

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В.ЛОМОНОСОВА»

МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МЕХАНИКИ ВЫПУСКНАЯ

КВАЛИФИКАЦИОННАЯ РАБОТА
(ДИПЛОМНАЯ РАБОТА)
специалиста

**«Многосеточный метод для спектрально-элементных дискретизаций
линейных уравнений теории упругости»**

Выполнил студент
626 группы
Аристеев Сергей Алексеевич

подпись студента

Научный руководитель:
Профессор Вершинин Анатолий Викторович

подпись научного руководителя

Москва

2025

Содержание

1	Введение	3
1.1	Актуальность	3
1.2	Цели	3
1.3	Апробация работы	3
2	Постановка задачи теории упругости	3
3	Метод спектральных элементов	4
3.1	Основные принципы метода конечных элементов	4
3.2	Слабая вариационная постановка	4
3.3	Применение к статической задаче линейной теории упругости	5
3.4	Матричная запись уравнений	5
3.5	Численное интегрирование	6
3.5.1	Квадратура Гаусса	6
3.5.2	Квадратура Гаусса-Лежандра-Лобатто	7
3.5.3	Естественная система координат	8
3.6	Ключевые особенности метода спектральных элементов	9
3.7	Повышение порядка спектральных элементов	10
3.8	Свойства матрицы жесткости	11
4	Многосеточные методы	13
4.1	Построение иерархии сеток и операторов перехода	14
4.2	Галеркинская проекция	14
4.3	Сглаживатель и грубый решатель	15
4.4	Циклы многосеточного метода	16
4.5	Алгоритм программного модуля, реализующего многосеточный метод.	18
5	Численные эксперименты	19
6	Заключение	25
A	Метод сопряженных градиентов и предобуславливание	26
B	Листинги	27

1 Введение

1.1 Актуальность

Линейные уравнения теории упругости лежат в основе моделирования деформаций и напряжений в конструкциях, геологических средах, биомеханических системах и других областях механики сплошных сред. Современные инженерные задачи требуют высокоточной дискретизации сложных геометрий, что делает спектрально-элементные методы особенно востребованными благодаря их экспоненциальной сходимости и гибкости в аппроксимации [13].

Однако увеличение размерности задач приводит к резкому росту вычислительных затрат, особенно при решении систем линейных уравнений, возникающих после дискретизации. Многосеточные методы [9] являются одним из наиболее эффективных подходов для ускорения итерационных решателей за счет комбинации сеток разной детализации. Актуальность работы обусловлена необходимостью разработки специализированных многосеточных алгоритмов для спектрально-элементных дискретизаций, которые сохраняют преимущества и обеспечат снижение вычислительной сложности, без требований переноса решения с между разными неструктурированными сетками.

Предыдущие курсовые работы были посвящены применению библиотечных (Nvidia AMG) реализаций алгебраических многосеточных методов, которые показали свою эффективность на ряде задач. В данной работе описан многосеточный метод реализованный самостоятельно.

1.2 Цели

- Разработка многосеточного метода для спектрально-элементных дискретизаций линейных уравнений теории упругости.
- Реализация программного модуля на языке C++, реализующий многосеточный метод в применение к методу спектральных элементов, а также иные итерационные методы.
- Оптимизация алгоритма для минимизации вычислительных ресурсов. Применение формата неявного хранения матрицы в памяти.
- Сравнительное тестирование с традиционными методами (метод сопряженных градиентов).

1.3 Апробация работы

Основные положения и результаты работы докладывались на XIII всероссийском съезде по теоретической и прикладной механике (21-25 августа 2023 г., Санкт-Петербург), а также на конференции студентов, аспирантов и молодых учёных «Ломоносов» 2023-2025 гг.

2 Постановка задачи теории упругости

Полная система уравнений статической задачи линейной теории упругости заданная в перемещениях [15] состоит из:

- уравнения равновесия в области Ω

$$\nabla \cdot \sigma(\varepsilon) = 0, \quad x \in \Omega \quad (1)$$

σ — тензор напряжений, ε — тензор деформаций

- определяющих соотношений - закон Гука для изотропного тела

$$\sigma(\varepsilon) = \lambda I_1(\varepsilon)I + 2\mu\varepsilon \quad (2)$$

λ, μ — коэффициенты Ламе, $I_1(\varepsilon)$ — первый инвариант тензора деформаций, I — единичный (тождественный) тензор второго ранга

- определения тензора малых деформаций

$$\varepsilon(u) = \frac{1}{2} (\nabla u + (\nabla u)^T) \quad (3)$$

u - вектор перемещений

- граничных условий Неймана

$$\vec{n} \cdot \sigma(\varepsilon) = \vec{n} \cdot P(x), \quad x \in \Gamma_1 \in \partial\Omega \quad (4)$$

P — сила, приложенная на границе, \vec{n} — нормаль к границе,

- граничных условий Дирихле

$$u(x) = u_0(x), \quad x \in \Gamma_2 \in \partial\Omega \quad (5)$$

u_0 - заданная функция на границе

3 Метод спектральных элементов

Метод спектральных элементов - модификация метода конечных элементов, поэтому начнем с описания классического метода конечных элементов

3.1 Основные принципы метода конечных элементов

Согласно [12] основные принципы:

- Сплошная среда разделяется воображаемыми линиями или поверхностями на некоторое количество конечных элементов
- Предполагается, что элементы связаны между собой в узловых точках, расположенных на их границах
- Выбирается система функций, однозначно определяющая значения внутри конечного элемента по значениям в узловых точках, которые могут находиться внутри и на границах конечного элемента

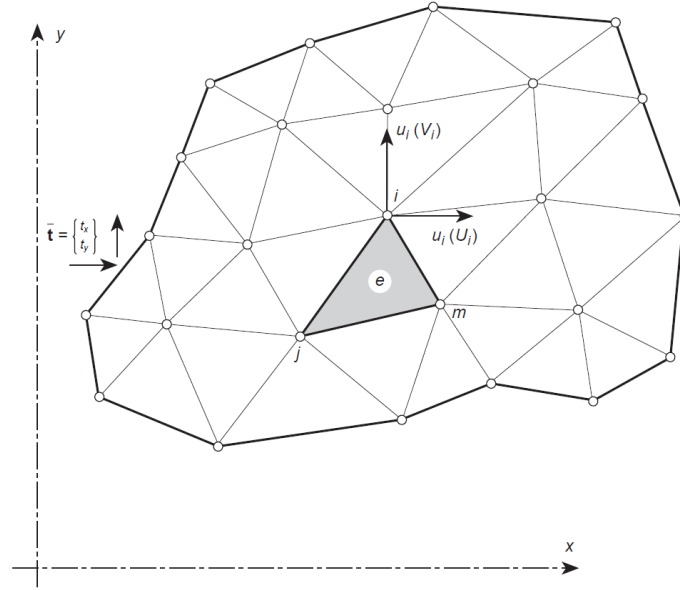


Рис. 1: Плоская область разбитая на конечные элементы

Произвольная величина $u(x)$ внутри элемента e (по рис. 1) представляется как

$$u|_e(x) = u_i N_i(x) + u_j N_j(x) + u_k N_k(x),$$

где u_i значение в точке x_i , $N_i(x)$ — произвольная функция, ассоциированная с узлом x_i , также называемая функцией форм.

3.2 Слабая вариационная постановка

Пусть дана система дифференциальных уравнений и заданы граничные условия:

$$Lu(x) = f \quad x \in \Omega$$

$$Bu(x) = 0 \quad x \in \Gamma = \partial\Omega$$

где L, B - дифференциальные операторы

Для приближенного решения системы дифференциальных уравнений необходимо минимизировать невязку

$$R\tilde{u}(x) := f - L\tilde{u}(x)$$

где \tilde{u} - приближенное решение, представляемое в каком-то заранее выбранном функциональном пространстве \mathcal{W} в виде комбинации базисных функций:

$$\tilde{u}(x) = \sum_{k=1}^N \tilde{u}_k \phi_k(x)$$

Чтобы минимизировать невязку на этом пространстве задается N условий с помощью скалярных произведений в \mathcal{W} в слабой формулировке:

$$(v, Ru)_\Omega + (v, Bu)_\Gamma = 0$$

3.3 Применение к статической задаче линейной теории упругости

При подстановке уравнения состояния:

$$\int_{\Omega} v \nabla \cdot \sigma(u) d\Omega + \int_{\Gamma} v (P \cdot \vec{n} - \vec{n} \cdot \sigma(u)) d\Gamma = 0, \quad \forall v \in \mathcal{W}$$

После выбора конечномерного пространства \mathcal{W}_n :

$$u(x) = \sum_{k=1}^n U_k N_k(x) \quad (6)$$

N_k - функции форм в \mathcal{W}_n

$$\int_{\Omega} N_i \nabla \cdot \sigma(u) d\Omega + \int_{\Gamma} N_i (P \cdot \vec{n} - \vec{n} \cdot \sigma) d\Gamma = 0, \quad i = 1, 2, \dots, n$$

После применения формулы Грина для первого интеграла:

$$\int_{\Gamma} N_i \vec{n} \cdot \sigma d\Gamma - \int_{\Omega} \nabla N_i \sigma d\Omega + \int_{\Gamma} N_i (P \cdot \vec{n} - \vec{n} \cdot \sigma) d\Gamma = 0$$

После раскрытия скобок можно сократить первый и последний интегралы

$$\int_{\Gamma} N_i \vec{n} \cdot \sigma d\Gamma - \int_{\Omega} \nabla N_i \sigma d\Omega + \int_{\Gamma} N_i P \cdot \vec{n} - \int_{\Gamma} N_i \vec{n} \cdot \sigma d\Gamma = 0$$

Уравнения состояния принимает вид:

$$\int_{\Omega} \nabla N_i \sigma d\Omega = \int_{\Gamma} N_i P \cdot \vec{n} d\Gamma, \quad i = 1, 2, \dots, n \quad (7)$$

3.4 Матричная запись уравнений

Выражение неизвестной функции (6) представляется в матричном виде:

$$u(x) = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = \begin{bmatrix} N_1 & 0 & 0 & \dots & N_n & 0 & 0 \\ 0 & N_1 & 0 & \dots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \dots & 0 & 0 & N_n \end{bmatrix} \begin{pmatrix} U_x^1 \\ U_y^1 \\ U_z^1 \\ \dots \\ U_x^n \\ U_y^n \\ U_z^n \end{pmatrix}$$

или

$$\{u\} = [N]\{U\}$$

Тензор деформаций ε имеет шесть независимых компонент, которые записываются в виде вектора:

$$\{\varepsilon\} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_y}{\partial y} \\ \frac{\partial u_z}{\partial z} \\ \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \\ \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y} \\ \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} = [\partial] \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix}$$

или

$$\{\varepsilon\} = [\partial]\{u\}$$

Аналогично записывается тензор напряжений. Для изотропного материала:

$$\{\sigma\} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{bmatrix} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \nu & \nu & 0 & 0 & 0 \\ \nu & 1 & \nu & 0 & 0 & 0 \\ \nu & \nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = [D] \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix}$$

или:

$$\{\sigma\} = [D]\{\varepsilon\}$$

Подстановка выражений в матричном виде (7) с учетом того, что $\{\nabla N_i\} = [\partial][N] = [B]$:

$$\int_{\Omega} [B]^T [D] [B] \{U\} d\Omega = \int_{\Gamma} [N]^T \{P \cdot \vec{n}\} d\Gamma.$$

Область Ω разбивается на непересекающиеся подобласти - элементы Ω_e

$$\Omega = \bigsqcup_{e=1}^E \Omega_e$$

При этом функции форм, по которым вычисляются значения внутри элемента, снаружи элемента 0. Поэтому интегралы преобразуются в суммы интегралов:

$$\sum_{e=1}^E \int_{\Omega_e} [B]^T [D] [B] \{U\} d\Omega_e = \sum_{e=1}^E \int_{\Gamma_k} [N]^T \{P \cdot \vec{n}\} d\Gamma_e$$

Выражение принимает вид:

$$[K]\{U\} = \{F\},$$

где $[K]$ - матрица жесткости (или глобальная матрица жесткости), $[K_e]$ - локальная матрица жесткости:

$$[K] = \sum_{e=1}^E [K_e] = \sum_{e=1}^E \int_{\Omega_e} [B]^T [D] [B] d\Omega_e$$

F - вектор правых частей:

$$\{F\} = \sum_{e=1}^E \int_{\Gamma_e} [N]^T \{P \cdot \vec{n}\} d\Gamma_e$$

или без скобок:

$$KU = F \tag{8}$$

полученная система линейных алгебраических уравнений.

3.5 Численное интегрирование

3.5.1 Квадратура Гаусса

Для численного интегрирования при составлении матрицы жесткости используются квадратурные формулы [11]:

$$\int_a^b f(x) dx \approx \sum_{i=1}^N f(x_i) w_i,$$

где точки x_i называются узлами квадратуры, а w_i - весами

Значения x_i и w_i , определяются из $2N$ условий точности квадратурной формулы для полиномов степени вплоть до $2N - 1$.

Граничные точки отрезка, вообще говоря, не являются узлами квадратуры Гаусса, что в применении к методу конечных элементов усложняет согласование соседних элементов и учет граничных условий.

3.5.2 Квадратура Гаусса-Лежандра-Лобатто

Квадратура Гаусса-Лежандра-Лобатто может определяться аналогично квадратуре Гаусса: граничные точки являются узлами квадратуры, остальные узлы и веса определяются из $2N - 2$ условий на точность численного интегрирования полиномов степени вплоть до $2N - 3$

Однако для подсчета узлов и весов квадратуры удобнее пользоваться другим определением квадратуры [2] с использованием полиномов Лежандра:

узлы квадратуры: ± 1 и $n - 2$ корня $L'_{n-1}(\xi)$ производной полинома Лежандра степени $n - 1$,

веса:

$$w_i = \begin{cases} \frac{2}{n(n+1)} & \text{если } i = 1 \text{ или } i = n \\ \frac{2}{n(n+1)L_n(\xi_i)^2} & \text{иначе} \end{cases}$$

Запишем выражения некоторых полиномов Лежандра:

$$L_0(\xi) = 1$$

$$L_1(\xi) = \xi$$

$$L_2(\xi) = \frac{1}{2}(3\xi^2 - 1)$$

Рекуррентная формула полиномов Лежандра:

$$L_n(\xi) = \frac{(2n-1)\xi L_{n-1}(\xi) - (n-1)L_{n-2}(\xi)}{n} \quad n \geq 2$$

Рекуррентная формулы для производных

$$L'_n(\xi) = \frac{n[L_{n-1}(\xi) - \xi L_n(\xi)]}{1 - \xi^2}$$

$$L''_n(\xi) = \frac{2\xi L'_n(\xi) - n(n+1)L_n(\xi)}{1 - \xi^2}$$

$$L'''_n(\xi) = \frac{4\xi L''_n(\xi) - [n(n+1) - 2]L'_n(\xi)}{1 - \xi^2}$$

Для поиска нулей производной $L'_{n-1}(\xi)$ производной полинома Лежандра воспользуемся методом Галлея, описанным в [2]

Алгоритм 1: Алгоритм поиска точек и узлов квадратуры Гаусса-Лежандра-Лобатто

```

1  Функция GLL( $n, x$ )
2   $x_1 := -1$ ;
3   $x_n := w_1$ ;
4   $w_1 := \frac{2}{n(n-1)}$ ;
5   $w_n := w_1$ ;
6  Для  $i := 2 \dots n-1$ 
7   $x_i := \left[1 - \frac{3(n-2)}{8(n-1)^3}\right] \cos\left(\frac{4i-3}{4(n-1)+1}\pi\right)$ ;
8   $e := 1$ ;
9  Пока  $e > 10^{-15}$ 
10   $y := L'_{n-1}(x_i)$ ;
11   $y' := L''_{n-1}(x_i)$ ;
12   $y'' := L'''_{n-1}(x_i)$ ;
13   $\Delta x := \frac{2yy'}{2(y')^2 - yy''}$ ;
14   $x_i := x_i - \Delta x$ ;
15   $e := |\Delta x|$ ;
16   $w_i := \frac{2}{n(n-1) \cdot L_{n-1}(x_i)^2}$ ;
17 возврат  $x, w$ ;
```

Количество узлов	x_i	w_i
2	-1	1
	1	1
3	-1	0.333
	0	1.333
	1	0.333
4	-1	0.167
	-0.44721	0.833
	0.44721	0.833
	1	0.167
5	-1	0.1
	-0.65465	0.544
	0	0.711
	0.65465	0.544
	1	0.1
6	-1	0.067
	-0.76506	0.378
	-0.28523	0.555
	0.28523	0.555
	0.76506	0.378
	1	0.067

Таблица 1: Пример вычисленных узлов и весов квадратуры Гаусса-Лежандра-Лобатто (значения округлены)

3.5.3 Естественная система координат

Для вычисления интегралов по элементу удобно пользоваться естественной системой координат:

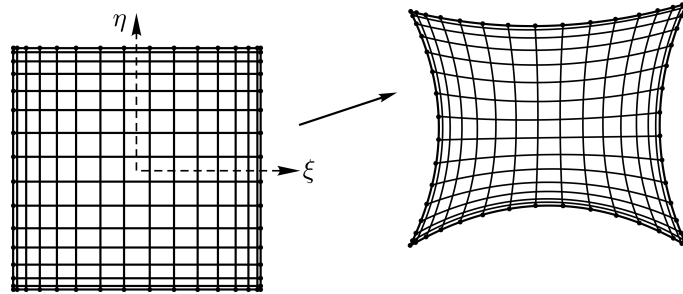


Рис. 2: Криволинейный четырехугольник в естественных координатах (слева) и в реальных (справа)

Пусть $\mathbf{x} = (x, y)$ и $\boldsymbol{\xi} = (\xi, \eta)$, тогда реальные координаты четырехугольника можно представить с помощью функций форм определенных на правильном четырехугольнике

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}) = \sum_{k=1}^n \mathbf{x}_k N_k(\boldsymbol{\xi}),$$

где \mathbf{x}_k - узлы в реальных координатах.

Интегралы по элементам можно переписать в виде:

$$\int_{\Omega_e} f(\mathbf{x}) d\Omega_e = \int_{\Omega_e} f(\mathbf{x}(\boldsymbol{\xi})) d\Omega_e = \int_{\Omega_e} f(\boldsymbol{\xi}) |J_e(\boldsymbol{\xi})| d\Omega_e$$

где

$$J_e(\boldsymbol{\xi}) = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi})$$

матрица Якоби, которую можно вычислить следующим образом

$$J_e(\xi) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} (\xi) = \begin{bmatrix} \sum_{k=1}^n x_k \frac{\partial N_k}{\partial \xi} & \sum_{k=1}^n y_k \frac{\partial N_k}{\partial \xi} & \sum_{k=1}^n z_k \frac{\partial N_k}{\partial \xi} \\ \sum_{k=1}^n x_k \frac{\partial N_k}{\partial \eta} & \sum_{k=1}^n y_k \frac{\partial N_k}{\partial \eta} & \sum_{k=1}^n z_k \frac{\partial N_k}{\partial \eta} \\ \sum_{k=1}^n x_k \frac{\partial N_k}{\partial \zeta} & \sum_{k=1}^n y_k \frac{\partial N_k}{\partial \zeta} & \sum_{k=1}^n z_k \frac{\partial N_k}{\partial \zeta} \end{bmatrix} (\xi)$$

Для вычисления производных функций форм также удобно пользоваться естественными координатами:

$$\begin{pmatrix} \frac{\partial N_k}{\partial \xi} \\ \frac{\partial N_k}{\partial \eta} \\ \frac{\partial N_k}{\partial \zeta} \end{pmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{pmatrix} \frac{\partial N_k}{\partial x} \\ \frac{\partial N_k}{\partial y} \\ \frac{\partial N_k}{\partial z} \end{pmatrix}$$

Для чего потребуется обратить матрицу Якоби

$$\begin{pmatrix} \frac{\partial N_k}{\partial x} \\ \frac{\partial N_k}{\partial y} \\ \frac{\partial N_k}{\partial z} \end{pmatrix} = J_e^{-1} \begin{pmatrix} \frac{\partial N_k}{\partial \xi} \\ \frac{\partial N_k}{\partial \eta} \\ \frac{\partial N_k}{\partial \zeta} \end{pmatrix}$$

3.6 Ключевые особенности метода спектральных элементов

Метод спектральных элементов [13] - это модификация метода конечных элементов. Впервые был применен в попытке объединить точность спектральных [3] и геометрическую гибкость конечно-элементных методов [5] в задачах волновой и газовой динамики для моделирования течения жидкостей.

Ключевые особенности:

- многомерные сетки получаются как декартово произведение одномерных сеток,
- отсюда ограничение на тип элементов - четырехугольные в двумерных задачах и гексаэдральные в трехмерных
- функции форм обладают дельта-свойством: $N_i(x_j) = \delta_{ij}$ - символ Кронекера в узле элемента x_j
- отсюда коэффициенты U_i в разложении (6) - значение в узле x_i
- узлы элемента совпадают с узлами квадратуры
- функции форм - полиномы Лагранжа, построенные по узлам

$$N_i(\xi) = l_i(\xi) = \prod_{j=0, i \neq j}^n \frac{(\xi - \xi_j)}{(\xi_i - \xi_j)}$$

Для двумерных и трехмерных задач

$$N_{i,j}(\xi, \eta) = l_i(\xi)l_j(\eta)$$

$$N_{i,j,k}(\xi, \eta, \zeta) = l_i(\xi)l_j(\eta)l_k(\zeta)$$

$-1 \leq \xi, \eta, \zeta \leq 1$ - естественные координаты элемента (для квадрата или куба)

С точки зрения классического метода конечных элементов метод спектральных элементов обладает рядом преимуществ:

- спектральная (экспоненциальная) сходимость с ростом порядка полиномов
- легкое варьирование порядка - перестроение сетки не требуется

- функции форм ортогональны при численном интегрировании (в силу дельта-свойства и совпадения узлов квадратуры с узлами элемента)
- узлы автоматически уплотняются вблизи к границе элемента

Метод широко применяется в динамических задачах механики твердого деформируемого тела, в частности в сейсмических задачах. Ортогональность при численном интегрировании приводит к удобному применению явных схем интегрирования по времени, а высокий порядок полиномов позволяет отлично моделировать волновые процессы.

Однако применимость только для гексаэдральных сеток может существенно ограничить круг задач для применения.

3.7 Повышение порядка спектральных элементов

Метод спектральных элементов для уточнения решения не требует полной перестройки расчетной неструктурированной сетки, вместо этого повышается порядок аппроксимирующих полиномов в представлении (6).

Таким образом при фиксированном разбиении области на элементы в каждый элемент сетки добавляются узлы согласно квадратуре Гаусса-Лежандра-Лобатто.

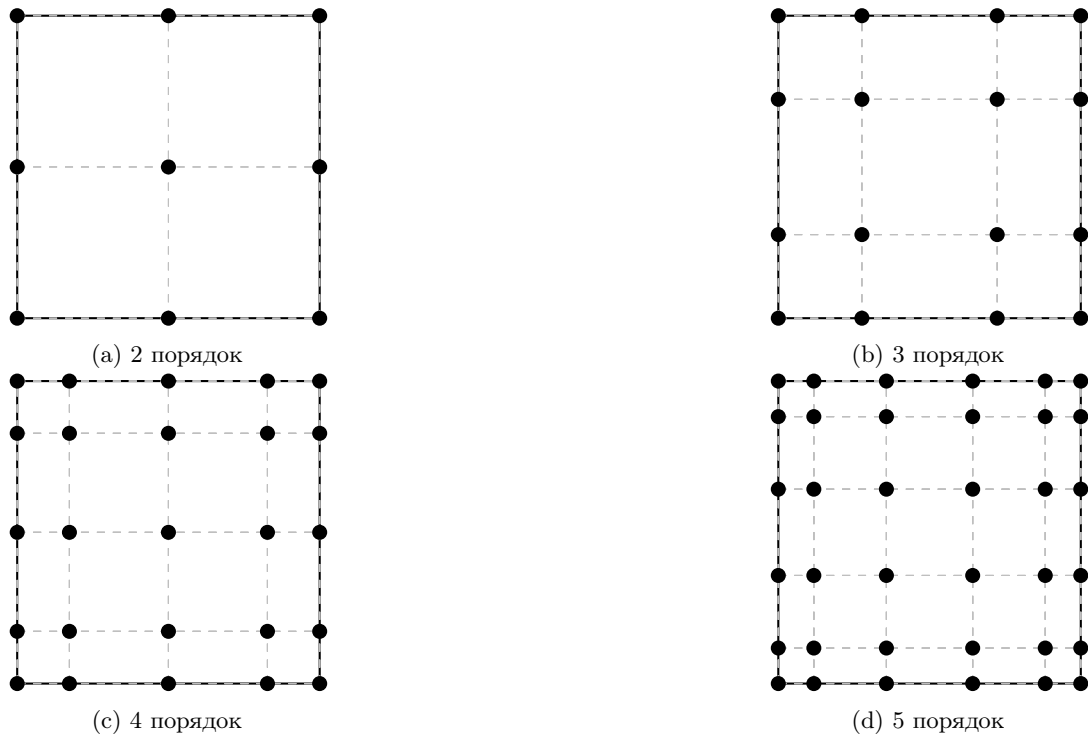
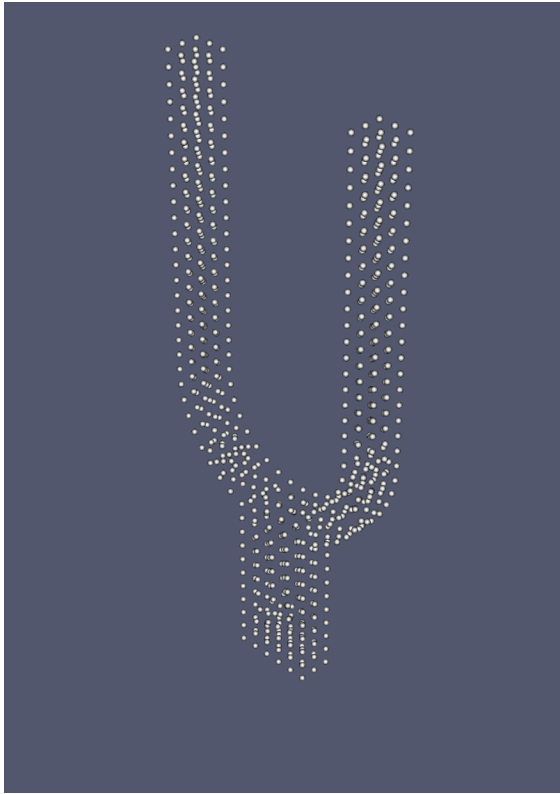
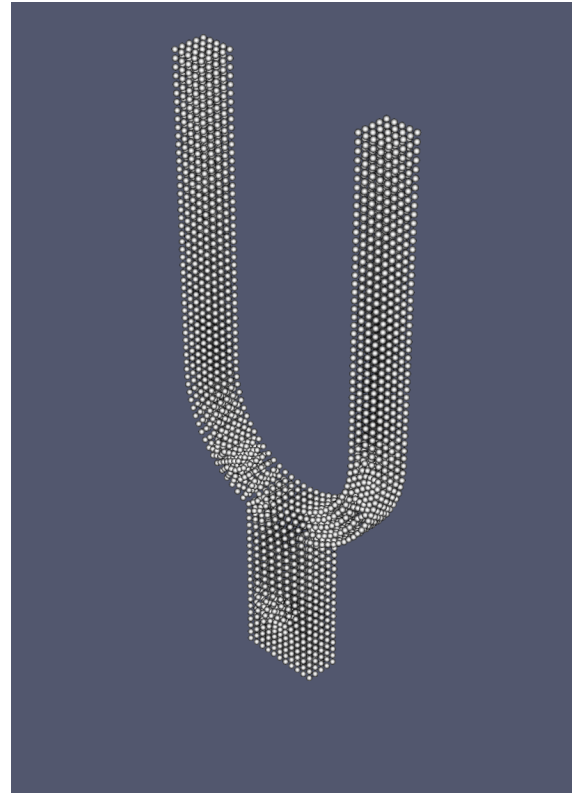


Рис. 3: Спектральный четырехугольник разных порядков

И пример для модели:



(a) 1 порядок



(b) 2 порядок

Рис. 4: Узлы в представлении модели спектральными элементами первого и второго порядков

Упорядоченность внутри одного элемента позволяет обойти основное ограничение геометрического многосеточного метода в применении к методу конечных элементов - необходимость перестраивать неструктурированную сетку и переносить решение с одной расчетной сетки на другую.

3.8 Свойства матрицы жесткости

С алгебраической точки зрения, матрица системы (8) симметрична и положительно определена при некоторых условиях.

Симметричность следует из вида локальных матриц жесткости, а так как $\frac{1}{2}(U, KU)$ отвечает энергии деформации [13] - части потенциальной энергии при перемещениях U , то K положительно определена при условии того, что перемещения U не реализует движение тела как абсолютно твердого [12].

С технической точки зрения матрица жесткости является разреженной, то есть количество ненулевых значений много меньше количества всех элементов матрицы, в силу разбиения области на элементы.

Это свойство отражается при хранении матрицы в памяти. В настоящее время наиболее активно применяются формат Block Compressed Sparse Row (BSR) - модификация Compressed Sparse Row (CSR), который в качестве элемента матрицы хранит не число, а квадратный блок фиксированного размера.

BSR использует три числа и три массива для представления матрицы:

1. *size* - размер блока
2. *n* - число количество блочных строк матрицы
3. *nnz* - число количество ненулевых элементов в матрице
4. *vals[nnz]* - массив длиной *nnz* - содержит все ненулевые элементы матрицы, перечисленные построчно
5. *cols[nnz]* - массив длиной *nnz* - для каждого ненулевого числа содержит его столбец
6. *rows[n + 1]* - массив длиной *n + 1* - для строки *i* указывает на начало *rows[n]* и конец *rows[n + 1]* каждой строки в массивах *vals* и *cols*

Однако стоит заметить, что матрица жесткости, вообще говоря, состоит из плотных пересекающихся подматриц, размерность которых при повышении порядка спектрального элемента растет пропорционально кубу порядка.

Для большинства итерационных методов решения систем линейных алгебраических уравнений требуется только лишь результаты некоторых матрично-векторных операций, в частности, умножения матрицы на вектор.

Приняв ограничение на выбор численных методов, можно использовать неявное представление матрицы в памяти: хранить только плотные локальные матрицы жесткости без вычисления глобальной матрицы жесткости, или вовсе отказаться от хранения матрицы в памяти, многократно вычисляя сразу результат матрично-векторного умножения. В данной работе использовался первый метод, в силу применения параллелизма на центральном процессоре (CPU), а не на графических процессорах (GPU), где полностью безматричный метод предпочтительнее [7].

Таким образом матрица жесткости представляется как:

- массив плотных подматриц
- граф связности элементов и его раскраска по вершинам
- массив индексов узлов для каждого элемента

Матрично-векторное умножение в таком случае выглядит как сумма частичных матрично-векторных умножения подматрицы на подвектор:

$$KU = \left[\sum_{e=1}^E K_e \right] U = \sum_{e=1}^E [K_e U_e]$$

Частичные матрично-векторные умножения происходят параллельно для несмежных элементов согласно раскраске графа их связности. Такой подход позволил ускорить работу метода сопряженных градиентов (см приложение А)

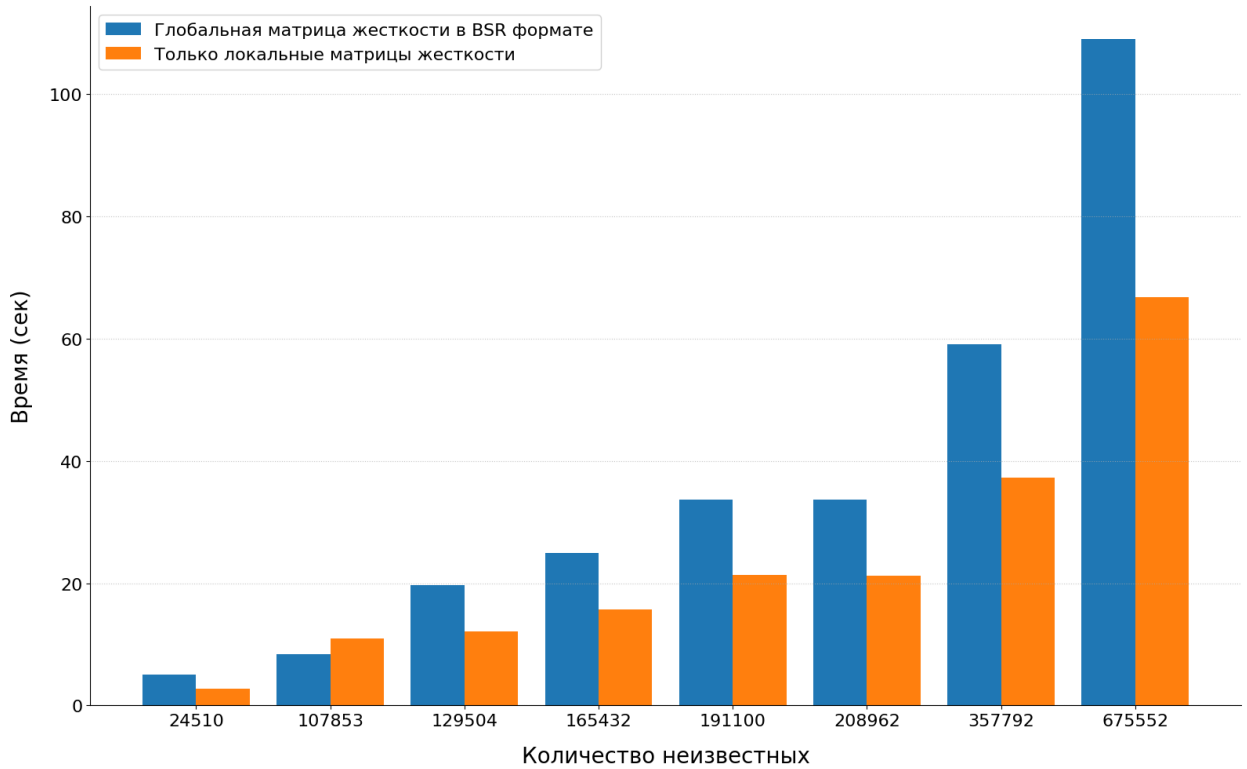
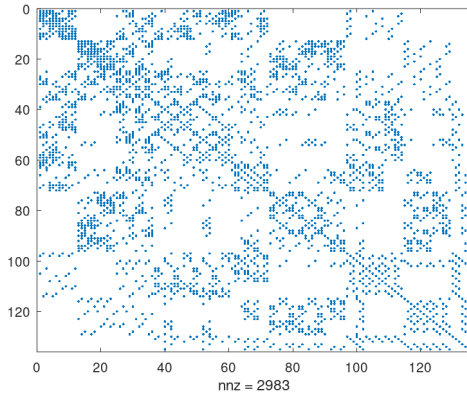
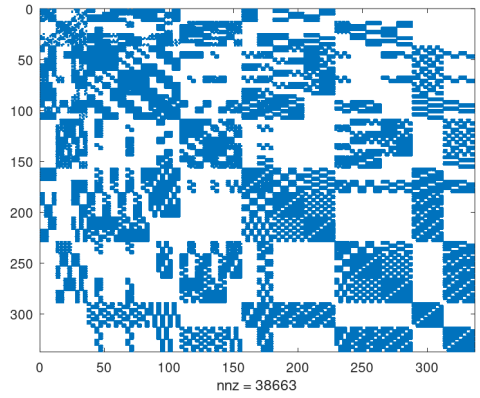


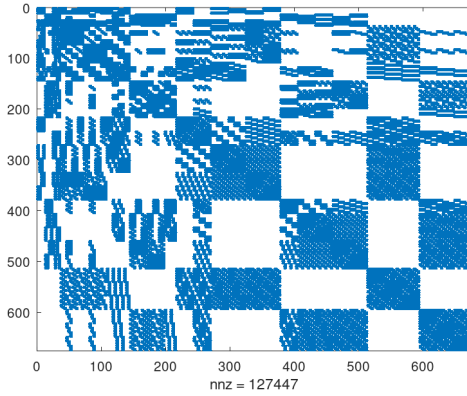
Рис. 5: Время 1000 итераций метода сопряженных градиентов с матрицей в BSR формате (Sparse Matrix-Vector Multiplication, SpMV Intel MKL) и неявном виде для спектральных элементов 5 порядка (частичные матрично-векторные умножения также с помощью Intel MKL - Basic Linear Algebra Subprograms - BLAS)



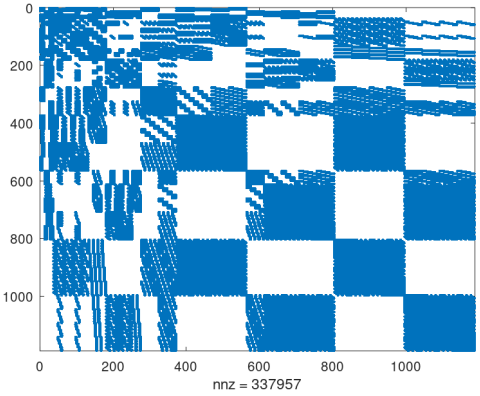
(a) 2 порядок



(b) 3 порядок



(c) 4 порядок



(d) 5 порядок

Рис. 6: Портреты матриц жесткости модели из двух спектральных элементов разного порядка

4 Многосеточные методы

Большинство итерационных методов, таких как метод Якоби, обладают медленной скоростью сходимости. Однако было установлено [17], [14], что компоненты ошибки, соответствующие собственным векторам матрицы с наибольшими собственными значениями (так называемые высокочастотные моды), затухают значительно быстрее остальных.

Напротив, низкочастотные моды, связанные с наименьшими собственными значениями, плохо подавляются стандартными итерационными алгоритмами. Это явление объясняет характерное замедление сходимости базовых итерационных методов.

Примечательно, что многие из низкочастотных мод естественным образом трансформируются в высокочастотные при переходе на более грубую сетку. Данное наблюдение легло в основу идеи многосеточных методов, где рекурсивное использование иерархии сеток позволяет эффективно устранять различные компоненты ошибки.

Многосеточные методы [9] одновременно используют сразу несколько уровней дискретизации с разными расчетными сетками для достижения оптимальной скорости сходимости.

Иерархия сеток в случае регулярных сеток может формироваться через равномерное деление ячеек: на две, четыре или восемь частей в зависимости от размерности задачи. Для метода конечных элементов, где преобладают нерегулярные сетки, применение этого подхода затруднено.

Описанные выше методы относятся к классу так называемых геометрических многосеточных алгоритмов, поскольку оперируют геометрической структурой сеток. Альтернативой выступают алгебраические многосеточные методы [8], [10], разработанные для случаев отсутствия информации о сеточной структуре.

Отдельного внимания заслуживают многосеточные методы, основанные на адаптации по порядку аппроксимации (р-адаптации) [6], [4]. Метод спектральных элементов для более точного решения повышает порядок базисных функций без изменения расчетной сетки. Это сводит задачу межсеточного переноса решения к преобразованию между полиномиальными представлениями разного порядка.

В силу своей гибкости многосеточный метод не устанавливает конкретный набор численных методов для решения той или иной задачи методами конечных разностей, конечных объемов или конечных элементов. Однако весьма условно можно сказать, что многосеточный подход состоит из:

- набора (иерархии) расчетных сеток и операторов переходов между ними

- сглаживателей - релаксационных алгоритмов для подавления высокочастотных мод ошибки, применяющихся на всех, кроме последнего уровня
- грубого решателя - численного метода для подавления низкочастотных мод ошибки, который применяется на последнем (грубом) уровне.

Опишем все эти три компонента в применении к методу спектральных элементов

4.1 Построение иерархии сеток и операторов перехода

Как ранее уже было указано метод спектральных элементов при повышении порядка не меняет структуру сетки, добавляя только узлы внутри элемента. Поэтому иерархия сеток строится просто: при фиксированном разбиении области на элементы понижается или повышается порядок каждого из элементов.

Построим теперь операторы перехода между уровнями иерархии. В МСЭ для любой характеристики $u(x)$ в представлении элементами порядка p_1 (здесь и далее будем использовать верхний индекс для обозначения различных порядков спектрального элемента.):

$$u(x) = \sum_{k=0}^{p_1} U_k^1 N_k^1(x)$$

При этом U_k^1 - значение в узле x_k^1 . Иными словами $u(x_k^1) = U_k^1$.
Также в представлении элементами порядка p_2 .

$$u(x) = \sum_{k=0}^{p_2} U_k^2 N_k^2(x)$$

При этом коэффициенты U_n^1 можно вычислить через другое представление

$$U_n^1 = u(x_n^1) = \sum_{k=0}^{p_1} U_k^2 N_k^2(x_n^1), \quad n = 0, \dots, p_1$$

и аналогично для коэффициентов U_m^2

$$U_m^2 = u(x_m^2) = \sum_{k=0}^{p_1} U_k^1 N_k^1(x_m^2), \quad m = 0, \dots, p_2$$

В матричном виде

$$\{U^1\} = [P^{1,2}]\{U^2\}, \quad \{U^2\} = [P^{2,1}]\{U^1\}, \quad P_{i,j}^{1,2} = N_j^2(x_i^1), \quad P_{i,j}^{2,1} = N_j^1(x_i^2) \quad (9)$$

И так как считаем, что $p_1 \neq p_2$, то матрицы $P^{2,1}$ и $P^{1,2}$ (здесь и далее квадратные скобки опущены) - прямоугольные. Причем эти соотношения верны для любых характеристик и одинаковы для всех элементов. Это позволит вычислить все матрицы перехода заранее и применять их аналогично матрично-векторному умножению на матрицу жесткости, хранимую неявно.

4.2 Галеркинская проекция

В многосеточном методе Галеркинская проекция - выражение грубой дискретизации A_H оператора через точное A_h с помощью операторов продолжения P и ограничения R :

$$A_H = R A_h P$$

В применении к задаче с нашими обозначениями

$$K^1 = R K^2 P$$

Для двух дискретизаций спектральными элементами разного порядка:

$$K^1 U^1 = F^1$$

$$K^2 U^2 = F^2$$

При подсчете энергии деформаций:

$$\Pi^1 = \frac{1}{2} U^{1T} K^1 U^1$$

$$\Pi^2 = \frac{1}{2} U^{2T} K^2 U^2$$

С учетом матрицы $P_{2,1}$:

$$\Pi^2 = \frac{1}{2} (P^{2,1} U^1)^T K^2 P^{2,1} U^1 = \frac{1}{2} U^{1T} P^{2,1T} K^2 P^{2,1} U^1$$

Энергия деформаций должна быть одинаковой на обоих уровнях дискретизации:

$$\Pi^1 = \Pi^2$$

$$\Pi^1 = \frac{1}{2} U^{1T} K^1 U^1 = \frac{1}{2} U^{1T} P^{2,1T} K^2 P^{2,1} U^1 = \Pi^2$$

или

$$K^1 = P^{2,1T} K^2 P^{2,1}$$

Отсюда следует, что оператор ограничения $R = P^{2,1T}$ и интерполяции $P = P^{2,1}$ транспонированы по отношению друг к другу.

4.3 Сглаживатель и грубый решатель

Для подавления высокочастотных мод ошибки применяется какой-то вычислительно недорогой итерационный метод, называемый сглаживателем, зачастую [14] [1] это метод Якоби, Гаусса-Зейделя или Чебышева.

Метод Гаусса-Зейделя требует плохо поддается распараллеливанию, для метода Чебышева необходимы минимальные и максимальные собственные значения матрицы жесткости, что затрудняет применение этих методов. Поэтому в реализации используется метод Якоби с параметром релаксации w , как наиболее простой и поддающийся распараллеливанию.

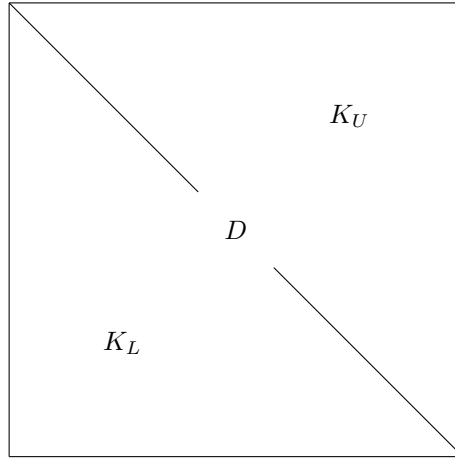


Рис. 7: Разбиение матрицы K

Алгоритм 2: Метод Якоби с параметром релаксации w

```

1 Функция  $\text{Jacobi}(K, U_0, F, w)$ 
2    $D := \text{diag}(K);$ 
3    $R_0 := F - KU;$ 
4    $i := 1;$ 
5   Пока не удовлетворяет критерию сходимости
6      $U_i := U_{i-1} + wD^{-1}R;$ 
7      $R_i := F - KU_i;$ 
8      $i := i + 1;$ 
9   возврат  $U_i;$ 
```

В качестве грубого решателя выступает MKL PARDISO, он поддерживает решение систем линейных алгебраических уравнений в формате BSR, методом LU разложения:

$$K = L_K U_K,$$

где L_K — нижнетреугольная, а U_K — верхнетреугольная. Тогда исходная задача (8) переписывается в виде:

$$KU = L_K U_K U = F$$

Для чего сначала прямой подстановкой решается задача:

$$L_K Z = F$$

После чего аналогично решается:

$$U_K U = Z$$

4.4 Циклы многосеточного метода

Многосеточные методы изначально применяли только две сетки - основную и дополнительную: решение происходит на основной, на дополнительной вычисляется поправка, которая должна обнулить невязку решения на основной сетке:

Алгоритм 3: Двухуровневый-цикл

```

1 Функция V2( $K^2, U_0^2, F^2$ )
2    $U^2 := \text{итерационный\_решатель}(K^2, U_0^2, F^2);$ 
3    $r^2 := F^2 - K^2 U^2;$ 
4    $r^1 := R^{1,2} r^2;$ 
5    $\delta^1 := \text{дополнительный\_решатель}(K^1, 0, r^1);$ 
6    $\delta^2 := P^{2,1} \delta^1;$ 
7    $U^2 := U^2 + \delta^2;$ 
8    $U^2 := \text{итерационный\_решатель}(K^2, U^2, F^2);$ 
9   возврат  $U^2;$ 

```

Данный алгоритм просто расширяется до рекурсивного V-цикла, состоящего из нескольких сеток, при этом итерационный решатель, называется сглаживателем, дополнительный - грубым.

Алгоритм 4: V-цикл

```

1 Функция V( $K^i, U_0^i, F^i$ )
2    $U^i := \text{сглаживатель}(K^i, U_0^i, F^i);$ 
3    $r^i := F^i - K^i U^i;$ 
4    $r^{i-1} := R^{i-1,i} r^i;$ 
5   если  $i \neq 1$  тогда
6      $\delta^{i-1} := V(K^{i-1}, 0, r^{i-1});$ 
7   иначе
8      $\delta^1 := \text{грубый\_решатель}(K^1, 0, r^1);$ 
9    $\delta^i := P^{i,i-1} \delta^{i-1};$ 
10   $U^i := U^i + \delta^i;$ 
11   $U^i := \text{сглаживатель}(K^i, U^i, F^i);$ 
12  возврат  $U^i;$ 

```

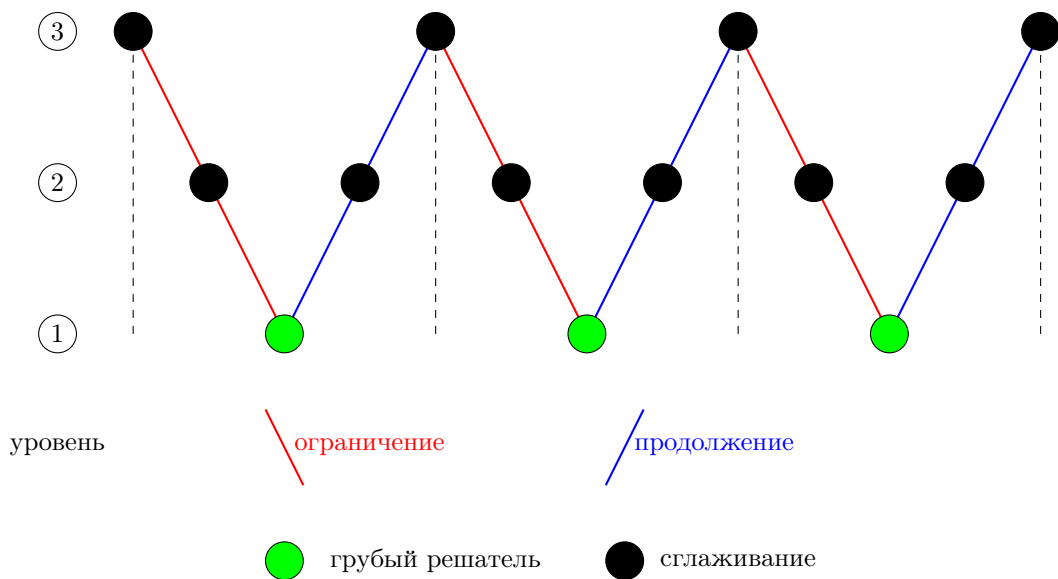


Рис. 8: Визуальная схема V-цикла - изображены 3 полные итерации цикла

Для более эффективного подавления низкочастотных мод ошибки, V-цикл может вызываться несколько раз на каждом уровне (W-цикл):

Алгоритм 5: W-цикл

```

1  Функция  $W(K^i, U_0^i, F^i)$ 
2   $U^i := \text{сглаживатель}(K^i, U_0^i, F^i);$ 
3   $r^i := F^i - K^i U^i;$ 
4   $r^{i-1} := R^{i-1,i} r^i;$ 
5  если  $i \neq 1$  тогда
6  |    $\delta^{i-1} := 0;$ 
7  |   Для  $m = 1 \dots \mu$ 
8  |   |    $\delta^{i-1} := v(K^{i-1}, \delta^{i-1}, r^{i-1});$ 
9  иначе
10 |    $\delta^1 := \text{грубый решатель}(K^1, 0, r^1);$ 
11  $\delta^i := P^{i,i-1} \delta^{i-1};$ 
12  $U^i := U^i + \delta^i;$ 
13  $U^i := \text{сглаживатель}(K^i, U^i, F^i);$ 
14  возврат  $U^i;$ 

```

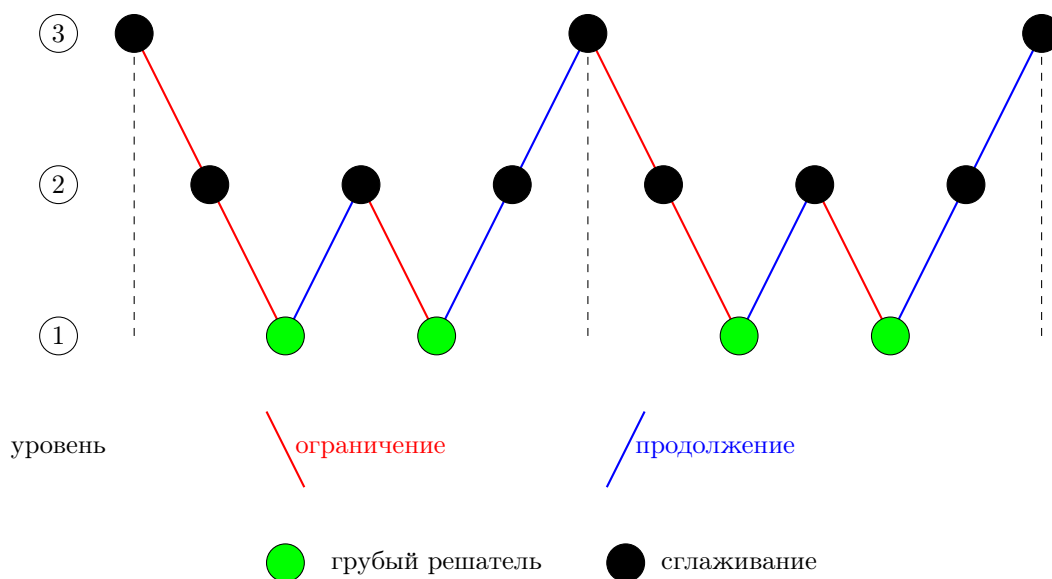


Рис. 9: Визуальная схема W-цикла - изображены 2 полные итерации

Многосеточные методы также используются для построения адаптивных численных методов. Так решение ищется сначала на грубой сетке, после чего строится более точная сетка, на которую интерполируется грубое решение, на ней вызывается V- или W-цикл, и так далее при необходимости в дальнейшем уточнении. Описанный алгоритм называется полным многосеточным методом или F-циклом.

В данной работе считается, что конечный порядок спектрального элемента задается пользователем, поэтому применение F-цикла ограничивается до поиска начального приближения для точного уровня:

Алгоритм 6: F-цикл

```

1  Функция  $F(K^i, U_0^i, F^i)$ 
2   $U^1 := \text{грубый решатель}(K^1, U^1, F^1);$ 
3  Для  $j = 2 \dots i$ 
4  |    $U^j := P^{j,j-1} U^{j-1};$ 
5  |    $U^j := W(K^j, U^j, F^j);$ 
6   возврат  $U^i;$ 

```

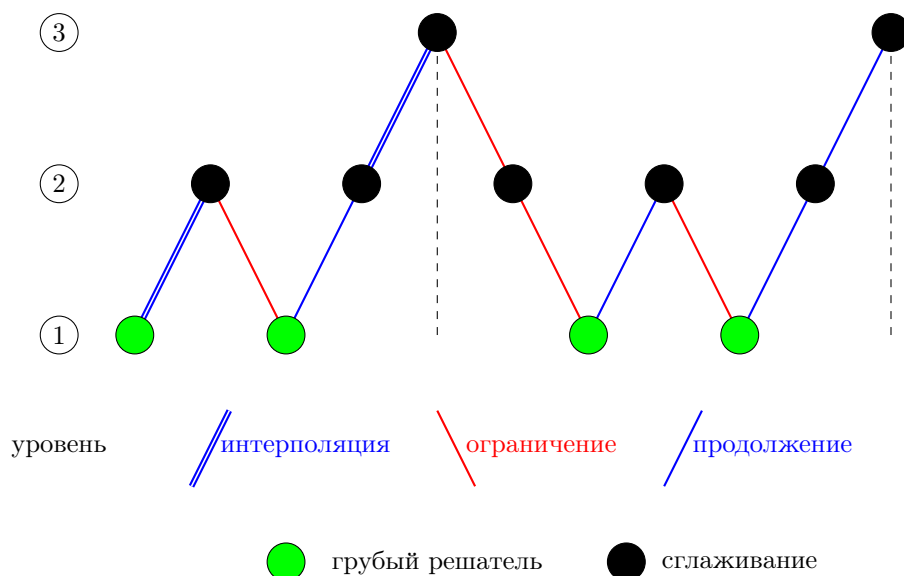


Рис. 10: Визуальная схема F-цикла, продолженного W-циклом (полный многосеточный метод)

Однако нужно отметить, важную роль F-цикла - можно полностью освободиться от выбора начального приближения.

Без применения F-цикла	7.5e+12	4.6e+10	6.0e+12	1.6+13	6.1e+12	6.3e+12	8.6e+11	3.3e+13
С применением F-цикла	1.5e+05	1.1e+02	4.5e+06	4.6+06	4.2e+04	5.3e+06	4.5e+03	3.9e+07

Таблица 2: Пример невязки на плохом начальном приближении - все единицы.

4.5 Алгоритм программного модуля, реализующего многосеточный метод.

1. Препроцессинг в CAE Fidesys

- Импорт CAD-геометрии и автоматическая генерация конечно-элементной (HEX-элементы) сетки.
- Назначение материалов и их параметров: модуль Юнга и коэффициент Пуассона.
- Формирование граничных условий Дирихле и Неймана.

2. Конфигурация решателя

- Порядок спектральных элементов в сетке.
- Тип итерационного решателя: метод сопряженных градиентов, многосеточный метод.
- Критерий сходимости.
- Для многосеточного решателя:
 - Порядок спектральных элементов в сетке.
 - Тип итерационного решателя: метод сопряженных градиентов, многосеточный метод.
 - Тип цикла: V, W(с параметром μ), F.
 - Алгоритм закрутки:
 - * уменьшением порядка всех элементов на заданное число, вплоть до заданного минимального.
 - * делением порядка всех элементов на заданное число, вплоть до заданного минимального.
 - * с помощью фиксированного набора значений.
 - Настройки сглаживателей: параметр релаксации метода Якоби, количество итераций.
 - Настройки грубого решателя: параметры для передачи в MKL PARDISO.

3. Чтение сетки, построение графа связности элементов и раскраска жадным алгоритмом.

4. Вычисление локальных матриц жесткости, вектора правой части, применение граничных условий.

5. Для многосеточного решателя повтор предыдущих двух пунктов, сборка глобальной матрицы жесткости на грубом уровне, факторизация.

6. Решение СЛАУ выбранным итерационным решателем.

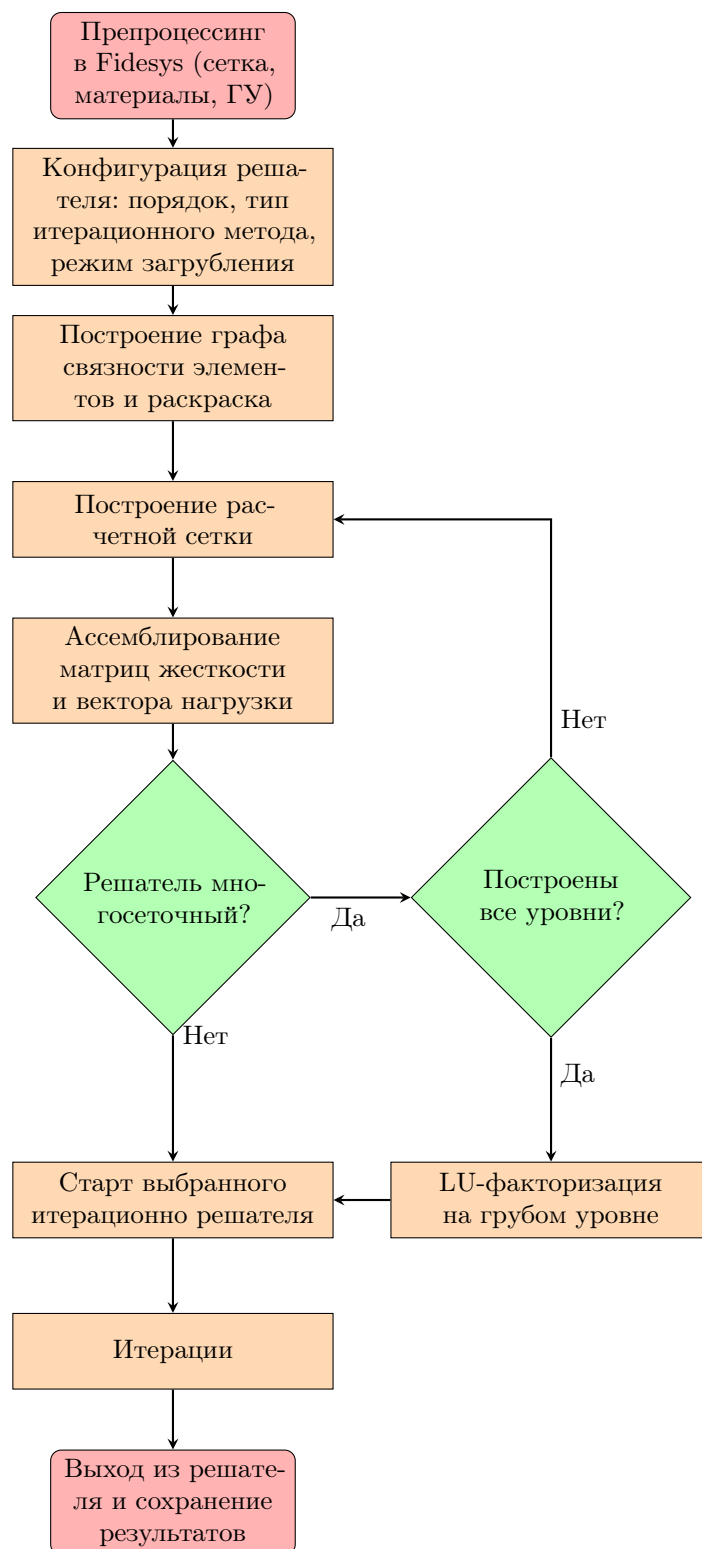


Рис. 11: Блок-схема программного модуля, реализующего многосеточный метод

5 Численные эксперименты

Расчеты проводились на серверном процессоре Intel Xeon 6710E (64 ядра, 2.4 ГГц), 256 Гб оперативной памяти.

Ниже приводится сравнение эффективности работы итерационных методов на 8 задачах. Были запущены 3 серии расчетов, с разным порядком спектральных элементов в сетке. Многосеточный метод сравнивается с методом сопряженных градиентов (приложение А), а также с предобусловленной версией.

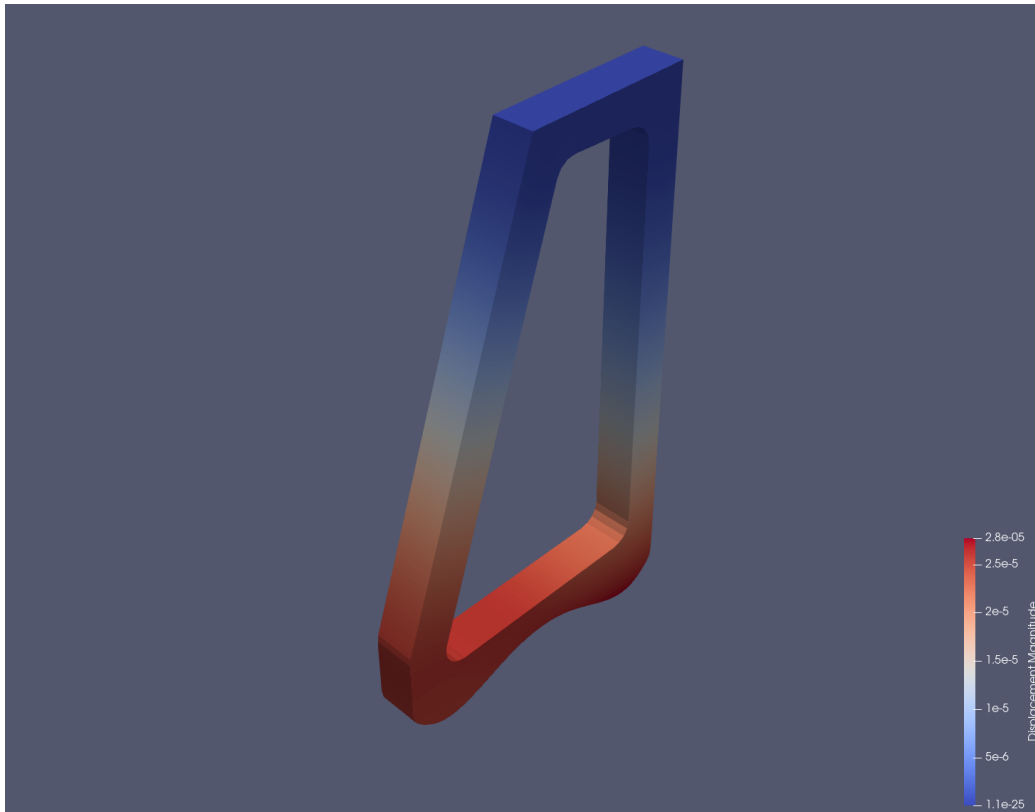


Рис. 12: Пример задачи: растяжение конструкции стального приклада

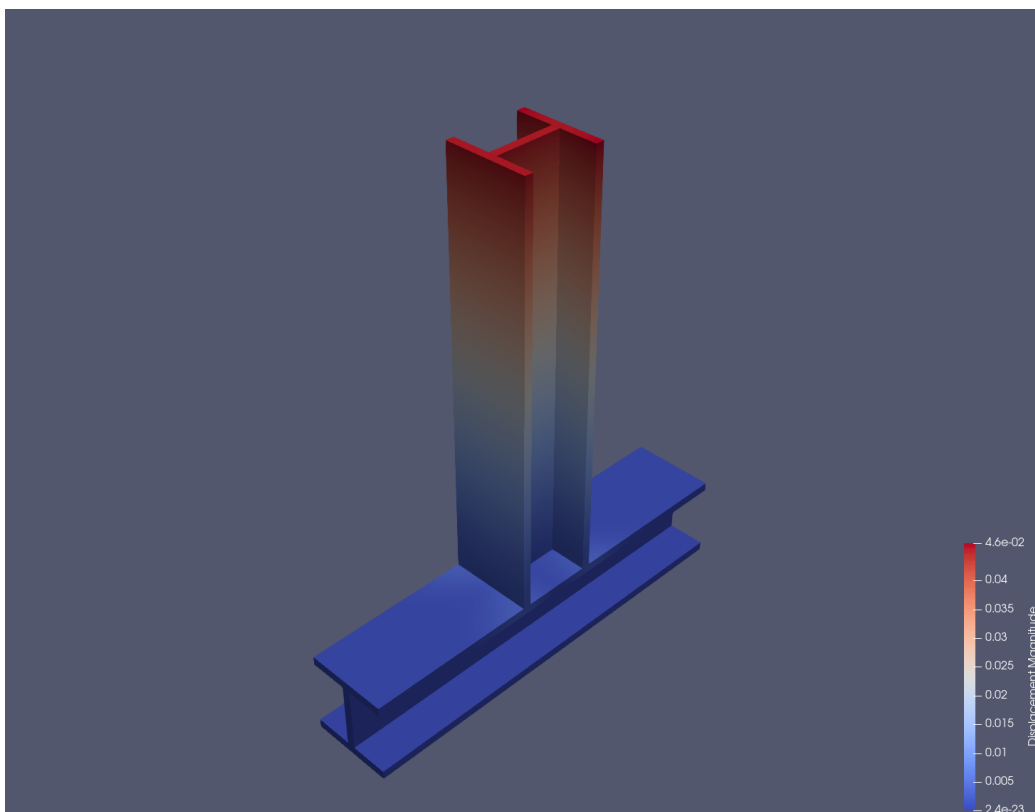


Рис. 13: Пример задачи: Т-сочленение двутавровых балок

На небольшом порядке спектральных элементов преимущества многосеточного метода не так ярко выражены или они вовсе отсутствуют (см 14 задача с 252 элементами)

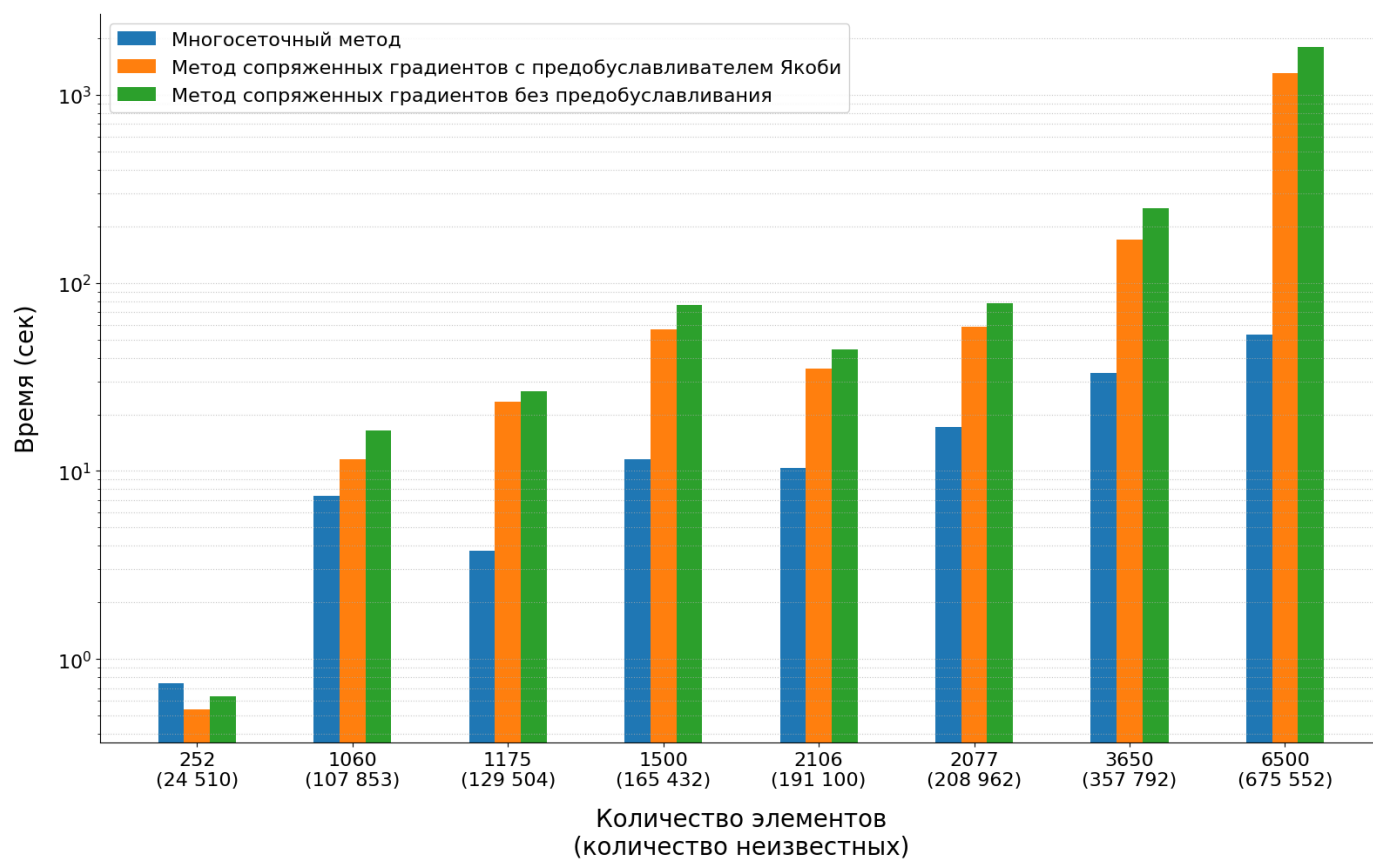


Рис. 14: В сетке спектральные элементы 3 порядка. Многосеточный решатель также достроил сетку с элементами 2 порядка

Однако при повышении порядка элементов многосеточный решатель становится наиболее эффективным - на задаче с самым большим количеством элементов (6500 элементов - 675 552 неизвестных, см рис. 15) ускорение по сравнению с методом сопряженных градиентов с предобусловливателем Якоби получилось в 41 раз (примерно 2 часа против 3 минут)

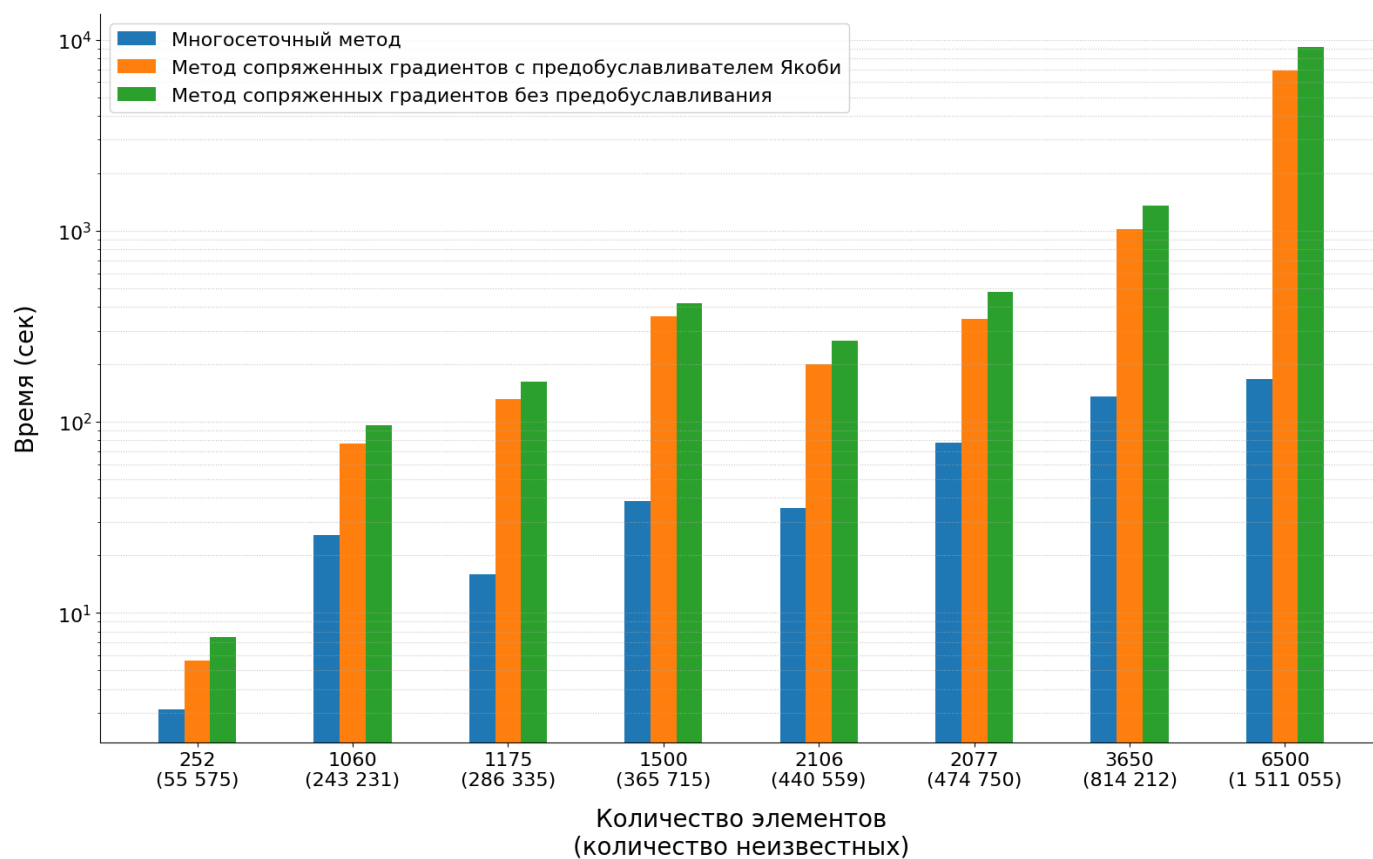


Рис. 15: В сетке спектральные элементы 4 порядка. Многосеточный решатель также достроил сетку с элементами 3 и 2 порядков

На 5 порядке ускорение по сравнению с методом сопряженных градиентов с предобуславливателем Якоби на этой же задаче получилось в 60 раз (примерно 7.5 часов против 7.5 минут)

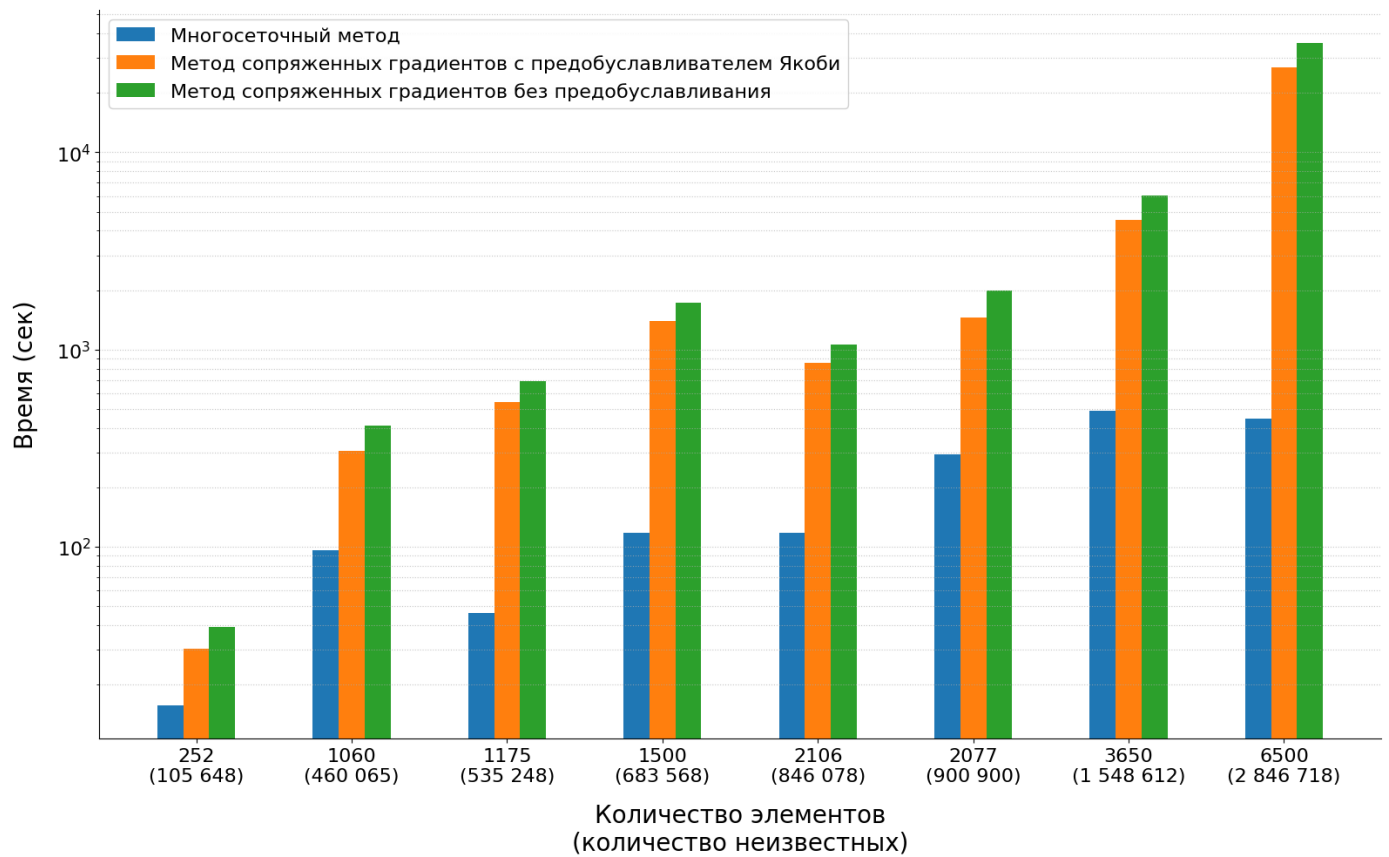


Рис. 16: В сетке спектральные элементы 5 порядка. Многосеточный решатель также построил сетку с элементами 4, 3 и 2 порядков

Можно пронаблюдать за сравнением V-цикла и W-цикла:

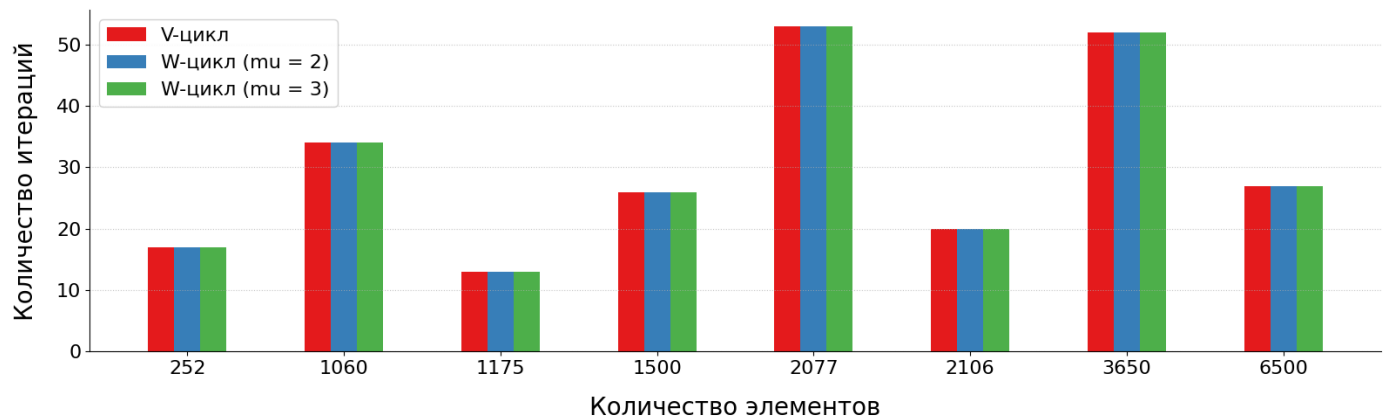


Рис. 17: В сетке спектральные элементы 5 порядка. Три цикла: V, $W(\mu = 2)$, $W(\mu = 3)$, сравнивается количество итераций

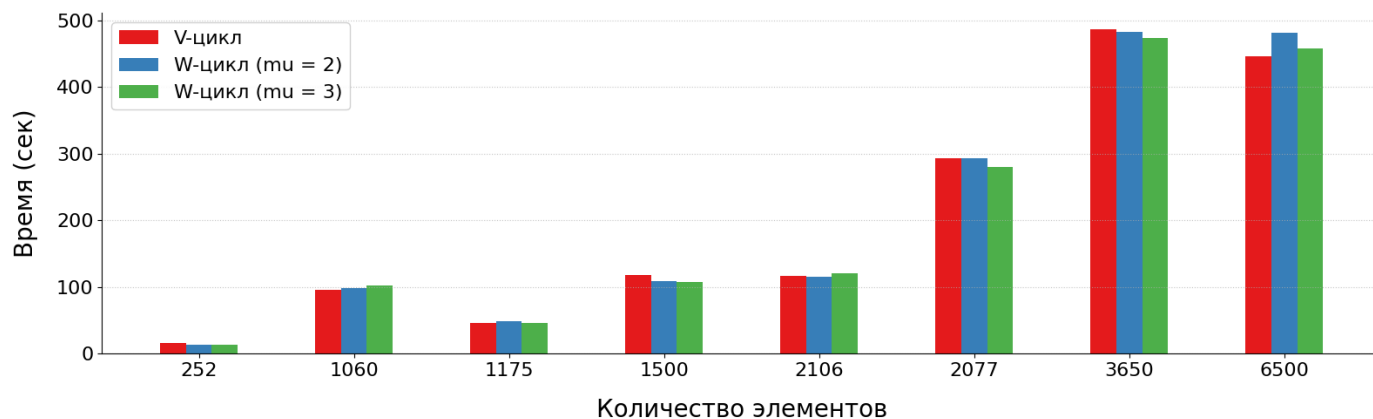


Рис. 18: В сетке спектральные элементы 5 порядка. Три цикла: V, W($\mu = 2$), W($\mu = 3$), сравнивается время решения

Такое поведение обусловлено использованию прямого метода на грубом решателе, на каждой итерации поправка вычисляется точно, последующие уточнения влияют незначительно.

Рассмотрим также количество итераций методов на каждой из 8 задач при повышении порядка:

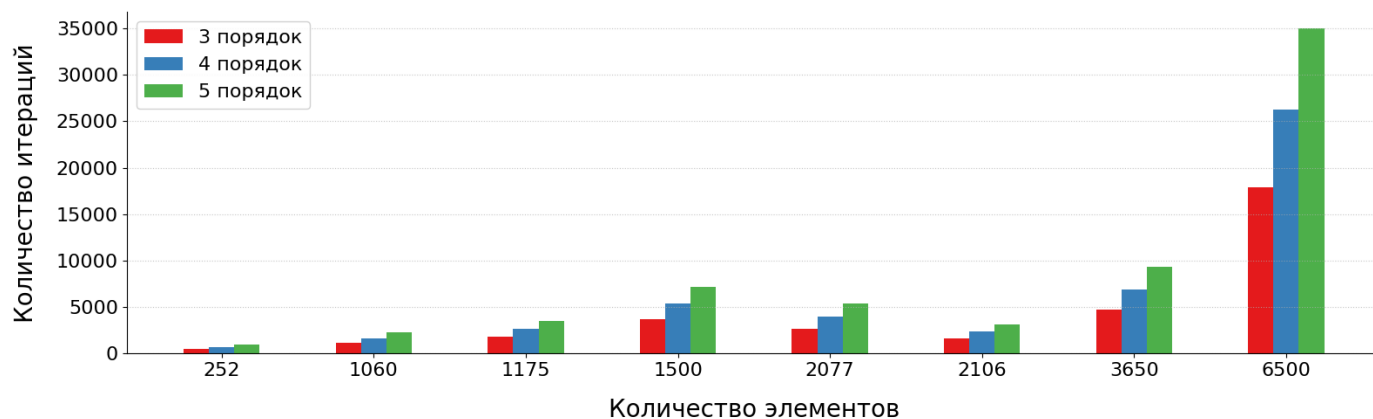


Рис. 19: Метод сопряженных градиентов на каждой из задач при повышении порядка с 3 по 5

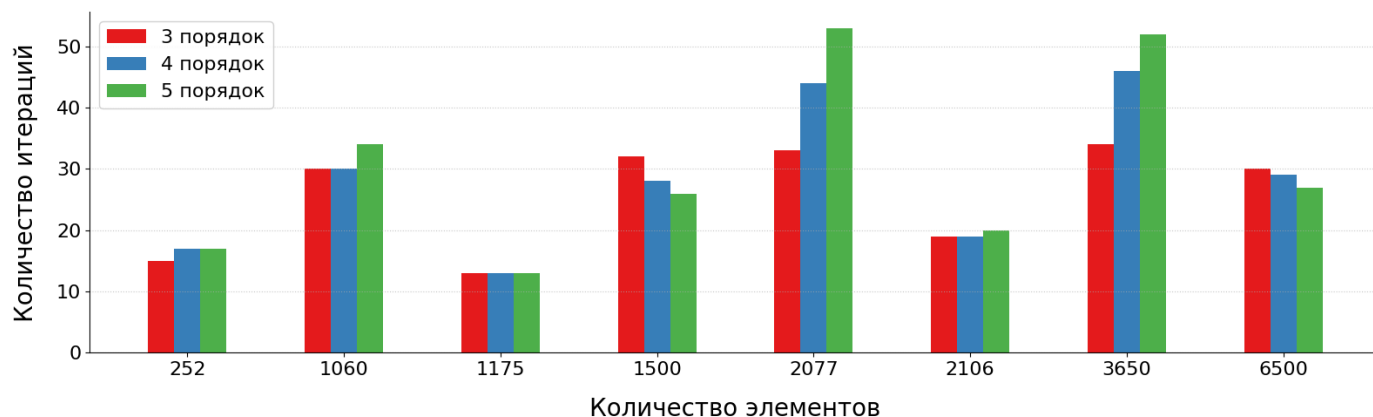


Рис. 20: Многосеточный метод на каждой из задач при повышении порядка с 3 до 5

Многосеточный решатель немного повышает количество требуемых итераций, это может быть связано с неоднозначной трактовкой критерия сходимости по L2 норме вектора U на разных порядках в силу разной размерности вектора. Однако результат вполне согласуется с оценкой скорости сходимости многосеточного метода [16].

6 Заключение

- Создан программный модуль, реализующий многосеточный подход для метода спектральных элементов (р-адаптация).
- Применен более оптимальный формат хранения матрицы жесткости в случае спектральных элементов.
- Многосеточный решатель на крупных задачах (количество неизвестных > 50 тыс.) показал ускорение по сравнению с методом сопряженных градиентов
- Применение W-цикла с прямым решателем избыточно, в будущем можно использовать либо только V-циклы, либо W-цикл с итерационным решателем в качестве грубого.
- Сходимость метода согласована с ее оценкой [16].

Список литературы

- [1] Mark Adams и др. «Parallel multigrid smoothing: polynomial versus Gauss–Seidel». В: *Journal of Computational Physics* 188.2 (2003), с. 593—610. ISSN: 0021-9991. DOI: [https://doi.org/10.1016/S0021-9991\(03\)00194-3](https://doi.org/10.1016/S0021-9991(03)00194-3). URL: <https://www.sciencedirect.com/science/article/pii/S0021999103001943>.
- [2] Caio Ciardelli и др. «SphGLLTools: A toolbox for visualization of large seismic model files based on 3D spectral-element meshes». В: *Computers and Geosciences* 159 (2022), с. 105007. ISSN: 0098-3004. DOI: <https://doi.org/10.1016/j.cageo.2021.105007>. URL: <https://www.sciencedirect.com/science/article/pii/S0098300421002909>.
- [3] David Gottlieb и Steven A. Orszag. *Numerical Analysis of Spectral Methods*. Society for Industrial и Applied Mathematics, 1977. DOI: 10.1137/1.9781611970425. eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611970425>. URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611970425>.
- [4] Yvon Maday и Rafael Munoz. «Spectral element multigrid. II. Theoretical justification». В: *Journal of Scientific Computing* 3.4 (1 дек. 1988), с. 323—353. DOI: 10.1007/BF01065177. URL: <https://doi.org/10.1007/BF01065177>.
- [5] Anthony T Patera. «A spectral element method for fluid dynamics: Laminar flow in a channel expansion». В: *Journal of Computational Physics* 54.3 (1984), с. 468—488. ISSN: 0021-9991. DOI: [https://doi.org/10.1016/0021-9991\(84\)90128-1](https://doi.org/10.1016/0021-9991(84)90128-1). URL: <https://www.sciencedirect.com/science/article/pii/0021999184901281>.
- [6] Einar M. Rønquist и Anthony T. Patera. «Spectral element multigrid. I. Formulation and numerical results». В: *Journal of Scientific Computing* 2.4 (1 дек. 1987), с. 389—406. DOI: 10.1007/BF01061297. URL: <https://doi.org/10.1007/BF01061297>.
- [7] Grigory Sabinin и др. «Full-wave Simulation in Media with Mesoscopic Fractures by the Matrix-free Finite Element Method on GPU». В: (дек. 2023).
- [8] K. Stüben. *Algebraic Multigrid (AMG): An Introduction with Applications ; Updated Version of GMD Report No 53, March 1999*. GMD-Report. GMD-Forschungszentrum Informationstechnik, 1999. URL: <https://books.google.ru/books?id=rzLiGwAACAAJ>.
- [9] Ulrich Trottenberg и Anton Schuller. *Multigrid*. USA: Academic Press, Inc., 2000. ISBN: 012701070X.
- [10] Jinchao Xu и Ludmil T Zikatanov. *Algebraic Multigrid Methods*. 2016. arXiv: 1611.01917 [math.NA]. URL: <https://arxiv.org/abs/1611.01917>.
- [11] Н. С. Бахвалов, Н. П. Жидков и Г. М. Кобельков. *Численные методы*. 10-е изд. Классический университетский учебник. Лаборатория знаний, 2021, с. 636. ISBN: 978-5-93208-224-9.
- [12] О. Зенкевич. *Метод конечных элементов в технике*. Рипол Классик, 1975. ISBN: 9785458376105. URL: <https://books.google.ru/books?id=iFD8AgAAQBAJ>.
- [13] В. А. Левин и А. В. Вершинин. *Численные методы. Параллельные вычисления на ЭВМ*. Под ред. В. А. Левин. Т. 2. Нелинейная вычислительная механика прочности. В 5 т. ФИЗМАТЛИТ, 2015, с. 544. ISBN: 978-5-9221-1632-9.
- [14] Ю. Саад. *Итерационные методы для разреженных линейных систем*. Т. 2. Суперкомпьютерное образование. Издательство Московского университета, 2014. ISBN: 978-5-211-06430-0.
- [15] Л. И. Седов. *Механика сплошной среды*. 6-е изд., стер. Т. 1. Учебник для вузов. Специальная литература. Издательство Лань, 2004, с. 528. ISBN: 5-8114-0541-3.
- [16] Р. П. Федоренко. «О скорости сходимости одного итерационного процесса». В: *Ж. вычисл. матем. и матем. физ.* 4.3 (1964), с. 559—564.
- [17] Р. П. Федоренко. «Релаксационный метод решения разностных эллиптических уравнений». В: *Ж. вычисл. матем. и матем. физ.* 1.5 (1961), с. 922—927.

А Метод сопряженных градиентов и предобуславливание

Приведем также алгоритм итерационного метода широко применяемый для решения СЛАУ с матрицей жесткости - метода сопряженных градиентов. Требование для его применения - симметричность и положительно определенность матрицы. Матрица жесткости отвечает этому требованию (см пункт 3.8).

Алгоритм 7: Метод сопряжённых градиентов

```
1  Функция CG( $K, U_0, F$ )
2   $i := 0$ ;
3   $r_0 := F - KU_0$ ;
4  Пока не удовлетворяет критерию сходимости
5       $\alpha_i := \frac{(r_i, r_i)}{(Kp_i, p_i)}$ ;
6       $U_{i+1} := U_i + \alpha_i p_i$ ;
7       $r_{i+1} := r_i - \alpha_i Kp_i$ ;
8       $\beta_i := \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)}$ ;
9       $p_{i+1} := r_{i+1} + \beta_i p_i$ ;
10      $i := i + 1$ ;
11 возврат  $U_i$ ;
```

Отсутствие устойчивости — недостаток итерационных методов решения систем линейных уравнений по сравнению с прямыми методами. Это ограничивает их применение в промышленных задачах, несмотря на потенциальную эффективность для очень больших систем. Улучшить как скорость, так и надежность итерационных методов позволяет предобуславливание - преобразование исходной системы в эквивалентную, но с лучшими свойствами для итерационного решения.

Исходная система

$$KU = F$$

заменяется на эквивалентную:

$$M^{-1}KU = M^{-1}F$$

Матрица М называется матрицей предобуславливателя: Ключевые требования к М:

- М должна быть легко обратимой
- М должна быть «близка» к исходной матрице
- М обязана быть невырожденной.

Алгоритм метода сопряженных градиентов можно переписать с учетом предобуславливания:

Алгоритм 8: Предобусловленный метод сопряжённых градиентов

```
1   $i := 0$ ;
2   $r_0 := F - KU_0$ ;
3   $z_0 := M^{-1}r_0$ ;
4   $p_0 := z_0$ ;
5  Пока не удовлетворяет критерию сходимости
6       $\alpha_i := \frac{(r_i, z_i)}{(Kp_i, p_i)}$ ;
7       $U_{i+1} := U_i + \alpha_i p_i$ ;
8       $r_{i+1} := r_i - \alpha_i Kp_i$ ;
9       $z_{i+1} := M^{-1}r_{i+1}$ ;
10      $\beta_i := \frac{(r_{i+1}, z_{i+1})}{(r_i, z_i)}$ ;
11      $p_{i+1} := z_{i+1} + \beta_i p_i$ ;
```

В работе используется предобуславливатель Якоби, он использует похожую идею с методом Якоби, используя диагональ матрицы:

$$M = \text{diag}(K)$$

Таким образом, в предобусловленном методе сопряженных градиентов операция:

$$z_{i+1} := M^{-1}r_{i+1}$$

представляет собой покоординатное деление вектора r_{i+1} на диагональ матрицы.

В Листинги

```
{
  "solver": {
    "type": "CG",
    "common": {
      "display_name": "CG",
      "print": true
    },
    "iterative_settings": {
      "convergence_type": "relative",
      "relative_residual": 1e-7,
      "relative_type": "r0",
      "max_iters": 50000,
      "periodic_print": 1
    },
    "preconditioner": {
      "type": "Diagonal"
    }
  }
}
```

Листинг 1: Настройки решателя методом сопряженных градиентов с предобуславливателем Якоби

```
{
  "solver": {
    "type": "CG",
    "common": {
      "display_name": "CG",
      "print": true
    },
    "iterative_settings": {
      "convergence_type": "relative",
      "relative_residual": 1e-7,
      "relative_type": "r0",
      "max_iters": 50000,
      "periodic_print": 1
    }
  }
}
```

Листинг 2: Настройки решателя методом сопряженных градиентов без предобуславливания

```

{
  "solver": {
    "type": "PMG",
    "common": {
      "display_name": "SEM-MG",
      "print": true
    },
    "iterative_settings": {
      "convergence_type": "relative",
      "relative_residual": 1e-7,
      "relative_type": "r0",
      "max_iters": 50000,
      "periodic_print": 1
    },
    "type_specific": {
      "cycle": "W",
      "mu": 1,
      "gamma": 1,
      "restriction": "transpose_interpolation",
      "prolongation": "interpolation",
      "coarsening": {
        "type": "subtraction",
        "subtractor": 1,
        "minimum_order": 2
      },
      "coarsening_rhs": true,
      "coarse_solver": {
        "type": "PARDISO",
        "common": {
          "print": false
        }
      },
      "presmoothen": {
        "type": "Jacobi",
        "common": {
          "print": false
        },
        "iterative_settings": {
          "convergence_type": "iters_only",
          "max_iters": 5
        },
        "type_specific": {
          "relaxation_factor": 0.5,
          "use_diagonal_scaling": false
        }
      },
      "postsmoothen": {
        "type": "Jacobi",
        "common": {
          "print": false
        },
        "iterative_settings": {
          "convergence_type": "iters_only",
          "max_iters": 5
        },
        "type_specific": {
          "relaxation_factor": 0.5,
          "use_diagonal_scaling": false
        }
      }
    }
  }
}

```

Листинг 3: Настройки решателя многосеточным методом