)$$

4.4 Запись для нижней границы

Перепишем наше уравнение с использованием нового индекса для $i \in (0, N_r)$ и $j = 0$:

$$-\left[\frac{h_z}{2} r_{i+\frac{1}{2}} k_1(r_{i+\frac{1}{2}}, z_0)\frac{v_{i+1,0} - v_{i,0}}{h_r} - \frac{h_z}{2} r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_0)\frac{v_{i,0} - v_{i-1,0}}{h_r}\right.$$
$$\left.+h_r r_i k_2(r_i, z_{\frac{1}{2}})\frac{v_{i,1} - v_{i,0}}{h_z} - h_r r_i(\chi_3 v_{i,0} - \varphi_3(r))\right] = h_r \frac{h_z}{2} r_i f_{i,0}$$

$$
\begin{aligned}
&-\frac{h_z}{2h_r}v_{i-\frac{1}{2}}k_1(r_{i-\frac{1}{2}},z_0)v_{i-1,0}\\
&+\left[h_r r_i\chi_3+\frac{h_z}{2h_r}r_{i+\frac{1}{2}}k_1(r_{i+\frac{1}{2}},z_0)+\frac{h_z}{2h_r}r_{i-\frac{1}{2}}k_1(r_{i-\frac{1}{2}},z_0)+\frac{h_r}{h_z}r_ik_2(r_i,z_{\frac{1}{2}})\right]v_{i,0}\\
&-\frac{h_z}{2h_r}r_{i+\frac{1}{2}}k_1(r_{i+\frac{1}{2}},z_0)v_{i+1,0}-\frac{h_r}{h_z}r_ik_2(r_i,z_{\frac{1}{2}})v_{i,1}=h_r\frac{h_z}{2}r_if_{i,0}+h_rr_i\varphi(r)
\end{aligned}
$$

Введём новые обозначения:

$$b_mw_{m-1}+c_mw_m+d_mw_{m+1}+e_mw_{m+L}=g_m$$

где коэффициенты ранвы:

$$b_m=-\frac{h_z}{2h_r}r_{i-\frac{1}{2}}k_1(r_{i-\frac{1}{2}},z_0)$$

$$c_m=h_rr_i\chi_3+\frac{h_z}{2h_r}r_{i+\frac{1}{2}}k_1(r_{i+\frac{1}{2}},z_0)+\frac{h_z}{2h_r}r_{i-\frac{1}{2}}k_1(r_{i-\frac{1}{2}},z_0)+\frac{h_r}{h_z}r_ik_2(r_i,z_{\frac{1}{2}})$$

$$d_m=-\frac{h_z}{2h_r}r_{i+\frac{1}{2}}k_1(r_{i+\frac{1}{2}},z_0)$$

$$e_m=-\frac{h_r}{h_z}r_ik_2(r_i,z_{\frac{1}{2}})$$

$$g_m=h_r\frac{h_z}{2}r_if_{i,0}+h_rr_i\varphi_3(r)$$

4.5 Запись для верхней границы

Перепишем наше уравнение с использованием нового индекса для $i \in [0, N_r]$ и $j = N_z$:

$$v_{i,N_z}=\varphi(r_i)$$

Перейдём к новым обозначениям:

$$c_mw_m=\varphi_m$$

где:

$$c_m=1,\quad \varphi_m=\varphi_4(r_i)$$

4.6 Запись для левой нижней граничной точки

Перепишем наше уравнение с использованием нового индекса для $i = 0$ и $j = 0$:

$$-\left[\frac{h_z}{2} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_0) \frac{v_{1,0} - v_{0,0}}{h_r} - 0 \right.$$
$$\left. + \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{\frac{1}{2}}) \frac{v_{0,1} - v_{0,0}}{h_z} - \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} (\chi_3 v_{0,0} - \varphi_3(r)) \right] = \frac{h_r}{2} \frac{h_z}{2} \frac{r_{\frac{1}{2}}}{2} f_{0,0}$$

$$\left[\frac{h_z}{2h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_0) + \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{\frac{1}{2}}) + \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} \chi_3 \right] v_{0,0} -$$
$$- \frac{h_z}{2h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_0) v_{1,0} - \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{\frac{1}{2}}) v_{0,1} = \frac{h_r}{2} \frac{h_z}{2} \frac{r_{\frac{1}{2}}}{2} f_{0,0} + \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} \varphi_3(r_0)$$

Введём новые обозначения:

$$c_m w_m + d_m w_{m+1} + e_m w_{m+L} = g_m$$

где коэффициенты ранвы:

$$c_m = \frac{h_z}{2h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_0) + \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{\frac{1}{2}}) + \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} \chi_3$$

$$d_m = -\frac{h_z}{2h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_0)$$

$$e_m = -\frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{\frac{1}{2}})$$

$$g_m = \frac{h_r}{2} \frac{h_z}{2} \frac{r_{\frac{1}{2}}}{2} f_{0,0} + \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} \varphi_3(r_0)$$

4.7 Запись для правой нижней граничной точки

Перепишем наше уравнение с использованием нового индекса для $i = N_r$ и $j = 0$:

$$-\left[-\frac{h_z}{2} r_{N_r} (\chi_2 v_{N_r,0} - \varphi_2(z)) - \frac{h_z}{2} r_{N_r - \frac{1}{2}} k_1(r_{N_r - \frac{1}{2}}, z_0) \frac{v_{N_r,0} - v_{N_r-1,0}}{h_r} \right.$$
$$\left. + \frac{h_r}{2} r_{N_r} k_2(r_{N_r}, z_{\frac{1}{2}}) \frac{v_{N_r,1} - v_{N_r,0}}{h_z} - \frac{h_r}{2} r_{N_r} (\chi_3 v_{N_r,0} - \varphi_3(r)) \right] = \frac{h_r}{2} \frac{h_z}{2} r_{N_r} f_{N_r,0}$$

$$
-\frac{h_z}{2h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_0) v_{N_r-1,0} +
$$
$$
+\left[\frac{h_z}{2} r_{N_r}\chi_2 + \frac{h_z}{2h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_0) + \frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r, z_{\frac{1}{2}}}) + \frac{h_r}{2} r_{N_r}\chi_3\right] v_{N_r,0} -
$$
$$
-\frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{\frac{1}{2}}) v_{N_r,1} = \frac{h_r}{2}\frac{h_z}{2} r_{N_r} f_{N_r,0} + \frac{h_z}{2} r_{N_r}\varphi_2(z) + \frac{h_r}{2} r_{N_r}\varphi_3(r)
$$

Введём новые обозначения:

$$
b_m w_{m-1} + c_m w_m + e_m w_{m+L} = g_m
$$

где коэффициенты равны:

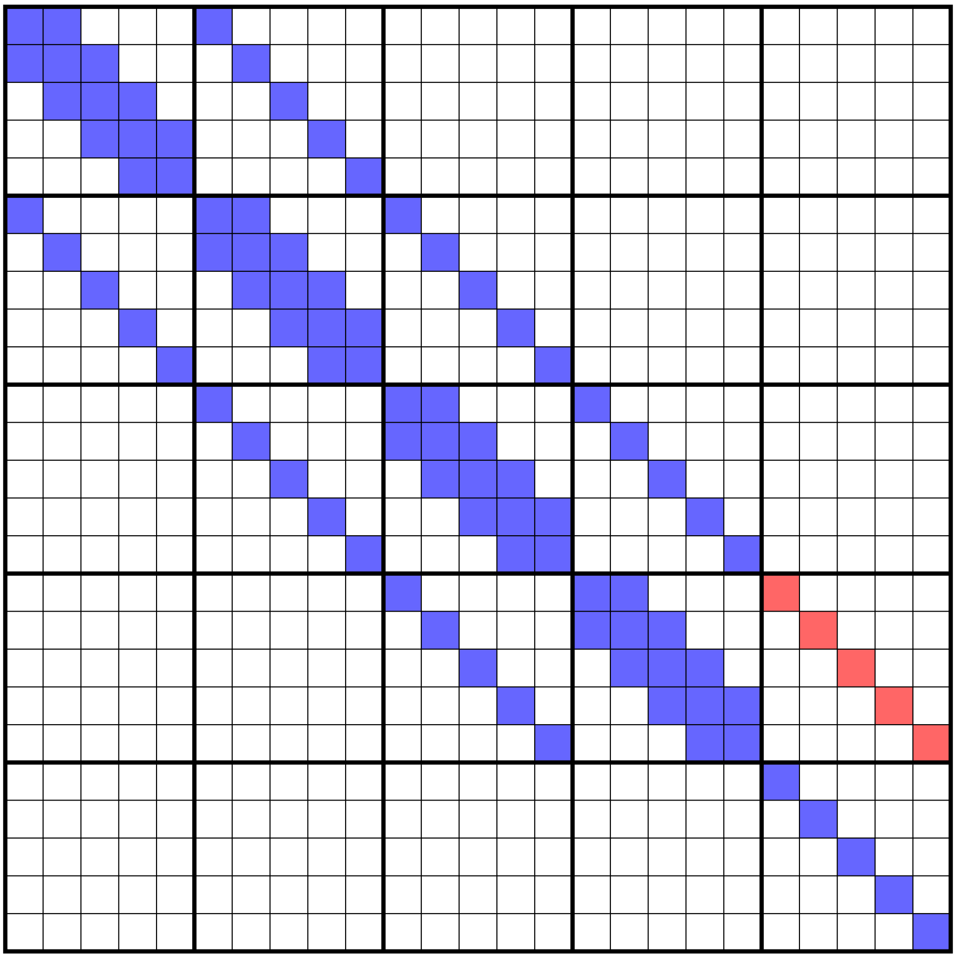
$$
b_m = -\frac{h_z}{2h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_0)
$$

$$
c_m = \frac{h_z}{2} r_{N_r}\chi_2 + \frac{h_z}{2h_r} r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_0) + \frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r, z_{\frac{1}{2}}}) + \frac{h_r}{2} r_{N_r}\chi_3
$$

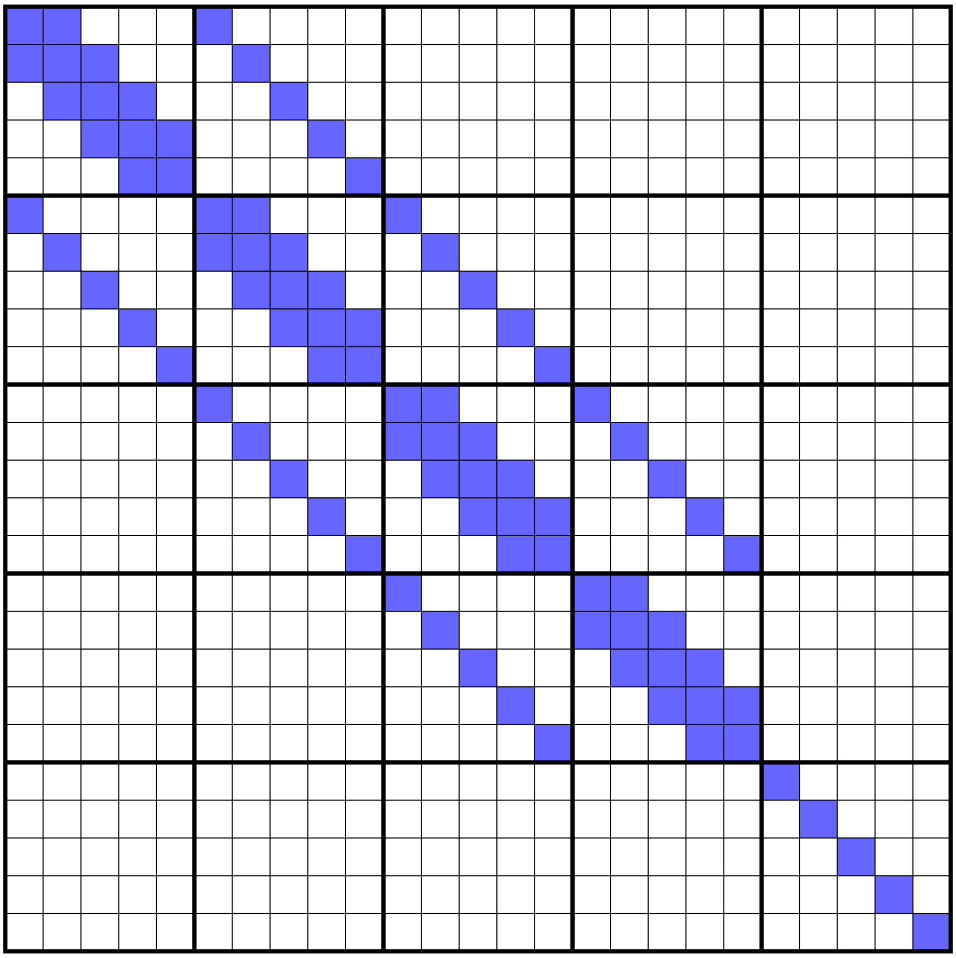
$$
e_m = -\frac{h_r}{2h_z} r_{N_r} k_2(r_{N_r}, z_{\frac{1}{2}})
$$

$$
g_m = \frac{h_r}{2}\frac{h_z}{2} r_{N_r} f_{N_r,0} + \frac{h_z}{2} r_{N_r}\varphi_2(z) + \frac{h_r}{2} r_{N_r}\varphi_3(r)
$$

В итоге мы получаем СЛАУ с нессиметричной матрицей, а метод сопряжённых градиентов требует от нас использования системы с симметричной матрицей. Для того чтобы привести нашу матрицу к симметричной, воспользуемся сложением и вычитанием строк.



В итоге мы получаем симметричную матрицу:



Данную систему мы будем решать методом сопряжённых градиентов.

5 Метод сопряжённых градиентов

Для решения системы линейных алгебраических уравнений использовать метод сопряженных градиентов с предобусловливанием. Опишем метод решения следующей системы:

$$Ax = b$$

где A - симметричная, положительно определённая, матрица.

5.1 Явный метод

$r^{(0)} = b - Ax^{(0)}$
$s^{(1)} = r^{(0)}$
$\gamma = \sqrt{(b, b)}$
for $k = 1$ **to** k_{max} **do**
    $\alpha_k = \dfrac{\left(r^{(k-1)},\ r^{(k-1)}\right)}{\left(As^{(k)},\ s^{(k)}\right)}$
    $x^{(k)} = x^{(k-1)} + \alpha_k s^{(k)}$
    $r^{(k)} = r^{(k-1)} - \alpha_k s^{(k-1)}$
    if $\sqrt{\left(r^{(k)},\ r^{(k)}\right)} < \gamma\varepsilon$ **then**
        break
    end
    $\beta_k = \dfrac{\left(r^{(k)},\ r^{(k)}\right)}{\left(r^{(k-1)},\ r^{(k-1)}\right)}$
    $s^{(k+1)} = r^{(k)} + \beta_k s^{(k)}$
end

5.2 Неявный метод

Неявный метод основывается на использовании предобуславливания. Идея заключается в том, чтобы выбрать матрицу **B**, которая является симметричной и положительно определенной, и приблизительно равна матрице **A**. Мы выбираем **B** в виде $B = \tilde{L}\tilde{L}^T$, где $\tilde{L}\tilde{L}^T$ представляет собой неполное разложение Холецкого для матрицы **A**. Позже мы объясним, почему такой выбор матрицы **B** считается оптимальным.

$r^{(0)} = b - Ax^{(0)}$
$Bw^{(0)} = r^{(0)}$
$s^{(1)} = w^{(0)}$
$Bg = b$
$\gamma = \sqrt{(g,\ b)}$
for $k = 1$ **to** k_{max} **do**
    $\alpha_k = \dfrac{\left(w^{(k-1)},\ r^{(k-1)}\right)}{\left(As^{(k)},\ s^{(k)}\right)}$
    $x^{(k)} = x^{(k-1)} + \alpha_k s^{(k)}$
    $r^{(k)} = r^{(k-1)} - \alpha_k s^{(k-1)}$
    $Bw^{(k)} = r^{(k)}$
    if $\sqrt{\left(w^{(k)},\ r^{(k)}\right)} < \gamma\varepsilon$ **then**
        break
    end
    $\beta_k = \dfrac{\left(w^{(k)},\ r^{(k)}\right)}{\left(w^{(k-1)},\ r^{(k-1)}\right)}$
    $s^{(k+1)} = w^{(k)} + \beta_k s^{(k)}$
end

Для решения системы $Bw(k) = r(k)$ мы используем метод Гаусса. При выборе $B = \tilde{L}\tilde{L}^T$, нам остается выполнить только две обратные подстановки: $\tilde{L}z^{(k)} = r^k$ и $\tilde{L}^T w^{(k)} = z^{(k)}$.

6 Тестирование

Протестируем постренную модель на следующих тестовых наборах:

№ теста	$k_1(r,z)$	$k_2(r,z)$	$u(r,z)$
1	$r+2z$	$3r+4z$	1
2	$(r+2z)^2$	$(3r+4z)^2$	1
3	$(r+2z)^2$	$(3r+4z)^2$	$r^2(z+1)$
4	$(r+2z)^3$	$(3r+4z)^3$	$r^2(z^2+z+1)$

Все тесты имеют общие значения:

$$r \in [0,1], \quad z \in [0,2]$$

$$\chi_2 = 2, \quad \chi_3 = 3$$

Функция f, φ_2, φ_3, φ_4 вычисляют программно с использованием символичных вычислений.

Погрешность решения задачи вычисляется следующим образом:

$$\delta_1 = \frac{||v - \tilde{v}||_1}{||v||_1}$$

$$\delta_2 = \frac{||v - \tilde{v}||_2}{||v||_2}$$

$$\delta_3 = \frac{||v - \tilde{v}||_\infty}{||v||_\infty}$$

Значение ε для метода сопряжённых градиентов равно 10^{-8}.

Тест №1

$$\varphi_2 = 2.0$$

$$\varphi_3 = 3.0$$

$$\varphi_4 = 1.0$$

$$f = 0$$

Имеем лишь ошибку округления.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	8	$9.899e-16$	$3.92e-16$	$3.67e-16$
4	4	23	$8.69e-11$	$7.79e-11$	$5.59e-11$
8	8	63	$8.44e-10$	$4.12e-10$	$1.51e-10$
16	16	126	$3.92e-10$	$2.99e-10$	$7.03e-10$
32	32	310	$6.49e-09$	$7.98e-09$	$3.19e-09$

Таблица 1: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	5	$6.52e-16$	$7.22e-16$	$3.38e-16$
4	4	8	$2.789e-11$	$4.15e-11$	$9.47e-11$
8	8	16	$2.805e-10$	$9.75e-10$	$3.06e-10$
16	16	23	$9.96e-10$	$1.96e-10$	$6.67e-10$
32	32	49	$2.97e-9$	$9.76e-9$	$3.46e-9$

Таблица 2: Неявный метод

Тест №2

$$\varphi_2 = 2.0$$

$$\varphi_3 = 3.0$$

$$\varphi_4 = 1.0$$

$$f = 0$$

Присутствует лишь ошибка округления.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	8	$5.31e-16$	$4.38e-16$	$2.10e-16$
4	4	25	$1.86e-11$	$7.96e-11$	$9.29e-11$
8	8	60	$2.71e-10$	$9.19e-10$	$1.81e-10$
16	16	111	$6.7e-10$	$6.74e-10$	$1.41e-10$
32	32	285	$5.71e-9$	$8.04e-9$	$9.09e-9$

Таблица 3: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	5	$3.57e-16$	$5.312e-16$	$3.31e-16$
4	4	8	$5.51e-11$	$9.49e-11$	$6.03e-11$
8	8	16	$2.88e-10$	$1.67e-10$	$9.416e-10$
16	16	24	$5.48e-10$	$5.97e-10$	$9.06e-10$
32	32	41	$1.77e-9$	$2.28e-9$	$1.61e-9$

Таблица 4: Неявный метод

Тест №3

$$\varphi_2 = 2(z+1) + 2(1+2z)^2(z+1)$$

$$\varphi_3 = -9r^4 + 3r^2$$

$$\varphi_4 = 3r^2$$

$$f = -8r^2\left(3r+4z\right) - 4r\left(r+2z\right)\left(z+1\right) - 4\left(r+2z\right)^2\left(z+1\right)$$

Присутствует ошибка округления и ошибка аппроксимации.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	7	$7.882e-2$	$3.49e-2$	$2.192e-2$
4	4	21	$1.962e-2$	$8.761e-3$	$5.477e-3$
8	8	53	$4.948e-3$	$2.186e-3$	$1.358e-3$
16	16	142	$1.222e-3$	$5.39e-4$	$3.419e-4$
32	32	247	$3.377e-4$	$1.266e-4$	$8.486e-5$

Таблица 5: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	5	$7.882e-2$	$3.49e-2$	$2.192e-2$
4	4	9	$1.962e-2$	$8.761e-3$	$5.477e-3$
8	8	15	$4.948e-3$	$2.186e-3$	$1.358e-3$
16	16	26	$1.222e-3$	$5.39e-4$	$3.419e-4$
32	32	45	$3.377e-4$	$1.266e-4$	$8.486e-5$

Таблица 6: Неявный метод

Тест №4

$$\varphi_2 = 2(z^2+z+1) + 2(1+2z)^3(z^2+z+1)$$

$$\varphi_3 = -27r^5 + 3r^2$$

$$\varphi_4 = 7r^2$$

$$f = -2r^2\left(3r+4z\right)^2\left(3r+16z+6\right) - 6r\left(r+2z\right)^2\left(z^2+z+1\right) - 4\left(r+2z\right)^3\left(z^2+z+1\right)$$

Присутствует ошибка округления и ошибка аппроксимации.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	9	$5.68e-2$	$6.32e-2$	$7.45e-2$
4	4	27	$1.422e-2$	$1.579e-2$	$1.854e-2$
8	8	54	$3.56e-3$	$3.937e-3$	$4.67e-3$
16	16	138	$8.87e-4$	$9.815e-4$	$1.161e-3$
32	32	284	$2.108e-4$	$2.38e-4$	$2.846e-4$

Таблица 7: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	5	$5.68e-2$	$6.32e-2$	$7.45e-2$
4	4	8	$1.422e-2$	$1.579e-2$	$1.854e-2$
8	8	15	$3.56e-3$	$3.937e-3$	$4.67e-3$
16	16	23	$8.87e-4$	$9.815e-4$	$1.161e-3$
32	32	42	$2.108e-4$	$2.38e-4$	$2.846e-4$

Таблица 8: Неявный метод

По значениям наших тестов можно сделать следующие выводы:

- Неявный метод имеет значительно большую скорость сходимости.
- Увеличение числа разбиений в два раза приводит к уменьшению погрешности примерно в четыре раза при наличии ошибки аппроксимации.

7 Заключение

В рамках данной курсовой работы мы использовали свои практические знания для написания программного обеспечения, которое моделирует стационарные процессы теплопроводности. Мы построили дискретную модель и протестировали её.

Была разработана программа моделирования стационарных процессов теплопроводности. Привидены результаты экспериментального исследования точности программного обеспечения.

Код программы

grid.py

```python
from typing import Tuple, Callable


def get_x(a: float, b: float, Nx: int) -> Tuple[Callable[[int], float], Callable[[int], float]]:
    hx = (b - a) / Nx

    def inner1(i: int) -> float:
        """returns x_{i}"""
        if i < 0 or i > Nx:
            raise KeyError()

        return a + i * hx

    def inner2(i: int) -> float:
        """returns x_{i+1/2}"""
        if i < 0 or i >= Nx:
            raise KeyError()

        return a + (i + 0.5) * hx

    return (inner1, inner2)


def get_y(c: float, d: float, Ny: int) -> Tuple[Callable[[int], float], Callable[[int], float]]:
    hy = (d - c) / Ny

    def inner1(j: int) -> float:
        """returns y_{j}"""
        if j < 0 or j > Ny:
            raise KeyError()

        return c + j * hy

    def inner2(j: int) -> float:
        """returns y_{j+1/2}"""
        if j < 0 or j >= Ny:
            raise KeyError()

        return c + (j + 0.5) * hy

    return (inner1, inner2)
```

linalg.py

```
import math
from typing import List
from sparse_matrix import *


def is_symmetric(m: SparseMatrix):
    sz = m.size()

    for i in range(sz):
        for j in range(sz):
            if m.get(i, j) != m.get(j, i):
                return False

    return True


def add_vv(v1: List[float], v2: List[float]) -> List[float]:
    sz = len(v1)
    rez = [0] * sz

    for i in range(sz):
        rez[i] = v1[i] + v2[i]

    return rez

def mul_sv(a: float, v: List[float]) -> List[float]:
    sz = len(v)
    rez = [0] * sz

    for i in range(sz):
        rez[i] = v[i] * a

    return rez

def sub_vv(v1: List[float], v2: List[float]) -> List[float]:
    return add_vv(v1, mul_sv(-1, v2))

def mul_vv(v1: List[float], v2: List[float]) -> float:
    sz = len(v1)
    rez = 0

    for i in range(sz):
        rez += v1[i] * v2[i]

    return rez

```

```
def mul_mv(m: SparseMatrix, v: List[float]) -> List[float]:
    sz = len(v)
    rez = [0] * sz

    for i in range(sz):
        for j, d in m.get(i):
            if i == j:
                continue
            rez[i] += d * v[j]
            if m.is_symmetric():
                rez[j] += d * v[i]

    for i in range(sz):
        rez[i] += m.get(i, i) * v[i]

    return rez

def mul_mm(m1: SparseMatrix, m2: SparseMatrix) -> SparseMatrixCOO:
    sz = m1.size()
    rez = SparseMatrixCOO()

    for i in range(sz):
        for j in range(sz):
            acc = 0
            for k in range(sz):
                acc += m1.get(i, k) * m2.get(k, j)

            rez.set(i, j, acc)

    return rez

def transpose(m: SparseMatrix) -> SparseMatrixCOO:
    sz = m.size()
    rez = SparseMatrixCOO()

    for i in range(sz):
        for j, d in m.get(i):
            rez.set(j, i, d)

    return rez

def copy_v(v: List[float]) -> List[float]:
    sz = len(v)
    rez = [0] * sz

    for i in range(sz):
```

```
            rez[i] = v[i]

    return rez

def _cholesky_solve(L: SparseMatrix, r: List[float]) -> List[float]:
    sz = len(r)

    LT = transpose(L)

    z = [0] * sz
    for i in range(sz):
        acc = r[i]
        for j, d in L.get(i):
            if j < i:
                acc -= d * z[j]

        z[i] = acc / L.get(i, i)

    w = [0] * sz
    for i in range(sz - 1, -1, -1):
        acc = z[i]
        for j, d in LT.get(i):
            if j > i:
                acc -= d * w[j]

        w[i] = acc / LT.get(i, i)

    return w

def conjgrad(A: SparseMatrix, b: List[float], x: List[float],
             eps: float, maxiter: int=10000, L: SparseMatrix=None) -> List[float]:
    nsteps = 0

    if L == None:
        r = sub_vv(b, mul_mv(A, x))
        p = copy_v(r)
        gamma = math.sqrt( mul_vv(b, b) )

        rsold = mul_vv(r, r)

        while nsteps < maxiter:
            nsteps += 1

            Ap = mul_mv(A, p)
            alpha = rsold / mul_vv(p, Ap)

```

```
            x = add_vv(x, mul_sv(alpha, p))
            r = sub_vv(r, mul_sv(alpha, Ap))

            rsnew = mul_vv(r, r)
            if math.sqrt(rsnew) < gamma * eps:
                break

            p = add_vv(r, mul_sv(rsnew / rsold, p))
            rsold = rsnew
    else:
        r = sub_vv(b, mul_mv(A, x))
        w = _cholesky_solve(L, r)
        p = w
        g = _cholesky_solve(L, b)
        gamma = math.sqrt( mul_vv(g, b) )

        rsold = mul_vv(w, r)

        while nsteps < maxiter:
            nsteps += 1

            Ap = mul_mv(A, p)
            alpha = rsold / mul_vv(p, Ap)

            x = add_vv(x, mul_sv(alpha, p))
            r = sub_vv(r, mul_sv(alpha, Ap))
            w = _cholesky_solve(L, r)

            rsnew = mul_vv(w, r)
            if math.sqrt(rsnew) < gamma * eps:
                break

            p = add_vv(w, mul_sv(rsnew / rsold, p))
            rsold = rsnew

    return (x, nsteps)
```

main.py

```
from sympy import symbols, Float, log, cos, sin
from utils import *

import math

from model import Model
from linalg import *
from grid import *

```

```
a = 0
b = 1
c = 0
d = 2

chi2 = 2
chi3 = 3

x, y = symbols("x y")

def print_matrix(m):
    dummy = ""

    sz = m.size()

    print(f"{dummy:3s}", end=" ")
    for i in range(sz):
        print(f"{i:7d}", end=" ")
    print()

    for i in range(sz):
        print(f"{i:3d}", end=" ")
        for j in range(sz):
            x = m.get(i, j)
            if x == 0:
                print(f" {0:+7.2f}", end=" ")
            else:
                print(f" {m.get(i, j):+7.2f}", end=" ")
        print()

def print_matrix2(m):
    dummy = ""

    sz = m.size()

    # print(f"{dummy:3s}", end=" ")
    for i in range(sz):
        for j in range(sz):
            x = m.get(i, j)
            if x == 0:
                pass
            else:
                print("\\fill[color=blue!60] (", i, "," , sz - j, ")" , "rectangle (",
                 ↪  j,",", sz - i,");")
        print()

```

```python
def print_vector(label, v):
    print(f"{label}: ", end="")

    for x in v:
        print(f"{x:+.5f}", end=" ")

    print()


test_cases = [
    # ( (x + y + 1)**2, (x + y + 1)**3, Float(1) * x**2 ),
    # ( (x + y + 1), (x + y + 1), (2 * x + 3 * y + 1) ),
    ( (x + y + 1)**2, (x + y + 1)**2, (2 * x + 3 * y + 1) * x**2 ),
    # ( (x + y + 1)**3, (x + y + 1)**3, (2 * x + 3 * y + 1) ),
    # ( Float(1), Float(1), (2 * x + 3 * y + 1)**2 ),
    # ( (x + y + 1), (x + y + 1), (2 * x + 3 * y + 1)**2 ),
    # ( (x + y + 1), (x + y + 1), (2 * x + 3 * y + 1)**3 * x**2),
    # ( (cos(x * y) + 2), (sin(x) * sin(y) + 2), (x * log(y + 1)) ),
]

fmt = "{:>2s}, {:>3s}, {:>3s}, {:>4s}, {:>8s}, {:>8s}, {:>8s}, {:>4s}, {:>8s}, {:>8s},
↪ {:>8s}"
header = fmt.format(
    "no", "Nx", "Ny",
    "n1", "delta11", "delta12", "delta13",
    "n2", "delta21", "delta22", "delta23"
)
print(header)

for test_num, test_case in enumerate(test_cases):

    k1, k2, u = test_case

    g2 = get_g2(b, chi2, k1, u)
    g3 = get_g3(c, chi3, k2, u)
    g4 = get_g4(d, u)

    f = get_f(u, k1, k2)
    k1 = get_k1(k1)
    k2 = get_k2(k2)
    u = get_u(u)

    mod = Model(a, b, c, d, [k1, k2], f, [chi2, chi3], [g2, g3, g4])

    sizes = [
        # (2, 2),
```

```
        (4, 4),
        # (8, 8),
        # (16, 16),
        # (32, 32),
        # (64, 64),
    ]

    for Nx, Ny in sizes:
        A, g, L = mod.init_data(Nx, Ny)
        sz = A.size()

        # for i in g:
        #     print("g =", i)

        x1, _ = get_x(a, b, Nx)
        y1, _ = get_y(c, d, Ny)

        x0 = [0] * sz

        x = []
        for j in range(Ny + 1):
            for i in range(Nx + 1):
                x.append(u(x1(i), y1(j)))

        print_matrix2(A)

        # y, n_steps_1 = conjgrad(A, g, x0, eps=1e-12)

        # # print_vector("x", x)
        # # print_vector("y", y)
        # # print_vector("g", g)
        # # print_vector("Ay", mul_mv(A, y))
        # # print_vector("Ax", mul_mv(A, x))

        # delta = sub_vv(x, y)

        # delta_11 = sum( [abs(i) for i in delta] ) / sum( [abs(i) for i in x] )
        # delta_12 = math.sqrt( mul_vv(delta, delta) / mul_vv(x, x) )
        # delta_13 = abs( max(delta, key=lambda x: abs(x)) / max(x, key=lambda x: abs(x)) )

        # y, n_steps_2 = conjgrad(A, g, x0, eps=1e-8, L=L)

        # delta = sub_vv(x, y)

        # delta_21 = sum( [abs(i) for i in delta] ) / sum( [abs(i) for i in x] )
```

```
        # delta_22 = math.sqrt( mul_vv(delta, delta) / mul_vv(x, x) )
        # delta_23 = abs( max(delta, key=lambda x: abs(x)) / max(x, key=lambda x:
        ↪   abs(x)) )

        # fmt = "{:2d}, {:3d}, {:3d}, {:4d}, {:8.5e}, {:8.5e}, {:8.5e}, {:4d}, {:8.5e},
        ↪  {:8.5e}, {:8.5e}"
        # record = fmt.format(
        #     test_num, Nx, Ny,
        #     n_steps_1, float(delta_11), float(delta_12), float(delta_13),
        #     n_steps_2, float(delta_21), float(delta_22), float(delta_23),
        # )
        # # print(record, flush=True)
```

model.py

```
import math
from typing import Callable, Tuple

from grid import *
from sparse_matrix import *
from linalg import *

real1_fn = Callable[[float], float]
real2_fn = Callable[[float, float], float]


def _cholesky_decomp(A, Nx, Ny):
    n = (Nx + 1) * (Ny + 1)
    m = Nx + 1

    a = [A.get(i, i) for i in range(n)]
    b = [A.get(i, i + 1) for i in range(n - 1)]
    c = [A.get(i, i + m) for i in range(n - m)]

    a_new = []
    b_new = []
    c_new = []

    L = SparseMatrixCOO()

    for i in range(n):

        if i - 1 < 0:
            ai = math.sqrt(a[i])
        elif i - m < 0:
            ai = math.sqrt(a[i] - b_new[i - 1]**2)
        else:
            ai = math.sqrt(a[i] - b_new[i - 1]**2 - c_new[i - m]**2)
```

```python

            a_new.append(ai)

            if i < n - 1:
                bi = b[i] / ai
                b_new.append(bi)

            if i < n - m:
                ci = c[i] / ai
                c_new.append(ci)

        for i in range(n):
            L.set(i, i, a_new[i])

        for i in range(n - 1):
            L.set(i, i + 1, b_new[i])

        for i in range(n - m):
            L.set(i, i + m, c_new[i])

        return transpose(L)


class Model:
    def __init__(self,
                 a: float, b: float,
                 c: float, d: float,
                 k: Tuple[real2_fn],
                 f: real2_fn,
                 chi: Tuple[float],
                 g: Tuple[real1_fn]) -> None:
        self.a = a
        self.b = b
        self.c = c
        self.d = d
        self.k = k
        self.f = f
        self.chi = chi
        self.g = g

    def init_data(self, Nx: int, Ny: int):

        k1, k2 = self.k
        f = self.f

        chi2, chi3 = self.chi #
```

```
        g2, g3, g4 = self.g #\varphi       ,        ,

        hx = (self.b - self.a) / Nx
        hy = (self.d - self.c) / Ny

        x1, x2 = get_x(self.a, self.b, Nx)
        y1, y2 = get_y(self.c, self.d, Ny)

        M = (Nx + 1) * (Ny + 1)
        L = (Nx + 1)

        mtrx = SparseMatrixCOO()
        g = [0] * M

        # inner space
        for i in range(1, Nx):
            for j in range(1, Ny):
                m = j * L + i

                #print("inner space i =", i, "j =", j, "m =", m)

                a1 = (hx / hy) * x1(i)  * k2( x1(i), y2(j - 1) )
                a2 = (hy / hx) * x2(i - 1) * k1( x2(i - 1), y1(j) )
                a3 = (hy / hx) * x2(i) * k1( x2(i), y1(j) )
                a4 = (hx / hy) * x1(i) * k2( x1(i), y2(j) )

                mtrx.set(m, m - L, -a1)
                mtrx.set(m, m - 1, -a2)
                mtrx.set(m, m, a1 + a2 + a3 + a4)
                mtrx.set(m, m + 1, -a3)
                mtrx.set(m, m + L, -a4)
                g[m] = hx * hy * x1(i) * f(x1(i), y1(j))

        # left edge
        i = 0
        for j in range(1, Ny):
            m = j * L + i

            #print("left edge i=", i, "j =", j, "m =", m)

            a1 = (hx / hy / 2) * (x2(i) / 2) * k2( x1(i), y2(j - 1) )
            a3 = (hy / hx) * x2(i) * k1( x2(i), y1(j) )
            a4 = (hx / hy / 2) * (x2(i) / 2) * k2( x1(i), y2(j) )

            mtrx.set(m, m - L, -a1)
            mtrx.set(m, m, a1 + a3 + a4)
```

```
    mtrx.set(m, m + 1, -a3)
    mtrx.set(m, m + L, -a4)
    g[m] = (hx / 2) * (x2(i) / 2) * hy * f(x1(i), y1(j))

# right edge
i = Nx
for j in range(1, Ny):
    m = j * L + i

    #print("right edge i=", i, "j =", j, "m =", m)

    a1 = (hx / hy / 2) * x1(i) * k2(x1(i), y2(j - 1))
    a2 = (hy / hx) * x2(i - 1) * k1(x2(i - 1), y1(j))
    a3 = hy * x1(i) * chi2
    a4 = (hx / hy / 2) * x1(i) * k2(x1(i), y2(j))

    mtrx.set(m, m - L, -a1)
    mtrx.set(m, m - 1, -a2)
    mtrx.set(m, m, a1 + a2 + a3 + a4)
    mtrx.set(m, m + L, -a4)
    g[m] = (hx / 2) * x1(i) * hy * f(x1(i), y1(j)) + hy * x1(i) * g2(y1(j))

# bottom edge
j = 0
for i in range(1, Nx):
    m = j * L + i

    #print("bottom edge i=", i, "j =", j, "m =", m)

    a1 = hx * x1(i) * chi3
    a2 = (hy / hx / 2) * x2(i - 1) * k1( x2(i - 1), y1(j) )
    a3 = (hy / hx / 2) * x2(i) * k1( x2(i), y1(j) )
    a4 = (hx / hy) * x1(i) * k2( x1(i), y2(j) )

    mtrx.set(m, m - 1, -a2)
    mtrx.set(m, m, a1 + a2 + a3 + a4)
    mtrx.set(m, m + 1, -a3)
    mtrx.set(m, m + L, -a4)
    g[m] = hx * (hy / 2) * x1(i) * f(x1(i), y1(j)) + hx * x1(i) * g3(x1(i))

# top edge
j = Ny
for i in range(1, Nx):
    m = j * L + i

    #print("top edge i=", i, "j =", j, "m =", m)
```

```
            mtrx.set(m, m, 1)
            g[m] = g4(x1(i))

        # left bottom
        i, j = 0, 0
        m = j * L + i

        #print("left bottom i=", i, "j =", j, "m =", m)

        a1 = (hx / 2) * (x2(i) / 2) * chi3
        a3 = (hy / hx / 2) * x2(i) * k1(x2(i), y1(j))
        a4 = (hx / hy / 2) * (x2(i) / 2) * k2(x1(i), y2(j))

        mtrx.set(m, m, a1 + a3 + a4)
        mtrx.set(m, m + 1, -a3)
        mtrx.set(m, m + L, -a4)
        g[m] = (hx / 2) * (hy / 2) * (x2(i) / 2) * f(x1(i), y1(j)) + (hx / 2) * (x2(i) / 2) * g3(x1(i))

        # right bottom
        i, j = Nx, 0
        m = j * L + i

        #print("right bottom i=", i, "j =", j, "m =", m)

        a1 = (hy / 2) * x1(i) * chi2
        a2 = (hy / hx / 2) * x2(i - 1) * k1(x2(i - 1), y1(j))
        a3 = (hx / 2) * x1(i) * chi3
        a4 = (hx / hy / 2) * x1(i) * k2(x1(i), y2(j))

        mtrx.set(m, m - 1, -a2)
        mtrx.set(m, m, a1 + a2 + a3 + a4)
        mtrx.set(m, m + L, -a4)
        g[m] = (hx / 2) * (hy / 2) * x1(i) * f(x1(i), y1(j)) + (hy / 2) * x1(i) * g2(y1(i)) + (hx / 2) * x1(i) * g3(x1(i))

        # right top corner
        i, j = Nx, Ny
        m = j * L + i

        #print("right top i=", i, "j =", j, "m =", m)

        mtrx.set(m, m, 1)
        g[m] = g4(x1(i))
```

```
        # left top corner
        i, j = 0, Ny
        m = j * L + i

       # print("left top i=", i, "j =", j, "m =", m)

        mtrx.set(m, m, 1)
        g[m] = g4(x1(i))

        #remove extra elements
        # for i in range(1, Nx):
        #     m1 = (Ny - 1) * L + i
        #     m2 = Ny * L + i
        #     x = mtrx.get(m1, m1 + L)
        #     mtrx.set(m1, m1 + L, 0)

        #     g[m1] -= g[m2] * x

        # m1 = (Ny - 1) * L
        # m2 = Ny * L
        # x = mtrx.get(m1, m1 + L)
        # mtrx.set(m1, m1 + L, 0)

        # g[m1] -= g[m2] * x

        # m1 = (Ny - 1) * L + Nx
        # m2 = Ny * L + Nx
        # x = mtrx.get(m1, m1 + L)
        # mtrx.set(m1, m1 + L, 0)

        # g[m1] -= g[m2] * x

        # L = _cholesky_decomp(mtrx, Nx, Ny)

        # mtrx = mtrx.to_symmetric().to_CSR()
        print(f"LOG: {is_symmetric(mtrx)}")
        # L = L.to_CSR()

        return (mtrx, g, L)
```

sparse_matrix.py

```
from typing import Tuple, List
from abc import ABC, abstractmethod
import numpy as np


class SparseMatrix(ABC):
```

```
    @abstractmethod
    def get(self, i: int, j: int=None) -> float:
        pass

    @abstractmethod
    def size(self) -> int:
        pass

    @abstractmethod
    def is_symmetric() -> bool:
        pass


class SparseMatrixCOO(SparseMatrix):
    def __init__(self, symmetric=False) -> None:
        self.symmetric = symmetric
        self.row = []
        self.col = []
        self.data = []

    def get(self, i: int, j: int=None) -> float:
        # self._validate_index(i, j)

        if j == None:
            return self._get_by_row(i)

        if self.symmetric:
            i, j = self._get_symmetric_index(i, j)

        idx_pairs = zip(self.row, self.col)

        for idx, pair in enumerate(idx_pairs):
            ir, ic = pair

            if ir == i and ic == j:
                return self.data[idx]

        return 0

    def _get_by_row(self, i):
        rez = [(self.col[idx], self.data[idx]) for idx in range(len(self.row)) if
        ↪  self.row[idx] == i]
        return rez

    def set(self, i: int, j: int, x: float) -> None:
        # self._validate_index(i, j)
```

```python

        if self.symmetric:
            i, j = self._get_symmetric_index(i, j)

        idx_pair = zip(self.row, self.col)

        for idx, pair in enumerate(idx_pair):
            ir, ic = pair

            if ir == i and ic == j:
                if x == 0:
                    self.row.pop(idx)
                    self.col.pop(idx)
                    self.data.pop(idx)
                else:
                    self.data[idx] = x
                return

        if x != 0:
            self._add_value(i, j, x)

    def size(self) -> int:
        max_row = max(self.row) + 1
        max_col = max(self.col) + 1
        return max(max_row, max_col)

    def is_symmetric(self) -> bool:
        return self.symmetric

    def to_symmetric(self):
        m = SparseMatrixCOO(symmetric=True)
        for i, j, d in zip(self.row, self.col, self.data):
            m.set(i, j, d)
        return m

    def to_CSR(self):
        row = np.array(self.row)
        col = np.array(self.col)
        data = np.array(self.data)

        permutation = sorted(range( len(self.row) ), key=lambda idx: row[idx])

        row = row[permutation]
        col = col[permutation]
        data= data[permutation]

```

```python
        new_row = []

        for r in np.unique(row):
            index = row == r
            new_row.append( row.tolist().index(r) )

            col_per_row = col[index]
            data_per_row = data[index]

            permutation = sorted(range( len(col_per_row) ), key=lambda idx: col_per_row[idx])

            col_per_row = col_per_row[permutation]
            data_per_row = data_per_row[permutation]

            col[index] = col_per_row
            data[index] = data_per_row

        row = np.array(new_row)

        return SparseMatrixCSR(row, col, data, symmetric=self.symmetric)

    def _add_value(self, i: int, j: int, x: float) -> None:
        self.row.append(i)
        self.col.append(j)
        self.data.append(x)

    def _validate_index(self, i: int, j: int) -> None:
        if i < 0 or j < 0:
            raise KeyError()

    def _get_symmetric_index(self, i: int, j: int) -> Tuple[int, int]:
        return (j, i) if j > i else (i, j)


class SparseMatrixCSR(SparseMatrix):
    def __init__(self, row: np.ndarray, col: np.ndarray, data: np.ndarray, symmetric=False) -> None:
        self.symmetric = symmetric
        self.row = row
        self.col = col
        self.data = data

    def get(self, i: int, j: int=None) -> float:
        # self._validate_index(i, j)

```

```python
        if j == None:
            return self._get_by_row(i)

        if self.symmetric:
            i, j = self._get_symmetric_index(i, j)

        sz = self.size()

        ic1 = self.row[i]
        ic2 = None if i + 1 == sz else self.row[i + 1]

        try:
            idx = ic1 + self.col[ic1:ic2].tolist().index(j)
        except ValueError:
            return 0

        return self.data[idx]

    def _get_by_row(self, i):
        sz = self.size()

        ic1 = self.row[i]
        ic2 = None if i + 1 == sz else self.row[i + 1]

        col = self.col[ic1:ic2].tolist()
        data = self.data[ic1:ic2].tolist()

        return [(j, d) for j, d in zip(col, data)]

    def size(self) -> int:
        return self.col.max() + 1

    def is_symmetric(self) -> bool:
        return self.symmetric

    def _validate_index(self, i: int, j: int) -> None:
        sz = self.size()
        if i < 0 or i > sz:
            raise KeyError()
        if j < 0 or j > sz:
            raise KeyError()

    def _get_symmetric_index(self, i: int, j: int) -> Tuple[int, int]:
        return (j, i) if j > i else (i, j)
```

utils.py

```python
from typing import Callable
```

```python
from sympy import Symbol, symbols, simplify, Subs, N, diff


def get_u(u: Symbol) -> Callable[[float, float], float]:
    def inner(x: float, y: float):
        expr = Subs(u, "x", x)
        expr = Subs(expr, "y", y)
        return N(expr)

    return inner

def get_k1(k1: Symbol) -> Callable[[float, float], float]:
    def inner(x: float, y: float):
        expr = Subs(k1, "x", x)
        expr = Subs(expr, "y", y)
        return N(expr)

    return inner

def get_k2(k2: Symbol) -> Callable[[float, float], float]:
    def inner(x: float, y: float):
        expr = Subs(k2, "x", x)
        expr = Subs(expr, "y", y)
        return N(expr)

    return inner

def get_f(u: Symbol, k1: Symbol, k2: Symbol) -> Callable[[float, float], float]:
    x= symbols("x")
    expr1 = x * k1 * diff(u, "x")
    expr2 = k2 * diff(u, "y")
    f = -( (1 / x) * diff(expr1, "x") + diff(expr2, "y") )
    f = simplify(f)

    # print("f =", f)

    def inner(x: float, y: float):
        expr = Subs(f, "x", x)
        expr = Subs(expr, "y", y)
        return N(expr)

    return inner

# def get_g1(a: float, chi1: float, k1: Symbol, u: Symbol) -> Callable[[float, float],
↪ float]:
#     g1 = Subs(chi1 * u - k1 * diff(u, "x"), "x", a)
```

```
#     def inner(y: float):
#         return N( Subs(g1, "y", y) )

#     return inner

# def get_g2(b: float, u: Symbol) -> Callable[[float, float], float]:
#     g2 = Subs(u, "x", b)
#     def inner(y: float):
#         return N( Subs(g2, "y", y) )

#     return inner

# def get_g3(c: float, chi3: float, k2: Symbol, u: Symbol) -> Callable[[float, float], float]:
#     g3 = Subs(chi3 * u - k2 * diff(u, "y"), "y", c)
#     def inner(x: float):
#         return N( Subs(g3, "x", x) )

#     return inner

# def get_g4(d: float, chi4: float, k2: Symbol, u: Symbol) -> Callable[[float, float], float]:
#     g4 = Subs(chi4 * u + k2 * diff(u, "y"), "y", d)
#     def inner(x: float):
#         return N( Subs(g4, "x", x) )

#     return inner

def get_g2(b: float, chi2: float, k1: Symbol, u: Symbol) -> Callable[[float, float], float]:
    g2 = Subs(chi2 * u + k1 * diff(u, "x"), "x", b)
    def inner(y:float):
        return N( Subs(g2, "y", y))

    return inner

def get_g3(c: float, chi3: float, k2: Symbol, u: Symbol) -> Callable[[float, float], float]:
    g3 = Subs(chi3 * u - k2 * diff(u, "y"), "y", c)
    def inner(x: float):
        return N( Subs(g3, "x", x))

    return inner

def get_g4(d: float, u: Symbol) -> Callable[[float, float], float]:
    g4 = Subs(u, "y", d)
```

```
    def inner(x: float):
        return N( Subs(g4, "x", x))

    return inner
```