

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

Курсовая Работа

по дисциплине
«Математические модели»

Выполнил:
Группа:

Ферапонтов М.В.
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	8
2.8	Правый-верхний угол	8
2.9	Правый-нижний угол	9
3	Невязка разностной схемы	10
3.1	Невязка во внутренних точках	10
3.2	Невязка на левой границе	12
3.3	Невязка на правой границе	14
4	Запись СЛАУ	16
4.1	Запись для внутренних точек	16
4.2	Запись для левой границы	17
4.3	Запись для правой границы	18
4.4	Запись для нижней границы	18
4.5	Запись для верхней границы	19
4.6	Запись для левой нижней граничной точки	20
4.7	Запись для правой нижней граничной точки	20
5	Метод сопряжённых градиентов	24
5.1	Явный метод	24
5.2	Неявный метод	24
6	Тестирование	26
7	Заключение	30
	Код программы	31

1 Вступление

1.1 Постановка задачи

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

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

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

С граничными условиями:

$$\begin{aligned} u|_{r=0} & - \text{ограничено} & -k_1 \frac{\partial u}{\partial r} \Big|_{r=R} & = \chi_2 u|_{r=R} - \varphi_2(z) \\ k_2 \frac{\partial u}{\partial z} \Big|_{z=0} & = \chi_3 u|_{z=0} - \varphi_3(r) & u|_{z=L} & = \varphi_4(r) \\ \chi_2 & \geq 0 & \chi_3 & \geq 0 \end{aligned}$$

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:

N_r — число разбиений на $[0, R]$

$$r_0 < r_1 < \dots < r_N$$

$$r_0 = 0, \quad r_N = R$$

$$h_r = \frac{R - 0}{N_r}$$

N_z — число разбиений на $[0, L]$

$$z_0 < z_1 < \dots < z_N$$

$$z_0 = 0, \quad z_N = L$$

$$h_z = \frac{L - 0}{N_z}$$

Введем дополнительную сетку:

$$r_{i-\frac{1}{2}} = \frac{r_i + r_{i-1}}{2} \quad i = 1, \dots, N_r$$

$$h_i = \begin{cases} \frac{h_r}{2}, & i = 0 \\ h_r, & i = 1, 2, \dots, N_r - 1 \\ \frac{h_r}{2}, & i = N_r \end{cases}$$

$$z_{j-\frac{1}{2}} = \frac{z_j + z_{j-1}}{2} \quad j = 1, \dots, N_z$$

$$h_j = \begin{cases} \frac{h_z}{2}, & j = 0 \\ h_z, & j = 1, 2, \dots, N_z - 1 \\ \frac{h_z}{2}, & j = N_z \end{cases}$$

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 внутри интервала:

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

Получим:

$$\begin{aligned} & - \left[\int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r, z) \frac{\partial u}{\partial r} \Big|_{r=r_{i+\frac{1}{2}}} dz - \int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rk_1(r, z) \frac{\partial u}{\partial r} \Big|_{r=r_{i-\frac{1}{2}}} dz \right. \\ & \left. + \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r, z) \frac{\partial u}{\partial z} \Big|_{z=z_{j+\frac{1}{2}}} dr - \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r, z) \frac{\partial u}{\partial z} \Big|_{z=z_{j-\frac{1}{2}}} dr \right] = \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} rf(r, z) dr dz \end{aligned}$$

Воспользуемся формулами численного дифференцирования:

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

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

Также воспользуемся формулой средних прямоугольников:

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

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

В итоге получаем разностную схему внутри интервала:

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

Перейдём к частному:

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

Теперь найдем значение разностной схемы на углах и границах интервалов

2.2 На левой границе

Проинтегрируем наше уравнение в $i = 0$ и z внутри промежутка

$$- \int_{r_i}^{r_{i+\frac{1}{2}}} \int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} \left[\frac{\partial}{\partial r} \left(r k_1(r, z) \frac{\partial u}{\partial r} \right) + \frac{\partial}{\partial z} \left(r k_2(r, z) \frac{\partial u}{\partial z} \right) \right] dr dz = \int_{r_i}^{r_{i+\frac{1}{2}}} \int_{z_{j-\frac{1}{2}}}^{z_{j+\frac{1}{2}}} r f(r, z) dr dz$$

Получаем:

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

Имеем граничное условие:

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

$$\int_{r_i}^{r_{i+\frac{1}{2}}} rf dr \approx f_i \int_{r_i}^{r_{i+\frac{1}{2}}} r dr = f_i \frac{r_{i+\frac{1}{2}}^2}{2} = \frac{h_r}{2} f_i \frac{r_{i+\frac{1}{2}}}{2}, \quad i = 0, \quad r_i = 0, r_{i+\frac{1}{2}} = \frac{h_r}{2}$$

Получаем разностную схему:

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

Перейдём к частному:

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

2.3 На правой границе

Проинтегрируем наше уравнение в $i = N_x$ и z внутри промежутка

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

Получаем:

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

Имеем граничное условие:

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

Получаем разностную схему:

$$\begin{aligned}
& - \left[-\hbar_j r_i (\chi_2 v_i - \varphi_2(z)) - \hbar_j r_{i-\frac{1}{2}} k_1(r_{i-\frac{1}{2}}, z_j) \frac{v_{i,j} - v_{i-1,j}}{h_r} \right. \\
& \left. + \hbar_i r_i k_2(r_i, z_{j+\frac{1}{2}}) \frac{v_{i,j+1} - v_{i,j}}{h_z} - \hbar_i r_i k_2(r_i, z_{j-\frac{1}{2}}) \frac{v_{i,j} - v_{i,j-1}}{h_z} \right] = \hbar_i \hbar_j r_i f_{i,j}
\end{aligned}$$

Перейдём к частному:

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

2.4 На нижней границе

Проинтегрируем наше уравнение $j = 0$ и i внутри промежутка

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

Получаем:

$$\begin{aligned}
& - \left[\int_{z_j}^{z_{j+\frac{1}{2}}} rk_1(r, z) \frac{\partial u}{\partial r} \Big|_{r=r_{i+\frac{1}{2}}} dz - \int_{z_j}^{z_{j+\frac{1}{2}}} rk_1(r, z) \frac{\partial u}{\partial r} \Big|_{r=r_{i-\frac{1}{2}}} dz \right. \\
& \left. + \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r, z) \frac{\partial u}{\partial z} \Big|_{z=z_{j+\frac{1}{2}}} dr - \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} rk_2(r, z) \frac{\partial u}{\partial z} \Big|_{z=z_j} dr \right] = \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \int_{z_j}^{z_{j+\frac{1}{2}}} rf(r, z) dr dz
\end{aligned}$$

Имеем граничное условие:

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

Получаем разностную схему:

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

Перейдём к частному:

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

2.5 На верхней границе

Имеем граничное условие:

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

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

2.6 Левый-нижний угол

Проинтегрируем наше уравнение в $i = 0$ и $j = 0$ внутри промежутка

$$- \int_{r_i}^{r_{i+\frac{1}{2}}} \int_{z_j}^{z_{j+\frac{1}{2}}} \left[\frac{\partial}{\partial r} \left(r k_1(r, z) \frac{\partial u}{\partial r} \right) + \frac{\partial}{\partial z} \left(r k_2(r, z) \frac{\partial u}{\partial z} \right) \right] dr dz = \int_{r_i}^{r_{i+\frac{1}{2}}} \int_{z_j}^{z_{j+\frac{1}{2}}} r f(r, z) dr dz$$

Получаем:

$$\begin{aligned} & - \left[\int_{z_j}^{z_{j+\frac{1}{2}}} r k_1(r, z) \frac{\partial u}{\partial r} \Big|_{r=r_{i+\frac{1}{2}}} dz - \int_{z_j}^{z_{j+\frac{1}{2}}} r k_1(r, z) \frac{\partial u}{\partial r} \Big|_{r=r_i} dz \right. \\ & \left. + \int_{r_i}^{r_{i+\frac{1}{2}}} r k_2(r, z) \frac{\partial u}{\partial z} \Big|_{z=z_{j+\frac{1}{2}}} dr - \int_{r_i}^{r_{i+\frac{1}{2}}} r k_2(r, z) \frac{\partial u}{\partial z} \Big|_{z=z_j} dr \right] = \int_{r_i}^{r_{i+\frac{1}{2}}} \int_{z_j}^{z_{j+\frac{1}{2}}} r f(r, z) dr dz \end{aligned}$$

Имеем граничное условие:

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

$$\int_{r_i}^{r_{i+\frac{1}{2}}} r f dr \approx f_i \int_{r_i}^{r_{i+\frac{1}{2}}} r dr = f_i \frac{r_{i+\frac{1}{2}}^2}{2} = h_r f_i \frac{r_{i+\frac{1}{2}}}{2}, \quad i = 0, \quad r_i = 0, r_{i+\frac{1}{2}} = \frac{h_r}{2}$$

Также:

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

Получаем:

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

Перейдём к частному:

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

2.7 Левый-верхний угол

При $i = 0$ и $z = N_z$ имеем граничное условие:

Имеем граничное условие:

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

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

2.8 Правый-верхний угол

При $i = N_r$ и $z = N_z$ имеем граничное условие:

Имеем граничное условие:

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

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

2.9 Правый-нижний угол

Проинтегрируем наше уравнение в $i = N_r$ и $j = 0$:

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

Получаем:

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

Имеем граничные условия:

$$\begin{aligned} -k_1 \frac{\partial u}{\partial r} \Big|_{r=R} &= \chi_2 u|_{r=R} - \varphi_2(z) \\ k_2 \frac{\partial u}{\partial z} \Big|_{z=0} &= \chi_3 u|_{z=0} - \varphi_3(r) \end{aligned}$$

Получаем разностную схему:

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

Перейдём к частному

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

3 Невязка разностной схемы

3.1 Невязка во внутренних точках

Запишем для уравнения разностной сетки во внутренних точках невязку:

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

Напишем разложение Тейлора для невязки:

$$\begin{aligned}
 u_{i,j-1} &= u(r_i, z_j - h_z) = \left[u - h_z \frac{\partial u}{\partial z} + \frac{h_z^2}{2} \frac{\partial^2 u}{\partial z^2} - \frac{h_z^3}{6} \frac{\partial^3 u}{\partial z^3} + \frac{h_z^4}{24} \frac{\partial^4 u}{\partial z^4} \right]_{i,j} + \mathcal{O}(h_y^5) \\
 \frac{u_{i,j} - u_{i,j-1}}{h_z} &= \left[\frac{\partial u}{\partial z} - \frac{h_z}{2} \frac{\partial^2 u}{\partial z^2} + \frac{h_z^2}{6} \frac{\partial^3 u}{\partial z^3} - \frac{h_z^3}{24} \frac{\partial^4 u}{\partial z^4} \right]_{i,j} + \mathcal{O}(h_y^4) \\
 r_i k_2(r_i, z_{j-\frac{1}{2}}) &= r_i k_2(r_i, z_j - \frac{h_z}{2}) = \left[r k_2 - \frac{h_z}{2} \frac{\partial r k_2}{\partial z} + \frac{h_z^2}{8} \frac{\partial^2 r k_2}{\partial z^2} - \frac{h_z^3}{48} \frac{\partial^3 r k_2}{\partial z^3} \right]_{i,j} + \mathcal{O}(h^4) \\
 r_i k_2(r_i, z_{j-\frac{1}{2}}) \frac{u_{i,j} - u_{i,j-1}}{h_z} &= \left[r k_2 \frac{\partial u}{\partial z} \right]_{i,j} - h_z \left[\frac{1}{2} r k_2 \frac{\partial^2 u}{\partial z^2} + \frac{1}{2} \frac{\partial r k_2}{\partial z} \frac{\partial u}{\partial z} \right]_{i,j} + \\
 &+ h_z^2 \left[\frac{1}{6} r k_2 \frac{\partial^3 u}{\partial z^3} + \frac{1}{4} \frac{\partial r k_2}{\partial z} \frac{\partial^2 u}{\partial z^2} + \frac{1}{8} \frac{\partial^2 r k_2}{\partial z^2} \frac{\partial u}{\partial z} \right]_{i,j} - \\
 &- h_z^3 \left[\frac{1}{24} r k_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{12} \frac{\partial r k_2}{\partial z} \frac{\partial^3 u}{\partial z^3} + \frac{1}{16} \frac{\partial^2 r k_2}{\partial z^2} \frac{\partial^2 u}{\partial z^2} + \frac{1}{48} \frac{\partial^3 r k_2}{\partial z^3} \frac{\partial u}{\partial z} \right]_{i,j} + \\
 &+ \mathcal{O}(h^4)
 \end{aligned}$$

Для $r_i k_2(r_i, z_{j+\frac{1}{2}})$ можно получить невязку аналогичным способом:

$$\begin{aligned}
 r_i k_2(r_i, z_{j+\frac{1}{2}}) \frac{u_{i,j+1} - u_{i,j}}{h_z} &= \left[r k_2 \frac{\partial u}{\partial z} \right]_{i,j} + h_z \left[\frac{1}{2} r k_2 \frac{\partial^2 u}{\partial z^2} + \frac{1}{2} \frac{\partial r k_2}{\partial z} \frac{\partial u}{\partial z} \right]_{i,j} + \\
 &+ h_z^2 \left[\frac{1}{6} r k_2 \frac{\partial^3 u}{\partial z^3} + \frac{1}{4} \frac{\partial r k_2}{\partial z} \frac{\partial^2 u}{\partial z^2} + \frac{1}{8} \frac{\partial^2 r k_2}{\partial z^2} \frac{\partial u}{\partial z} \right]_{i,j} + \\
 &+ h_z^3 \left[\frac{1}{24} r k_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{12} \frac{\partial r k_2}{\partial z} \frac{\partial^3 u}{\partial z^3} + \frac{1}{16} \frac{\partial^2 r k_2}{\partial z^2} \frac{\partial^2 u}{\partial z^2} + \frac{1}{48} \frac{\partial^3 r k_2}{\partial z^3} \frac{\partial u}{\partial z} \right]_{i,j} + \\
 &+ \mathcal{O}(h^4)
 \end{aligned}$$

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

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

$$r_i k_1(r_{i-\frac{1}{2}}, z_j) = r_i k_1(r_i - \frac{h_r}{2}, z_j) = \left[r k_1 - \frac{h_r}{2} \frac{\partial r k_1}{\partial r} + \frac{h_r^2}{8} \frac{\partial^2 r k_1}{\partial r^2} - \frac{h_r^3}{48} \frac{\partial^3 r k_1}{\partial r^3} \right]_{i,j} + \mathcal{O}(h^4)$$

$$\begin{aligned} r_i k_1(r_{i-\frac{1}{2}}, z_j) \frac{u_{i,j} - u_{i-1,j}}{h_r} &= \left[r k_1 \frac{\partial u}{\partial r} \right]_{i,j} - h_r \left[\frac{1}{2} r k_1 \frac{\partial^2 u}{\partial r^2} + \frac{1}{2} \frac{\partial r k_1}{\partial r} \frac{\partial u}{\partial r} \right]_{i,j} + \\ &+ h_r^2 \left[\frac{1}{6} r k_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4} \frac{\partial r k_1}{\partial r} \frac{\partial^2 u}{\partial r^2} + \frac{1}{8} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial u}{\partial r} \right]_{i,j} - \\ &- h_r^3 \left[\frac{1}{24} r k_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{12} \frac{\partial r k_1}{\partial r} \frac{\partial^3 u}{\partial r^3} + \frac{1}{16} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial^2 u}{\partial r^2} + \frac{1}{48} \frac{\partial^3 r k_1}{\partial r^3} \frac{\partial u}{\partial r} \right]_{i,j} + \\ &+ \mathcal{O}(h^4) \end{aligned}$$

Для $r_i k_1(r_{i=\frac{1}{2}}, z_j) \frac{u_{i+1,j} - u_{i,j}}{h_r}$ можно получить невязку аналогично:

$$\begin{aligned} r_i k_1(r_{i+\frac{1}{2}}, z_j) \frac{u_{i+1,j} - u_{i,j}}{h_r} &= \left[r k_1 \frac{\partial u}{\partial r} \right]_{i,j} + h_r \left[\frac{1}{2} r k_1 \frac{\partial^2 u}{\partial r^2} + \frac{1}{2} \frac{\partial r k_1}{\partial r} \frac{\partial u}{\partial r} \right]_{i,j} + \\ &+ h_r^2 \left[\frac{1}{6} r k_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4} \frac{\partial r k_1}{\partial r} \frac{\partial^2 u}{\partial r^2} + \frac{1}{8} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial u}{\partial r} \right]_{i,j} + \\ &+ h_r^3 \left[\frac{1}{24} r k_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{12} \frac{\partial r k_1}{\partial r} \frac{\partial^3 u}{\partial r^3} + \frac{1}{16} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial^2 u}{\partial r^2} + \frac{1}{48} \frac{\partial^3 r k_1}{\partial r^3} \frac{\partial u}{\partial r} \right]_{i,j} + \\ &+ \mathcal{O}(h^4) \end{aligned}$$

Тогда:

$$\begin{aligned} r_i k_1(r_{i+\frac{1}{2}}, z_j) \frac{u_{i+1,j} - u_{i,j}}{h_r} - r_i k_1(r_{i-\frac{1}{2}}, z_j) \frac{u_{i,j} - u_{i-1,j}}{h_r} &= h_r h_z \left(\left[r k_1 \frac{\partial^2 u}{\partial r^2} + \frac{\partial r k_1}{\partial r} \frac{\partial u}{\partial r} \right]_{i,j} + \right. \\ &+ h_r^2 \left[\frac{1}{12} r k_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{6} \frac{\partial r k_1}{\partial r} \frac{\partial^3 u}{\partial r^3} + \frac{1}{8} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial^2 u}{\partial r^2} + \frac{1}{24} \frac{\partial^3 r k_1}{\partial r^3} \frac{\partial u}{\partial r} \right] + \mathcal{O}(h_r^4) \Big) \end{aligned}$$

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

Будем искать невязку в следующем виде:

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

Тогда:

$$\begin{aligned} \tilde{\xi}_{i,j} = & r_i f_{i,j} + \left[rk_1 \frac{\partial^2 u}{\partial r^2} + \frac{\partial rk_1}{\partial r} \frac{\partial u}{\partial r} \right]_{i,j} \left[rk_2 \frac{\partial^2 u}{\partial z^2} + \frac{\partial rk_2}{\partial z} \frac{\partial u}{\partial z} \right]_{i,j} + \\ & + h_r^2 \left[\frac{1}{12} rk_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{6} \frac{\partial rk_1}{\partial r} \frac{\partial^3 u}{\partial r^3} + \frac{1}{8} \frac{\partial^2 rk_1}{\partial r^2} \frac{\partial^2 u}{\partial r^2} + \frac{1}{24} \frac{\partial^3 rk_1}{\partial r^3} \frac{\partial u}{\partial r} \right] + \mathcal{O}(h_r^3) + \\ & + h_z^2 \left[\frac{1}{12} rk_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{6} \frac{\partial rk_2}{\partial z} \frac{\partial^3 u}{\partial z^3} + \frac{1}{8} \frac{\partial^2 rk_2}{\partial z^2} \frac{\partial^2 u}{\partial z^2} + \frac{1}{24} \frac{\partial^3 rk_2}{\partial z^3} \frac{\partial u}{\partial z} \right]_{i,j} + \mathcal{O}(h_z^3) \end{aligned}$$

Можно заметить, что:

$$\begin{aligned} \left[rk_1 \frac{\partial^2 u}{\partial r^2} + \frac{\partial rk_1}{\partial r} \frac{\partial u}{\partial r} \right]_{i,j} &= \frac{\partial}{\partial r} \left(k_1 \frac{\partial u}{\partial r} \right) \\ \left[rk_2 \frac{\partial^2 u}{\partial z^2} + \frac{\partial rk_2}{\partial z} \frac{\partial u}{\partial z} \right]_{i,j} &= \frac{\partial}{\partial z} \left(k_2 \frac{\partial u}{\partial z} \right) \end{aligned}$$

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:

$$\begin{aligned} \tilde{\xi}_{i,j} = & h_r^2 \left[\frac{1}{12} rk_1 \frac{\partial^4 u}{\partial r^4} - \frac{1}{6} \frac{\partial rk_1}{\partial r} \frac{\partial^3 u}{\partial r^3} + \frac{1}{8} \frac{\partial^2 rk_1}{\partial r^2} \frac{\partial^2 u}{\partial r^2} + \frac{1}{24} \frac{\partial^3 rk_1}{\partial r^3} \frac{\partial u}{\partial r} \right] + \mathcal{O}(h_r^3) + \\ & + h_z^2 \left[\frac{1}{12} rk_2 \frac{\partial^4 u}{\partial z^4} - \frac{1}{6} \frac{\partial rk_2}{\partial z} \frac{\partial^3 u}{\partial z^3} + \frac{1}{8} \frac{\partial^2 rk_2}{\partial z^2} \frac{\partial^2 u}{\partial z^2} + \frac{1}{24} \frac{\partial^3 rk_2}{\partial z^3} \frac{\partial u}{\partial z} \right]_{i,j} + \mathcal{O}(h_z^3) \end{aligned}$$

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

3.2 Невязка на левой границе

Запишем для уравнения разностной сетки на левой границе невязку:

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

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

Напишем разложение Тейлора для невязки:

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

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

$$h_r k_2(r_0, z_{j+\frac{1}{2}}) \frac{u_{0,j+1} - u_{0,j}}{h_z} - h_r k_2(r_0, z_{j-\frac{1}{2}}) \frac{u_{0,j} - u_{0,j-1}}{h_z} = h_r \left[h_z \left(\frac{\partial}{\partial z} \left(k_2 \frac{\partial u}{\partial z} \right) \right)_{i,j} + \right.$$

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

$$\xi_{0,j} = \frac{h_r}{2} h_z f_{0,j} + \left[2h_z \left(\left[k_1 \frac{\partial u}{\partial r} \right]_{i,j} + \frac{h_r}{2} \left[\frac{\partial}{\partial r} \left(k_1 \frac{\partial u}{\partial r} \right) \right]_{i,j} + \right.$$

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

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

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:

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

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

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

3.3 Невязка на правой границе

Запишем для уравнения разностной сетки на правой границе невязку:

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

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

Напишем разложение Тейлора для невязки:

$$r_{N_r-\frac{1}{2}} k_1(r_{N_r-\frac{1}{2}}, z_j) \frac{u_{N_r,j} - u_{N_r-1,j}}{h_r} = \left[r k_1 \frac{\partial u}{\partial r} \right]_{N_r,j} - \frac{h_2}{2} \left[\frac{\partial}{\partial r} \left(r k \frac{\partial u}{\partial r} \right) \right]_{N_r,j} + h_r^2 \left[\frac{1}{6} r k_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4} \frac{\partial r k_1}{\partial r} \frac{\partial^2 u}{\partial r^2} + \frac{1}{8} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial u}{\partial r} \right]_{N_r,j} - h_r^3 \left[\frac{1}{24} r k_1 \frac{\partial^4 u}{\partial r^4} + \frac{1}{12} \frac{\partial r k_1}{\partial r} \frac{\partial^3 u}{\partial r^3} + \frac{1}{16} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial^2 u}{\partial r^2} + \frac{1}{48} \frac{\partial^3 r k_1}{\partial r^3} \frac{\partial u}{\partial r} \right]_{N_r,j} + \mathcal{O}(h_r^4)$$

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

$$\begin{aligned} \xi_{N_r,j} = & \frac{h_r}{2} r_{N_r} h_z f_{N_r,j} - h_z(\chi_2 u_{N_r} - \varphi_2(z)) - h_z \left[\left[r k_1 \frac{\partial u}{\partial r} \right]_{N_r,j} - \frac{h_2}{2} \left[\frac{\partial}{\partial r} \left(r k \frac{\partial u}{\partial r} \right) \right]_{N_r,j} + \right. \\ & + h_r^2 \left[\frac{1}{6} r k_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4} \frac{\partial r k_1}{\partial r} \frac{\partial^2 u}{\partial r^2} + \frac{1}{8} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial u}{\partial r} \right]_{N_r,j} - \\ & - h_r^3 \left[\frac{1}{24} r k_1 \frac{\partial^4 u}{\partial r^4} + \frac{1}{12} \frac{\partial r k_1}{\partial r} \frac{\partial^3 u}{\partial r^3} + \frac{1}{16} \frac{\partial^2 r k_1}{\partial r^2} \frac{\partial^2 u}{\partial r^2} + \frac{1}{48} \frac{\partial^3 r k_1}{\partial r^3} \frac{\partial u}{\partial r} \right]_{N_r,j} + \mathcal{O}(h_r^4) \Big] + \\ & + \frac{h_r}{2} \left[h_z \left(\frac{\partial}{\partial z} \left(r k_2 \frac{\partial u}{\partial z} \right) \right)_{N_r,j} + h_z^3 \left(\frac{1}{12} r k_2 \frac{\partial^4 u}{\partial z^4} + \frac{1}{6} \frac{\partial r k_2}{\partial z} \frac{\partial^3 u}{\partial z^3} + \frac{1}{8} \frac{\partial^2 r k_2}{\partial z^2} \frac{\partial^2 u}{\partial z^2} + \frac{1}{24} \frac{\partial^3 r k_2}{\partial z^3} \frac{\partial u}{\partial z} \right)_{N_r,j} \right. \\ & \left. + \mathcal{O}(h_z^4) \right] \end{aligned}$$

Будем искать невязку в следующем виде:

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

$$\begin{aligned} \tilde{\xi}_{N_r,j} = & \frac{h_r}{2} \left[f + \frac{\partial}{\partial r} \left(rk_1 \frac{\partial u}{\partial r} \right) + \frac{\partial}{\partial z} \left(rk_2 \frac{\partial u}{\partial z} \right) \right]_{N_r,j} - \left[rk_1 \frac{\partial u}{\partial r} + (\chi_2 u - \varphi(z)) \right]_{N_r,j} - \\ & - h_r^2 \left[\frac{1}{6} rk_1 \frac{\partial^3 u}{\partial r^3} + \frac{1}{4} \frac{\partial rk_1}{\partial r} \frac{\partial^2 u}{\partial r^2} + \frac{1}{8} \frac{\partial^2 rk_1}{\partial r^2} \frac{\partial u}{\partial r} \right]_{N_r,j} + \mathcal{O}(h_r^3) + \\ & + \frac{h_r}{2} \left[h_z^2 \left(\frac{1}{12} rk_2 \frac{\partial^4 u}{\partial z^4} + \frac{1}{6} \frac{\partial rk_2}{\partial z} \frac{\partial^3 u}{\partial z^3} + \frac{1}{8} \frac{\partial^2 rk_2}{\partial z^2} \frac{\partial^2 u}{\partial z^2} + \frac{1}{24} \frac{\partial^3 rk_2}{\partial z^3} \frac{\partial u}{\partial z} \right) \right]_{N_r,j} + \mathcal{O}(h_z^3) \end{aligned}$$

Заметим, что:

$$\begin{aligned} \left[rk_1 \frac{\partial u}{\partial r} + (\chi_2 u - \varphi(z)) \right]_{N_r,j} &= 0 \\ \left[f + \frac{\partial}{\partial r} \left(rk_1 \frac{\partial u}{\partial r} \right) + \frac{\partial}{\partial z} \left(rk_2 \frac{\partial u}{\partial z} \right) \right]_{N_r,j} &= 0 \end{aligned}$$

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:

$$0 \leq i \leq N_r$$

$$0 \leq j \leq N_z$$

Тогда имеем:

$$0 \leq m < (N_r + 1)(N_z + 1)$$

4.1 Запись для внутренних точек

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

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

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 $j \in (0, N_z)$:

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

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 $j \in (0, N_z)$:

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

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 $j = 0$:

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

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

Введём новые обозначения:

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

где коэффициенты равны:

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

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

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

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

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

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:

$$c_m w_m = \varphi_m$$

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 $j = 0$:

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

Введём новые обозначения:

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

где коэффициенты ранвы:

$$\begin{aligned}
c_m &= \frac{h_z}{2h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_0) + \frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{\frac{1}{2}}) + \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} \chi_3 \\
d_m &= -\frac{h_z}{2h_r} r_{\frac{1}{2}} k_1(r_{\frac{1}{2}}, z_0) \\
e_m &= -\frac{h_r}{2h_z} \frac{r_{\frac{1}{2}}}{2} k_2(r_0, z_{\frac{1}{2}}) \\
g_m &= \frac{h_r}{2} \frac{h_z}{2} \frac{r_{\frac{1}{2}}}{2} f_{0,0} + \frac{h_r}{2} \frac{r_{\frac{1}{2}}}{2} \varphi_3(r_0)
\end{aligned}$$

4.7 Запись для правой нижней граничной точки

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

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

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

Введём новые обозначения:

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

где коэффициенты равны:

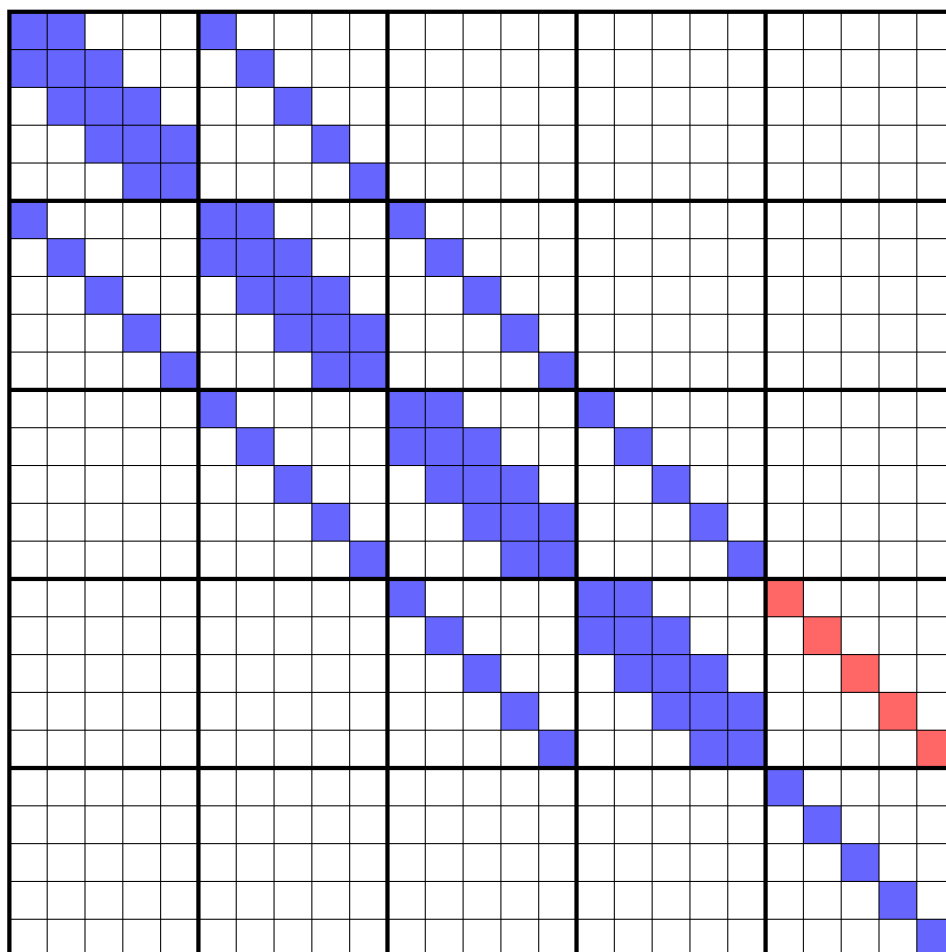
$$b_m = -\frac{h_z}{2h_r}r_{N_r-\frac{1}{2}}k_1(r_{N_r-\frac{1}{2}}, z_0)$$

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

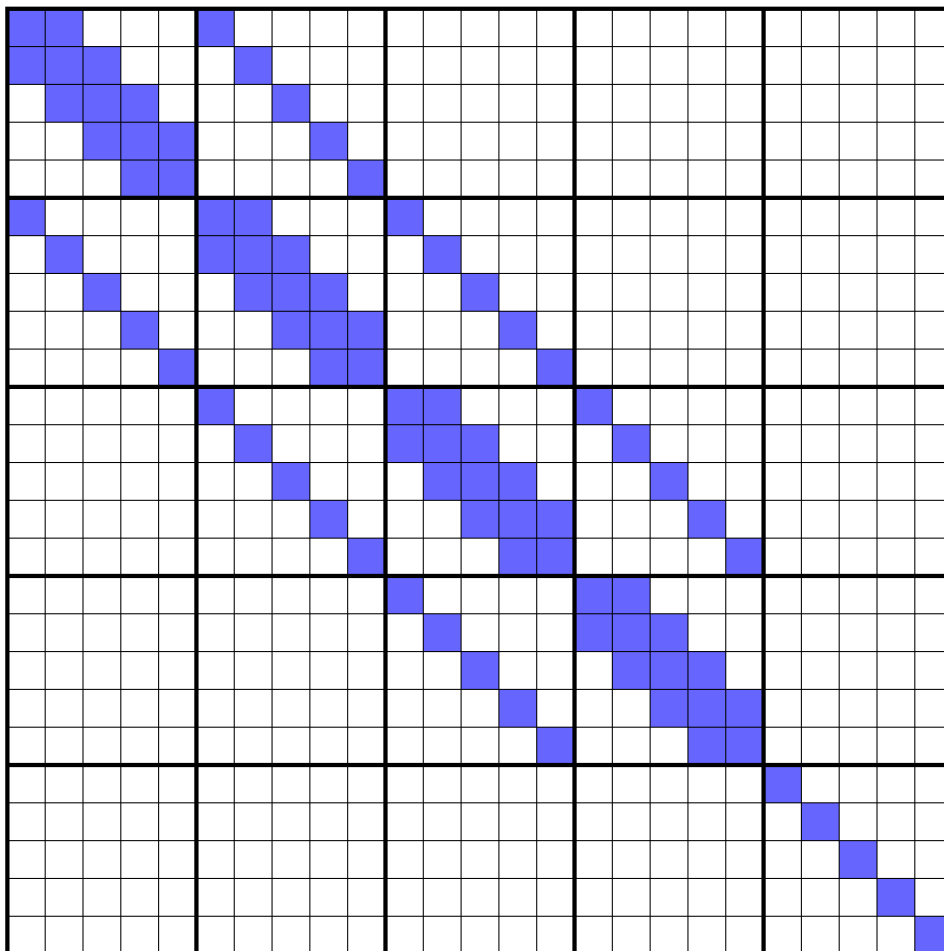
$$e_m = -\frac{h_r}{2h_z}r_{N_r}k_2(r_{N_r}, z_{\frac{1}{2}})$$

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

Получаем следующую систему:



Уберем лишние элементы и получим систему с симметричной матрицей:



Данную систему мы будем решать методом сопряжённых градиентов.

5 Метод сопряжённых градиентов

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

$$Ax = b$$

где A - симметричная, положительно определённая, матрица.

5.1 Явный метод

```
1  $r^{(0)} = b - Ax^{(0)}$ 
2  $s^{(1)} = r^{(0)}$ 
3  $\gamma = \sqrt{(b, b)}$ 
4 for  $k = 1$  to  $k_{max}$  do
5      $\alpha_k = \frac{(r^{(k-1)}, r^{(k-1)})}{(As^{(k)}, s^{(k)})}$ 
6      $x^{(k)} = x^{(k-1)} + \alpha_k s^{(k)}$ 
7      $r^{(k)} = r^{(k-1)} - \alpha_k s^{(k-1)}$ 
8     if  $\sqrt{(r^{(k)}, r^{(k)})} < \gamma\varepsilon$  then
9         break
10    end
11     $\beta_k = \frac{(r^{(k)}, r^{(k)})}{(r^{(k-1)}, r^{(k-1)})}$ 
12     $s^{(k+1)} = r^{(k)} + \beta_k s^{(k)}$ 
13 end
```

5.2 Неявный метод

Неявный метод основывается на использовании предобуславливания. Идея заключается в том, чтобы выбрать матрицу B , которая является симметричной и положительно определённой, 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.

```

1   $r^{(0)} = b - Ax^{(0)}$ 
2   $Bw^{(0)} = r^{(0)}$ 
3   $s^{(1)} = w^{(0)}$ 
4   $Bg = b$ 
5   $\gamma = \sqrt{(g, b)}$ 
6  for  $k = 1$  to  $k_{max}$  do
7       $\alpha_k = \frac{(w^{(k-1)}, r^{(k-1)})}{(As^{(k)}, s^{(k)})}$ 
8       $x^{(k)} = x^{(k-1)} + \alpha_k s^{(k)}$ 
9       $r^{(k)} = r^{(k-1)} - \alpha_k s^{(k-1)}$ 
10      $Bw^{(k)} = r^{(k)}$ 
11     if  $\sqrt{(w^{(k)}, r^{(k)})} < \gamma\varepsilon$  then
12         break
13     end
14      $\beta_k = \frac{(w^{(k)}, r^{(k)})}{(w^{(k-1)}, r^{(k-1)})}$ 
15      $s^{(k+1)} = w^{(k)} + \beta_k s^{(k)}$ 
16 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 построенную модель на следующих тестовых наборах:

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

Все тесты имеют общие значения:

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

$$\xi_2 = 2, \quad \xi_3 = 3$$

Погрешность решения задачи вычисляется следующим образом:

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

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

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

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

Тест №1

Имеем лишь ошибку округления.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	8	$4.324e - 16$	$7.63e - 16$	$4.064e - 16$
4	4	27	$2.514e - 11$	$2.419e - 11$	$2.88e - 11$
8	8	62	$1.354e - 10$	$1.284e - 10$	$1.868e - 10$
16	16	117	$2.3e - 10$	$6.527e - 10$	$1.151e - 10$
32	32	306	$5.921e - 09$	$1.849e - 09$	$4.102e - 09$

Таблица 1: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	5	$9.509e-16$	$4.186e-16$	$9.29e-16$
4	4	9	$2.395e-11$	$2.574e-11$	$4.137e-11$
8	8	15	$1.62e-10$	$2.34e-10$	$1.97e-10$
16	16	25	$6.74e-10$	$9.82e-10$	$2.23e-10$
32	32	46	$1.12e-9$	$1.59e-9$	$3.37e-9$

Таблица 2: Неявный метод

Тест №2

Присутствует ошибка округления и ошибка аппроксимации.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	8	$3.471e-4$	$4.164e-4$	$7.953e-4$
4	4	21	$8.689e-5$	$1.041e-4$	$1.986e-4$
8	8	55	$2.181e-5$	$2.624e-5$	$4.971e-5$
16	16	107	$5.378e-6$	$6.47e-6$	$1.249e-5$
32	32	208	$1.485e-6$	$1.589e-6$	$3.136e-6$

Таблица 3: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	5	$3.471e-4$	$4.164e-4$	$7.953e-4$
4	4	8	$8.689e-5$	$1.041e-4$	$1.986e-4$
8	8	14	$2.181e-5$	$2.624e-5$	$4.971e-5$
16	16	27	$5.378e-6$	$6.47e-6$	$1.249e-5$
32	32	44	$1.485e-6$	$1.589e-6$	$3.136e-6$

Таблица 4: Неявный метод

Тест №3

Присутствует ошибка округления и ошибка аппроксимации.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	7	$6.74e-2$	$7.374e-2$	$6.27e-2$
4	4	22	$1.989e-2$	$1.842e-2$	$1.566e-2$
8	8	41	$4.975e-3$	$4.587e-3$	$3.918e-3$
16	16	163	$1.214e-3$	$1.11e-3$	$9.937e-4$
32	32	269	$3.256e-4$	$2.976e-4$	$2.232e-4$

Таблица 5: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	5	$6.74e-2$	$7.374e-2$	$6.27e-2$
4	4	8	$1.989e-2$	$1.842e-2$	$1.566e-2$
8	8	15	$4.975e-3$	$4.587e-3$	$3.918e-3$
16	16	25	$1.214e-3$	$1.11e-3$	$9.937e-4$
32	32	47	$3.256e-4$	$2.976e-4$	$2.232e-4$

Таблица 6: *
Неявный метод

Тест №4

Присутствует ошибка округления и ошибка аппроксимации.

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	7	$1.074e-1$	$9.374e-2$	$1.28e-1$
4	4	22	$2.7029e-2$	$2.4e-3$	$3.176e-2$
8	8	41	$6.64e-3$	$6.036e-3$	$8.085e-3$
16	16	107	$1.568e-3$	$1.491e-3$	$1.78e-3$
32	32	246	$4.755e-4$	$3.628e-4$	$5.098e-4$

Таблица 7: Явный метод

N_r	N_z	n	δ_1	δ_2	δ_3
2	2	4	$1.074e-1$	$9.374e-2$	$1.28e-1$
4	4	7	$2.7029e-2$	$2.4e-3$	$3.176e-2$
8	8	17	$6.64e-3$	$6.036e-3$	$8.085e-3$
16	16	29	$1.568e-3$	$1.491e-3$	$1.78e-3$
32	32	51	$4.755e-4$	$3.628e-4$	$5.098e-4$

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. Приведены результаты экспериментального исследования точности программного обеспечения.

Код программы

grid.py

```
1  from typing import Tuple, Callable
2
3
4  def get_x(a: float, b: float, Nx: int) -> Tuple[Callable[[int], float], Callable[[int],
    ↪ float]]:
5      hx = (b - a) / Nx
6
7      def inner1(i: int) -> float:
8          """returns  $x_{\{i\}}$ """
9          if i < 0 or i > Nx:
10             raise KeyError()
11
12             return a + i * hx
13
14     def inner2(i: int) -> float:
15         """returns  $x_{\{i+1/2\}}$ """
16         if i < 0 or i >= Nx:
17             raise KeyError()
18
19         return a + (i + 0.5) * hx
20
21     return (inner1, inner2)
22
23
24 def get_y(c: float, d: float, Ny: int) -> Tuple[Callable[[int], float], Callable[[int],
    ↪ float]]:
25     hy = (d - c) / Ny
26
27     def inner1(j: int) -> float:
28         """returns  $y_{\{j\}}$ """
29         if j < 0 or j > Ny:
30             raise KeyError()
31
32         return c + j * hy
33
34     def inner2(j: int) -> float:
35         """returns  $y_{\{j+1/2\}}$ """
36         if j < 0 or j >= Ny:
37             raise KeyError()
38
39         return c + (j + 0.5) * hy
40
41     return (inner1, inner2)
```



```
1  import math
2  from typing import List
3  from sparse_matrix import *
4
5
6  def is_symmetric(m: SparseMatrix):
7      sz = m.size()
8
9      for i in range(sz):
10         for j in range(sz):
11             if m.get(i, j) != m.get(j, i):
12                 return False
13
14     return True
15
16
17  def add_vv(v1: List[float], v2: List[float]) -> List[float]:
18      sz = len(v1)
19      rez = [0] * sz
20
21      for i in range(sz):
22          rez[i] = v1[i] + v2[i]
23
24     return rez
25
26  def mul_sv(a: float, v: List[float]) -> List[float]:
27      sz = len(v)
28      rez = [0] * sz
29
30      for i in range(sz):
31          rez[i] = v[i] * a
32
33     return rez
34
35  def sub_vv(v1: List[float], v2: List[float]) -> List[float]:
36      return add_vv(v1, mul_sv(-1, v2))
37
38  def mul_vv(v1: List[float], v2: List[float]) -> float:
39      sz = len(v1)
40      rez = 0
41
42      for i in range(sz):
43          rez += v1[i] * v2[i]
44
45     return rez
46
```

```

47 def mul_mv(m: SparseMatrix, v: List[float]) -> List[float]:
48     sz = len(v)
49     rez = [0] * sz
50
51     for i in range(sz):
52         for j, d in m.get(i):
53             if i == j:
54                 continue
55             rez[i] += d * v[j]
56             if m.is_symmetric():
57                 rez[j] += d * v[i]
58
59     for i in range(sz):
60         rez[i] += m.get(i, i) * v[i]
61
62     return rez
63
64 def mul_mm(m1: SparseMatrix, m2: SparseMatrix) -> SparseMatrixCOO:
65     sz = m1.size()
66     rez = SparseMatrixCOO()
67
68     for i in range(sz):
69         for j in range(sz):
70             acc = 0
71             for k in range(sz):
72                 acc += m1.get(i, k) * m2.get(k, j)
73
74             rez.set(i, j, acc)
75
76     return rez
77
78 def transpose(m: SparseMatrix) -> SparseMatrixCOO:
79     sz = m.size()
80     rez = SparseMatrixCOO()
81
82     for i in range(sz):
83         for j, d in m.get(i):
84             rez.set(j, i, d)
85
86     return rez
87
88 def copy_v(v: List[float]) -> List[float]:
89     sz = len(v)
90     rez = [0] * sz
91
92     for i in range(sz):

```

```

93         rez[i] = v[i]
94
95     return rez
96
97 def _cholesky_solve(L: SparseMatrix, r: List[float]) -> List[float]:
98     sz = len(r)
99
100    LT = transpose(L)
101
102    z = [0] * sz
103    for i in range(sz):
104        acc = r[i]
105        for j, d in L.get(i):
106            if j < i:
107                acc -= d * z[j]
108
109        z[i] = acc / L.get(i, i)
110
111    w = [0] * sz
112    for i in range(sz - 1, -1, -1):
113        acc = z[i]
114        for j, d in LT.get(i):
115            if j > i:
116                acc -= d * w[j]
117
118        w[i] = acc / LT.get(i, i)
119
120    return w
121
122 def conjgrad(A: SparseMatrix, b: List[float], x: List[float],
123             eps: float, maxiter: int=10000, L: SparseMatrix=None) -> List[float]:
124     nsteps = 0
125
126     if L == None:
127         r = sub_vv(b, mul_mv(A, x))
128         p = copy_v(r)
129         gamma = math.sqrt( mul_vv(b, b) )
130
131         rsold = mul_vv(r, r)
132
133         while nsteps < maxiter:
134             nsteps += 1
135
136             Ap = mul_mv(A, p)
137             alpha = rsold / mul_vv(p, Ap)
138

```

```

139         x = add_vv(x, mul_sv(alpha, p))
140         r = sub_vv(r, mul_sv(alpha, Ap))
141
142         rsnew = mul_vv(r, r)
143         if math.sqrt(rsnew) < gamma * eps:
144             break
145
146         p = add_vv(r, mul_sv(rsnew / rsold, p))
147         rsold = rsnew
148     else:
149         r = sub_vv(b, mul_mv(A, x))
150         w = _cholesky_solve(L, r)
151         p = w
152         g = _cholesky_solve(L, b)
153         gamma = math.sqrt( mul_vv(g, b) )
154
155         rsold = mul_vv(w, r)
156
157         while nsteps < maxiter:
158             nsteps += 1
159
160             Ap = mul_mv(A, p)
161             alpha = rsold / mul_vv(p, Ap)
162
163             x = add_vv(x, mul_sv(alpha, p))
164             r = sub_vv(r, mul_sv(alpha, Ap))
165             w = _cholesky_solve(L, r)
166
167             rsnew = mul_vv(w, r)
168             if math.sqrt(rsnew) < gamma * eps:
169                 break
170
171             p = add_vv(w, mul_sv(rsnew / rsold, p))
172             rsold = rsnew
173
174     return (x, nsteps)

```

main.py

```

1 from sympy import symbols, Float, log, cos, sin
2 from utils import *
3
4 import math
5
6 from model import Model
7 from linalg import *
8 from grid import *
9

```

```

10  a = 0
11  b = 1
12  c = 0
13  d = 2
14
15  chi2 = 2
16  chi3 = 3
17
18  x, y = symbols("x y")
19
20  def print_matrix(m):
21      dummy = ""
22
23      sz = m.size()
24
25      print(f"{dummy:3s}", end=" ")
26      for i in range(sz):
27          print(f"{i:7d}", end=" ")
28      print()
29
30      for i in range(sz):
31          print(f"{i:3d}", end=" ")
32          for j in range(sz):
33              x = m.get(i, j)
34              if x == 0:
35                  print(f" {0:+7.2f}", end=" ")
36              else:
37                  print(f" {m.get(i, j):+7.2f}", end=" ")
38          print()
39
40  def print_matrix2(m):
41      dummy = ""
42
43      sz = m.size()
44
45      # print(f"{dummy:3s}", end=" ")
46      for i in range(sz):
47          for j in range(sz):
48              x = m.get(i, j)
49              if x == 0:
50                  pass
51              else:
52                  print("\fill[color=blue!60] (" , i, "," , sz - j, ")" , "rectangle (" ,
53                      ↪ j, ",", sz - i, ");")
54      print()

```

```

55 def print_vector(label, v):
56     print(f"{label}: ", end="")
57
58     for x in v:
59         print(f"{x:+.5f}", end=" ")
60
61     print()
62
63
64 test_cases = [
65     # ( (x + y + 1)**2, (x + y + 1)**3, Float(1) * x**2 ),
66     # ( (x + y + 1), (x + y + 1), (2 * x + 3 * y + 1) ),
67     ( (x + y + 1)**2, (x + y + 1)**2, (2 * x + 3 * y + 1) * x**2 ),
68     # ( (x + y + 1)**3, (x + y + 1)**3, (2 * x + 3 * y + 1) ),
69     # ( Float(1), Float(1), (2 * x + 3 * y + 1)**2 ),
70     # ( (x + y + 1), (x + y + 1), (2 * x + 3 * y + 1)**2 ),
71     # ( (x + y + 1), (x + y + 1), (2 * x + 3 * y + 1)**3 * x**2 ),
72     # ( (cos(x * y) + 2), (sin(x) * sin(y) + 2), (x * log(y + 1)) ),
73 ]
74
75 fmt = "{:>2s}, {:>3s}, {:>3s}, {:>4s}, {:>8s}, {:>8s}, {:>8s}, {:>4s}, {:>8s}, {:>8s},
    ↪ {:>8s}"
76 header = fmt.format(
77     "no", "Nx", "Ny",
78     "n1", "delta11", "delta12", "delta13",
79     "n2", "delta21", "delta22", "delta23"
80 )
81 print(header)
82
83 for test_num, test_case in enumerate(test_cases):
84
85     k1, k2, u = test_case
86
87     g2 = get_g2(b, chi2, k1, u)
88     g3 = get_g3(c, chi3, k2, u)
89     g4 = get_g4(d, u)
90
91     f = get_f(u, k1, k2)
92     k1 = get_k1(k1)
93     k2 = get_k2(k2)
94     u = get_u(u)
95
96     mod = Model(a, b, c, d, [k1, k2], f, [chi2, chi3], [g2, g3, g4])
97
98     sizes = [
99         # (2, 2),

```

```

100     (4, 4),
101     # (8, 8),
102     # (16, 16),
103     # (32, 32),
104     # (64, 64),
105 ]
106
107 for Nx, Ny in sizes:
108     A, g, L = mod.init_data(Nx, Ny)
109     sz = A.size()
110
111     # for i in g:
112     #     print("g =", i)
113
114     x1, _ = get_x(a, b, Nx)
115     y1, _ = get_y(c, d, Ny)
116
117     x0 = [0] * sz
118
119     x = []
120     for j in range(Ny + 1):
121         for i in range(Nx + 1):
122             x.append(u(x1(i), y1(j)))
123
124     print_matrix2(A)
125
126     # y, n_steps_1 = conjgrad(A, g, x0, eps=1e-12)
127
128     # # print_vector("x", x)
129     # # print_vector("y", y)
130     # # print_vector("g", g)
131     # # print_vector("Ay", mul_mv(A, y))
132     # # print_vector("Ax", mul_mv(A, x))
133
134     # delta = sub_vv(x, y)
135
136     # delta_11 = sum( [abs(i) for i in delta] ) / sum( [abs(i) for i in x] )
137     # delta_12 = math.sqrt( mul_vv(delta, delta) / mul_vv(x, x) )
138     # delta_13 = abs( max(delta, key=lambda x: abs(x)) / max(x, key=lambda x:
139     ↪      abs(x)) )
140
141     # y, n_steps_2 = conjgrad(A, g, x0, eps=1e-8, L=L)
142
143     # delta = sub_vv(x, y)
144
145     # delta_21 = sum( [abs(i) for i in delta] ) / sum( [abs(i) for i in x] )

```

```

145     # delta_22 = math.sqrt( mul_vv(delta, delta) / mul_vv(x, x) )
146     # delta_23 = abs( max(delta, key=lambda x: abs(x)) / max(x, key=lambda x:
    ↪     abs(x)) )
147
148     # fmt = "{:2d}, {:3d}, {:3d}, {:4d}, {:8.5e}, {:8.5e}, {:8.5e}, {:4d}, {:8.5e},
    ↪     {:8.5e}, {:8.5e}"
149     # record = fmt.format(
150     #     test_num, Nx, Ny,
151     #     n_steps_1, float(delta_11), float(delta_12), float(delta_13),
152     #     n_steps_2, float(delta_21), float(delta_22), float(delta_23),
153     # )
154     # # print(record, flush=True)

```

model.py

```

1  import math
2  from typing import Callable, Tuple
3
4  from grid import *
5  from sparse_matrix import *
6  from linalg import *
7
8  real1_fn = Callable[[float], float]
9  real2_fn = Callable[[float, float], float]
10
11
12  def _cholesky_decomp(A, Nx, Ny):
13      n = (Nx + 1) * (Ny + 1)
14      m = Nx + 1
15
16      a = [A.get(i, i) for i in range(n)]
17      b = [A.get(i, i + 1) for i in range(n - 1)]
18      c = [A.get(i, i + m) for i in range(n - m)]
19
20      a_new = []
21      b_new = []
22      c_new = []
23
24      L = SparseMatrixCOO()
25
26      for i in range(n):
27
28          if i - 1 < 0:
29              ai = math.sqrt(a[i])
30          elif i - m < 0:
31              ai = math.sqrt(a[i] - b_new[i - 1]**2)
32          else:
33              ai = math.sqrt(a[i] - b_new[i - 1]**2 - c_new[i - m]**2)

```



```

34
35     a_new.append(ai)
36
37     if i < n - 1:
38         bi = b[i] / ai
39         b_new.append(bi)
40
41     if i < n - m:
42         ci = c[i] / ai
43         c_new.append(ci)
44
45     for i in range(n):
46         L.set(i, i, a_new[i])
47
48     for i in range(n - 1):
49         L.set(i, i + 1, b_new[i])
50
51     for i in range(n - m):
52         L.set(i, i + m, c_new[i])
53
54     return transpose(L)
55
56
57 class Model:
58     def __init__(self,
59                 a: float, b: float,
60                 c: float, d: float,
61                 k: Tuple[real2_fn],
62                 f: real2_fn,
63                 chi: Tuple[float],
64                 g: Tuple[real1_fn]) -> None:
65         self.a = a
66         self.b = b
67         self.c = c
68         self.d = d
69         self.k = k
70         self.f = f
71         self.chi = chi
72         self.g = g
73
74     def init_data(self, Nx: int, Ny: int):
75
76         k1, k2 = self.k
77         f = self.f
78
79         chi2, chi3 = self.chi #

```

```

80     g2, g3, g4 = self.g #\varphi      ,      ,
81
82     hx = (self.b - self.a) / Nx
83     hy = (self.d - self.c) / Ny
84
85     x1, x2 = get_x(self.a, self.b, Nx)
86     y1, y2 = get_y(self.c, self.d, Ny)
87
88     M = (Nx + 1) * (Ny + 1)
89     L = (Nx + 1)
90
91     mtrx = SparseMatrixCOO()
92     g = [0] * M
93
94     # inner space
95     for i in range(1, Nx):
96         for j in range(1, Ny):
97             m = j * L + i
98
99             #print("inner space i =", i, "j =", j, "m =", m)
100
101             a1 = (hx / hy) * x1(i) * k2( x1(i), y2(j - 1) )
102             a2 = (hy / hx) * x2(i - 1) * k1( x2(i - 1), y1(j) )
103             a3 = (hy / hx) * x2(i) * k1( x2(i), y1(j) )
104             a4 = (hx / hy) * x1(i) * k2( x1(i), y2(j) )
105
106             mtrx.set(m, m - L, -a1)
107             mtrx.set(m, m - 1, -a2)
108             mtrx.set(m, m, a1 + a2 + a3 + a4)
109             mtrx.set(m, m + 1, -a3)
110             mtrx.set(m, m + L, -a4)
111             g[m] = hx * hy * x1(i) * f(x1(i), y1(j))
112
113     # left edge
114     i = 0
115     for j in range(1, Ny):
116         m = j * L + i
117
118         #print("left edge i=", i, "j =", j, "m =", m)
119
120         a1 = (hx / hy / 2) * (x2(i) / 2) * k2( x1(i), y2(j - 1) )
121         a3 = (hy / hx) * x2(i) * k1( x2(i), y1(j) )
122         a4 = (hx / hy / 2) * (x2(i) / 2) * k2( x1(i), y2(j) )
123
124         mtrx.set(m, m - L, -a1)
125         mtrx.set(m, m, a1 + a3 + a4)

```

```

126         mtrx.set(m, m + 1, -a3)
127         mtrx.set(m, m + L, -a4)
128         g[m] = (hx / 2) * (x2(i) / 2) * hy * f(x1(i), y1(j))
129
130     # right edge
131     i = Nx
132     for j in range(1, Ny):
133         m = j * L + i
134
135         #print("right edge i=", i, "j =", j, "m =", m)
136
137         a1 = (hx / hy / 2) * x1(i) * k2(x1(i), y2(j - 1))
138         a2 = (hy / hx) * x2(i - 1) * k1(x2(i - 1), y1(j))
139         a3 = hy * x1(i) * chi2
140         a4 = (hx / hy / 2) * x1(i) * k2(x1(i), y2(j))
141
142         mtrx.set(m, m - L, -a1)
143         mtrx.set(m, m - 1, -a2)
144         mtrx.set(m, m, a1 + a2 + a3 + a4)
145         mtrx.set(m, m + L, -a4)
146         g[m] = (hx / 2) * x1(i) * hy * f(x1(i), y1(j)) + hy * x1(i) * g2(y1(j))
147
148     # bottom edge
149     j = 0
150     for i in range(1, Nx):
151         m = j * L + i
152
153         #print("bottom edge i=", i, "j =", j, "m =", m)
154
155         a1 = hx * x1(i) * chi3
156         a2 = (hy / hx / 2) * x2(i - 1) * k1(x2(i - 1), y1(j))
157         a3 = (hy / hx / 2) * x2(i) * k1(x2(i), y1(j))
158         a4 = (hx / hy) * x1(i) * k2(x1(i), y2(j))
159
160         mtrx.set(m, m - 1, -a2)
161         mtrx.set(m, m, a1 + a2 + a3 + a4)
162         mtrx.set(m, m + 1, -a3)
163         mtrx.set(m, m + L, -a4)
164         g[m] = hx * (hy / 2) * x1(i) * f(x1(i), y1(j)) + hx * x1(i) * g3(x1(i))
165
166     # top edge
167     j = Ny
168     for i in range(1, Nx):
169         m = j * L + i
170
171         #print("top edge i=", i, "j =", j, "m =", m)

```

```

172
173         mtrx.set(m, m, 1)
174         g[m] = g4(x1(i))
175
176     # left bottom
177     i, j = 0, 0
178     m = j * L + i
179
180     #print("left bottom i=", i, "j =", j, "m =", m)
181
182     a1 = (hx / 2) * (x2(i) / 2) * chi3
183     a3 = (hy / hx / 2) * x2(i) * k1(x2(i), y1(j))
184     a4 = (hx / hy / 2) * (x2(i) / 2) * k2(x1(i), y2(j))
185
186     mtrx.set(m, m, a1 + a3 + a4)
187     mtrx.set(m, m + 1, -a3)
188     mtrx.set(m, m + L, -a4)
189     g[m] = (hx / 2) * (hy / 2) * (x2(i) / 2) * f(x1(i), y1(j)) + (hx / 2) * (x2(i)
        ↪ / 2) * g3(x1(i))
190
191     # right bottom
192     i, j = Nx, 0
193     m = j * L + i
194
195     #print("right bottom i=", i, "j =", j, "m =", m)
196
197     a1 = (hy / 2) * x1(i) * chi2
198     a2 = (hy / hx / 2) * x2(i - 1) * k1(x2(i - 1), y1(j))
199     a3 = (hx / 2) * x1(i) * chi3
200     a4 = (hx / hy / 2) * x1(i) * k2(x1(i), y2(j))
201
202     mtrx.set(m, m - 1, -a2)
203     mtrx.set(m, m, a1 + a2 + a3 + a4)
204     mtrx.set(m, m + L, -a4)
205     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))
206
207     # right top corner
208     i, j = Nx, Ny
209     m = j * L + i
210
211     #print("right top i=", i, "j =", j, "m =", m)
212
213     mtrx.set(m, m, 1)
214     g[m] = g4(x1(i))
215

```

```

216     # left top corner
217     i, j = 0, Ny
218     m = j * L + i
219
220     # print("left top i=", i, "j =", j, "m =", m)
221
222     mtrx.set(m, m, 1)
223     g[m] = g4(x1(i))
224
225     #remove extra elements
226     # for i in range(1, Nx):
227     #     m1 = (Ny - 1) * L + i
228     #     m2 = Ny * L + i
229     #     x = mtrx.get(m1, m1 + L)
230     #     mtrx.set(m1, m1 + L, 0)
231
232     #     g[m1] -= g[m2] * x
233
234     # m1 = (Ny - 1) * L
235     # m2 = Ny * L
236     # x = mtrx.get(m1, m1 + L)
237     # mtrx.set(m1, m1 + L, 0)
238
239     # g[m1] -= g[m2] * x
240
241     # m1 = (Ny - 1) * L + Nx
242     # m2 = Ny * L + Nx
243     # x = mtrx.get(m1, m1 + L)
244     # mtrx.set(m1, m1 + L, 0)
245
246     # g[m1] -= g[m2] * x
247
248     # L = _cholesky_decomp(mtrx, Nx, Ny)
249
250     # mtrx = mtrx.to_symmetric().to_CSR()
251     print(f"LOG: {is_symmetric(mtrx)}")
252     # L = L.to_CSR()
253
254     return (mtrx, g, L)

```

sparse_matrix.py

```

1  from typing import Tuple, List
2  from abc import ABC, abstractmethod
3  import numpy as np
4
5
6  class SparseMatrix(ABC):

```

```

7     @abstractmethod
8     def get(self, i: int, j: int=None) -> float:
9         pass
10
11    @abstractmethod
12    def size(self) -> int:
13        pass
14
15    @abstractmethod
16    def is_symmetric() -> bool:
17        pass
18
19
20    class SparseMatrixCOO(SparseMatrix):
21        def __init__(self, symmetric=False) -> None:
22            self.symmetric = symmetric
23            self.row = []
24            self.col = []
25            self.data = []
26
27        def get(self, i: int, j: int=None) -> float:
28            # self._validate_index(i, j)
29
30            if j == None:
31                return self._get_by_row(i)
32
33            if self.symmetric:
34                i, j = self._get_symmetric_index(i, j)
35
36            idx_pairs = zip(self.row, self.col)
37
38            for idx, pair in enumerate(idx_pairs):
39                ir, ic = pair
40
41                if ir == i and ic == j:
42                    return self.data[idx]
43
44            return 0
45
46        def _get_by_row(self, i):
47            rez = [(self.col[idx], self.data[idx]) for idx in range(len(self.row)) if
48                ↪ self.row[idx] == i]
49            return rez
50
51        def set(self, i: int, j: int, x: float) -> None:
52            # self._validate_index(i, j)

```

```

52
53     if self.symmetric:
54         i, j = self._get_symmetric_index(i, j)
55
56     idx_pair = zip(self.row, self.col)
57
58     for idx, pair in enumerate(idx_pair):
59         ir, ic = pair
60
61         if ir == i and ic == j:
62             if x == 0:
63                 self.row.pop(idx)
64                 self.col.pop(idx)
65                 self.data.pop(idx)
66             else:
67                 self.data[idx] = x
68             return
69
70         if x != 0:
71             self._add_value(i, j, x)
72
73     def size(self) -> int:
74         max_row = max(self.row) + 1
75         max_col = max(self.col) + 1
76         return max(max_row, max_col)
77
78     def is_symmetric(self) -> bool:
79         return self.symmetric
80
81     def to_symmetric(self):
82         m = SparseMatrixCOO(symmetric=True)
83         for i, j, d in zip(self.row, self.col, self.data):
84             m.set(i, j, d)
85         return m
86
87     def to_CSR(self):
88         row = np.array(self.row)
89         col = np.array(self.col)
90         data = np.array(self.data)
91
92         permutation = sorted(range(len(self.row)), key=lambda idx: row[idx])
93
94         row = row[permutation]
95         col = col[permutation]
96         data = data[permutation]
97

```

```

98         new_row = []
99
100        for r in np.unique(row):
101            index = row == r
102            new_row.append( row.tolist().index(r) )
103
104            col_per_row = col[index]
105            data_per_row = data[index]
106
107            permutation = sorted(range( len(col_per_row) ), key=lambda idx:
108                                ↪ col_per_row[idx])
109
110            col_per_row = col_per_row[permutation]
111            data_per_row = data_per_row[permutation]
112
113            col[index] = col_per_row
114            data[index] = data_per_row
115
116        row = np.array(new_row)
117
118        return SparseMatrixCSR(row, col, data, symmetric=self.symmetric)
119
120    def _add_value(self, i: int, j: int, x: float) -> None:
121        self.row.append(i)
122        self.col.append(j)
123        self.data.append(x)
124
125    def _validate_index(self, i: int, j: int) -> None:
126        if i < 0 or j < 0:
127            raise KeyError()
128
129    def _get_symmetric_index(self, i: int, j: int) -> Tuple[int, int]:
130        return (j, i) if j > i else (i, j)
131
132    class SparseMatrixCSR(SparseMatrix):
133        def __init__(self, row: np.ndarray, col: np.ndarray, data: np.ndarray,
134                    ↪ symmetric=False) -> None:
135            self.symmetric = symmetric
136            self.row = row
137            self.col = col
138            self.data = data
139
140        def get(self, i: int, j: int=None) -> float:
141            # self._validate_index(i, j)

```



```

142         if j == None:
143             return self._get_by_row(i)
144
145         if self.symmetric:
146             i, j = self._get_symmetric_index(i, j)
147
148         sz = self.size()
149
150         ic1 = self.row[i]
151         ic2 = None if i + 1 == sz else self.row[i + 1]
152
153         try:
154             idx = ic1 + self.col[ic1:ic2].tolist().index(j)
155         except ValueError:
156             return 0
157
158         return self.data[idx]
159
160     def _get_by_row(self, i):
161         sz = self.size()
162
163         ic1 = self.row[i]
164         ic2 = None if i + 1 == sz else self.row[i + 1]
165
166         col = self.col[ic1:ic2].tolist()
167         data = self.data[ic1:ic2].tolist()
168
169         return [(j, d) for j, d in zip(col, data)]
170
171     def size(self) -> int:
172         return self.col.max() + 1
173
174     def is_symmetric(self) -> bool:
175         return self.symmetric
176
177     def _validate_index(self, i: int, j: int) -> None:
178         sz = self.size()
179         if i < 0 or i > sz:
180             raise KeyError()
181         if j < 0 or j > sz:
182             raise KeyError()
183
184     def _get_symmetric_index(self, i: int, j: int) -> Tuple[int, int]:
185         return (j, i) if j > i else (i, j)

```

utils.py

```

1 from typing import Callable

```

```

2  from sympy import Symbol, symbols, simplify, Subs, N, diff
3
4
5  def get_u(u: Symbol) -> Callable[[float, float], float]:
6      def inner(x: float, y: float):
7          expr = Subs(u, "x", x)
8          expr = Subs(expr, "y", y)
9          return N(expr)
10
11     return inner
12
13 def get_k1(k1: Symbol) -> Callable[[float, float], float]:
14     def inner(x: float, y: float):
15         expr = Subs(k1, "x", x)
16         expr = Subs(expr, "y", y)
17         return N(expr)
18
19     return inner
20
21 def get_k2(k2: Symbol) -> Callable[[float, float], float]:
22     def inner(x: float, y: float):
23         expr = Subs(k2, "x", x)
24         expr = Subs(expr, "y", y)
25         return N(expr)
26
27     return inner
28
29 def get_f(u: Symbol, k1: Symbol, k2: Symbol) -> Callable[[float, float], float]:
30     x= symbols("x")
31     expr1 = x * k1 * diff(u, "x")
32     expr2 = k2 * diff(u, "y")
33     f = -( (1 / x) * diff(expr1, "x") + diff(expr2, "y") )
34     f = simplify(f)
35
36     # print("f =", f)
37
38     def inner(x: float, y: float):
39         expr = Subs(f, "x", x)
40         expr = Subs(expr, "y", y)
41         return N(expr)
42
43     return inner
44
45 # def get_g1(a: float, chi1: float, k1: Symbol, u: Symbol) -> Callable[[float, float],
46 #     float]:
47 #     g1 = Subs(chi1 * u - k1 * diff(u, "x"), "x", a)

```

```

47 #     def inner(y: float):
48 #         return N( Subs(g1, "y", y) )
49
50 #     return inner
51
52 # def get_g2(b: float, u: Symbol) -> Callable[[float, float], float]:
53 #     g2 = Subs(u, "x", b)
54 #     def inner(y: float):
55 #         return N( Subs(g2, "y", y) )
56
57 #     return inner
58
59 # def get_g3(c: float, chi3: float, k2: Symbol, u: Symbol) -> Callable[[float, float],
60 #     ↪ float]:
61 #     g3 = Subs(chi3 * u - k2 * diff(u, "y"), "y", c)
62 #     def inner(x: float):
63 #         return N( Subs(g3, "x", x) )
64
65 #     return inner
66
67 # def get_g4(d: float, chi4: float, k2: Symbol, u: Symbol) -> Callable[[float, float],
68 #     ↪ float]:
69 #     g4 = Subs(chi4 * u + k2 * diff(u, "y"), "y", d)
70 #     def inner(x: float):
71 #         return N( Subs(g4, "x", x) )
72
73 #     return inner
74
75 def get_g2(b: float, chi2: float, k1: Symbol, u: Symbol) -> Callable[[float, float],
76     ↪ float]:
77     g2 = Subs(chi2 * u + k1 * diff(u, "x"), "x", b)
78     def inner(y: float):
79         return N( Subs(g2, "y", y))
80
81     return inner
82
83 def get_g3(c: float, chi3: float, k2: Symbol, u: Symbol) -> Callable[[float, float],
84     ↪ float]:
85     g3 = Subs(chi3 * u - k2 * diff(u, "y"), "y", c)
86     def inner(x: float):
87         return N( Subs(g3, "x", x))
88
89     return inner
90
91 def get_g4(d: float, u: Symbol) -> Callable[[float, float], float]:
92     g4 = Subs(u, "y", d)

```

```
89     def inner(x: float):
90         return N( Subs(g4, "x", x))
91
92     return inner
```
