



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Фундаментальные науки»

КАФЕДРА «Прикладная математика»

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
НА ТЕМУ:
РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ
НЕЛОКАЛЬНОЙ УПРУГОСТИ И ИХ
ЧИСЛЕННОЕ РЕШЕНИЕ

Студент группы ФН2-42М

А.А. Соколов

(Подпись, дата)

Руководитель ВКР

Г.Н. Кувыркин

(Подпись, дата)

Нормоконтролер

М.М. Лукашин

(Подпись, дата)

2021 г.

АННОТАЦИЯ

Расчётно-пояснительная записка N с., M рис., K табл., L источников.

Объектом исследования в данной работе выступает двумерное уравнение равновесия в нелокальной постановке. В качестве численного метода решения был выбран метод конечных элементов с использованием изопараметрических конечных элементов.

Цель работы — реализация эффективного конечно-элементного решателя уравнения равновесия в нелокальной постановке с граничными условиями первого и второго родов. Проанализировать полученные результаты и сравнить их с решением аналогичного уравнения в классической постановке.

В работе описан эффективный параллельный и распределённый алгоритм сборки матрицы жёсткости. Выполнена проверка выполнения принципа Сен-Венана, а также проанализированы результаты в Т-образных областях и областях с эллиптическими вырезами.

СОДЕРЖАНИЕ

АННОТАЦИЯ	2
ВВЕДЕНИЕ	4
1. Основные соотношения	5
1.1. Постановка задачи	5
1.2. Выбор функций нелокального влияния	7
1.3. Построение численной схемы решения на основе метода конеч- ных элементов	10
2. Программная реализация	15
2.1. Аппроксимация зоны нелокального влияния	15
2.2. Распараллеливание алгоритма	17
2.3. Балансировка данных	18
3. Результаты расчётов	20
3.1. Результаты распараллеливания	20
3.2. Задача Неймана и проверка принципа Сен-Венана	25
3.3. Расчёт на Т-образной области	29
3.4. Задача Кирша	29
Список использованных источников	30

ВВЕДЕНИЕ

1. Основные соотношения

1.1. Постановка задачи

Дано двумерное евклидово пространство \mathbb{R}^2 с произвольно выбранной прямоугольной декартовой системой координат Ox_1x_2 , в которой положение точки фиксировано радиус-вектором $\mathbf{x} = x_i \mathbf{e}_i$, где \mathbf{e}_i , $i = \overline{1, 2}$ — единичные орты координатных осей; x_i , $i = \overline{1, 2}$ — компоненты вектора \mathbf{x} . В произвольной замкнутой области $S \subset \mathbb{R}^2$ с кусочно-гладкой границей ∂S уравнение равновесия сплошной среды имеет вид [1]

$$\nabla_{\mathbf{x}} \cdot \hat{\boldsymbol{\sigma}} + \mathbf{b} = \mathbf{0}, \quad (1)$$

где $\nabla_{\mathbf{x}} = \partial/\partial x_i \mathbf{e}_i$, $i = \overline{1, 2}$ — дифференциальный оператор набла;

$\mathbf{b} = b_i \mathbf{e}_i$, $i = \overline{1, 2}$ — вектор объёмных сил;

$\hat{\boldsymbol{\sigma}} = \sigma_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, $i, j = \overline{1, 2}$ — тензор напряжений.

Тензор напряжений $\hat{\boldsymbol{\sigma}}$ определим следующим образом

$$\hat{\boldsymbol{\sigma}} = p_1 \hat{\mathbf{C}} \cdot \cdot \hat{\boldsymbol{\varepsilon}} + p_2 \iint_{S'(\mathbf{x}) \cap S} \varphi(\mathbf{x}, \mathbf{x}') \hat{\mathbf{C}} \cdot \cdot \hat{\boldsymbol{\varepsilon}} dS'(\mathbf{x}), \quad (2)$$

где $\hat{\mathbf{C}} = C_{ijkl} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$, $i, j, k, l = \overline{1, 2}$ — тензор коэффициентов упругости;

$\hat{\boldsymbol{\varepsilon}} = \varepsilon_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$, $i, j = \overline{1, 2}$ — тензор упругой деформации;

$p_1 > 0$ и $p_2 \geq 0$ — весовые доли, такие, что $p_1 + p_2 = 1$;

φ — функция нелокального влияния, некоторая нормированная положительная функция в области $S'(\mathbf{x})$;

$S'(\mathbf{x})$ — область нелокального влияния;

$$\mathbf{x}' \in S'(\mathbf{x}).$$

Будем считать, что деформации малы, поэтому для определения компонентов тензора деформации $\hat{\boldsymbol{\varepsilon}}$ воспользуемся соотношением Коши [1]

$$\varepsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}), \quad i, j = \overline{1, 2}, \quad (3)$$

где u_i , $i = \overline{1, 2}$ — компоненты вектора перемещений \mathbf{u} .

В случае линейного упругого изотропного тела для задачи в плосконапряжённой постановке компоненты тензора упругости $\hat{\mathbf{C}}$ определяются следующим образом

$$C_{ijkl} = \frac{\nu E}{1 - \nu^2} \delta_{ij} \delta_{kl} + \frac{E}{2(1 + \nu)} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}), \quad i, j, k, l = \overline{1, 2},$$

где E — модуль Юнга;

ν — коэффициент Пуассона.

Будем рассматривать граничные условия первого и второго родов, также именуемые кинематическими и силовыми соответственно [1]

$$\mathbf{u}|_{\Gamma_1} = \mathbf{d}(\mathbf{x}), \quad \hat{\boldsymbol{\sigma}} \cdot \mathbf{n}|_{\Gamma_2} = \mathbf{p}(\mathbf{x}),$$

где $\Gamma_1, \Gamma_2 \subset \partial S$ и $\Gamma_1 \cup \Gamma_2 = \emptyset$;

$\mathbf{d}(\mathbf{x}) = d_i(\mathbf{x}) \mathbf{e}_i$ — некоторая функция, задающая перемещение на границе Γ_1 ;

$\mathbf{p}(\mathbf{x}) = p_i(\mathbf{x}) \mathbf{e}_i$, $i = \overline{1, 2}$ — некоторая функция, задающая давление на границе Γ_2 .

1.2. Выбор функций нелокального влияния

В определении тензора напряжений (2) не конкретизируется геометрия области нелокального влияния $S'(\mathbf{x})$ и вид функции нелокального влияния φ . Чаще всего в расчётах используют функцию нормального распределения Гаусса, а зону нелокального влияния аппроксимируют в виде круга по правилу трёх сигм, но полученные при такой аппроксимации зоны получаются достаточно большими, что накладывает серьёзные требования на объёмы используемой оперативной памяти. Вместе с тем, вычисление экспонент является весьма затратным с вычислительной точки зрения, поэтому предлагается использовать весьма широкий класс полиномиальных функций нелокального влияния с фиксированными областями нелокального влияния $S'(\mathbf{x})$, которые можно представить в виде

$$\varphi(\mathbf{x}, \mathbf{x}') = \begin{cases} A(1 - \rho(\mathbf{x}, \mathbf{x}')^p)^q, & \rho(\mathbf{x}, \mathbf{x}') \leq 1, \\ 0, & \rho(\mathbf{x}, \mathbf{x}') > 1, \end{cases} \quad (4)$$

где ρ — метрическая функция, порождающая область $S'(\mathbf{x})$, которую в общем случае можно определить следующим образом

$$\rho(\mathbf{x}, \mathbf{x}') = \sqrt[n]{\left| \frac{x_1 - x'_1}{r_1} \right|^n + \left| \frac{x_2 - x'_2}{r_2} \right|^n}.$$

Тогда нормировочный множитель A примет значение

$$A = \frac{pn}{4r_1r_2B(1/n, 1/n)B(2/p, q+1)},$$

где $r_1, r_2 > 0$ — радиусы нелокального влияния вдоль каждой оси;

$n > 0$ — параметр лебегова пространства;

$p, q > 0$ — параметры плотности распределения нелокального влияния;

B — бета-функция Эйлера.

Вид зоны нелокального влияния $S'(\mathbf{x})$ при различных параметрах лебегова пространства n и фиксированных радиусах влияния проиллюстрированы на рис. 1.

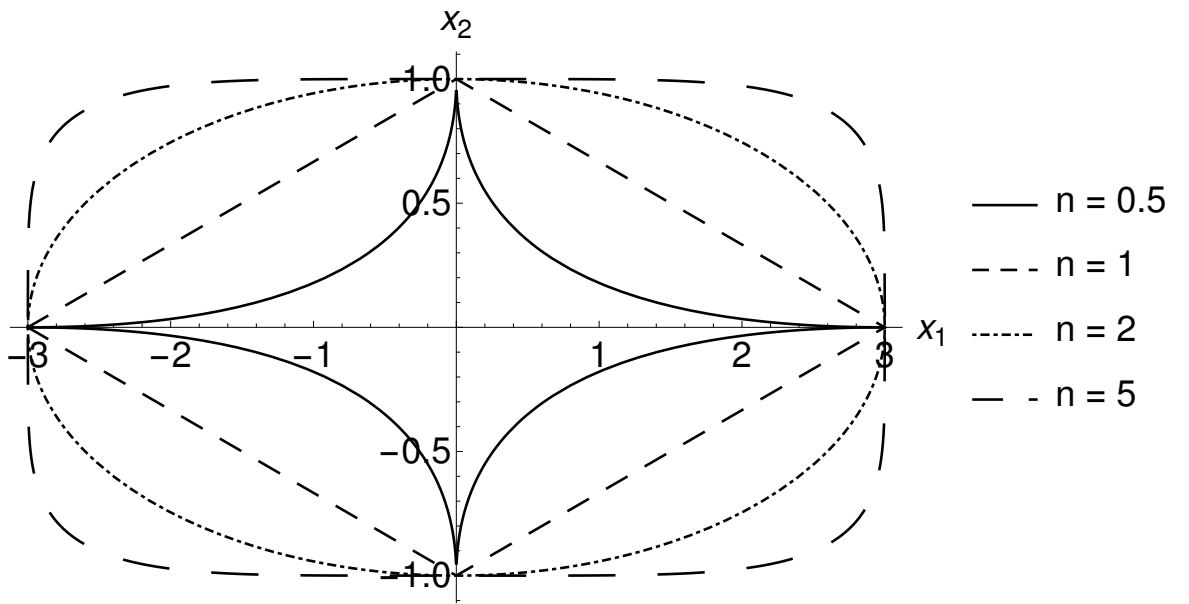


Рис. 1. Вид зоны нелокального влияния при $r_1 = 3$, $r_2 = 1$ и различных параметрах n

Предполагается, что изучаемые тела изотропны, поэтому будем считать, что область нелокального влияния представляет из себя круг, то есть $r_1 = r_2 = r$, а параметр $n = 2$. Что касается параметров плотности распределения влияния, то стоит отметить, что при увеличении параметра p распределение влияния становится более равномерным и стремится к константе, что в теории должно приводить к увеличению отклонения от классического закона. При увеличении параметра q распределение влияния концентрируется в центре области и стремится к дельта-функции Дирака, которая при подстановке в (2) даст нам классический закон Гука. На рис. 2 и 3 представ-

лены примеры функций влияния в разрезе по оси симметрии при различных параметрах p и q . Для простоты и наглядности будем считать параметры p и q целыми и введём обозначение $\varphi_{p,q}$, которое даст понять какие параметры были выбраны.

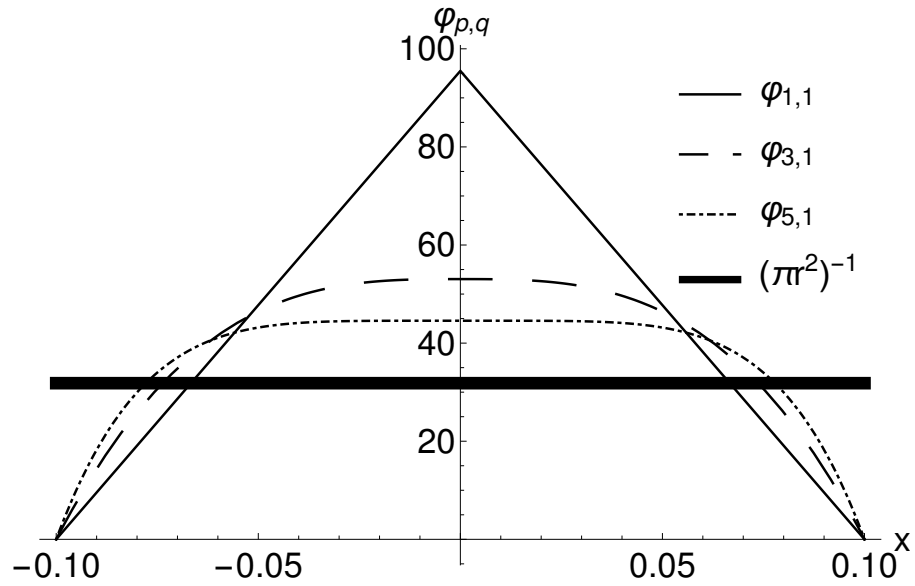


Рис. 2. Портреты функций влияния при вариации p

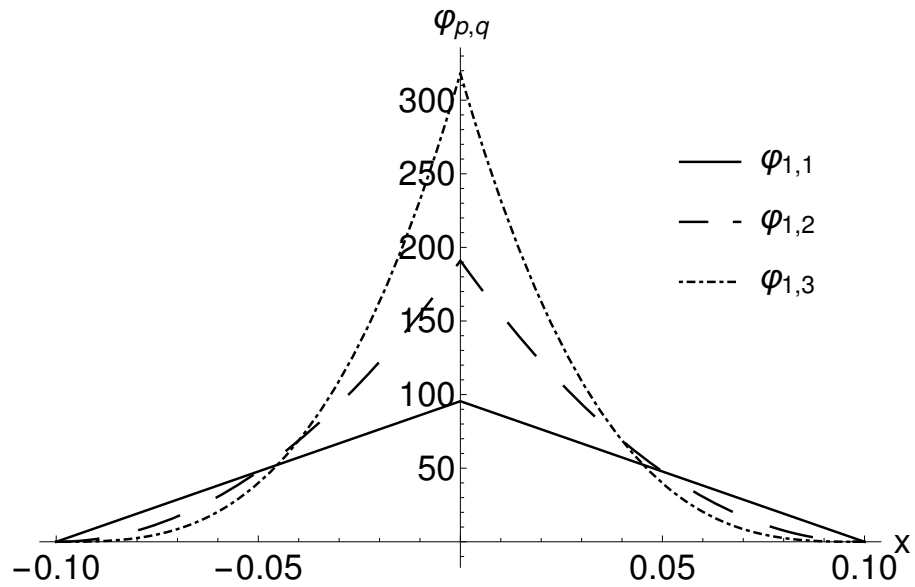


Рис. 3. Портреты функций влияния при вариации q

1.3. Построение численной схемы решения на основе метода конечных элементов

В качестве численного метода решения уравнения (1) выберем метод конечных элементов с использованием изопараметрических конечных элементов [2], [3]. Для этого на области S введём сетку конечно-элементной модели S_h , которая включает в себя множества номеров узлов и элементов. Каждый элемент $(e) \in S_h$ содержит в себе наборы узлов $\{\mathbf{x}_i\}_{i \in I^{(e)}}$ и базисных функций $\{N_i^{(e)}\}_{i \in I^{(e)}}$ таких, что

$$N_i^{(e)}(\mathbf{x}_j) = \delta_{ij}, \quad i, j \in I^{(e)},$$

$$\sum_{i \in I^{(e)}} N_i^{(e)}(\mathbf{x}) = 1, \quad \forall \mathbf{x} \in S^{(e)},$$

где $I^{(e)}$ — набор индексов узлов элемента (e) ;

δ_{ij} — дельта-функция Кронекера;

$S^{(e)}$ — область элемента (e) .

Для каждого конечного элемента (e) введём локальную систему координат $O\xi_1^{(e)}\xi_2^{(e)}$. Отображение из локальной системы координат $O\xi_1^{(e)}\xi_2^{(e)}$ в глобальную Ox_1x_2 будем строить следующим образом

$$\mathbf{x}(\boldsymbol{\xi}^{(e)}) = N_i^{(e)}(\boldsymbol{\xi}^{(e)}) \mathbf{x}_i, \quad i \in I^{(e)}, \quad (e) \in S_h,$$

где \mathbf{x}_i — значение глобальных координат в узлах сетки. Тогда матрицу Якоби перехода из локальной системы координат в глобальную аппроксимируем следующим образом

$$\hat{\mathbf{j}}^{(e)} = \left(\frac{\partial \boldsymbol{\xi}^{(e)}}{\partial \mathbf{x}} \right) = \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}^{(e)}} \right)^{-1} \approx \left(\mathbf{x}_i \frac{\partial N_i^{(e)}}{\partial \boldsymbol{\xi}^{(e)}} \right)^{-1}, \quad i \in I^{(e)}, \quad (e) \in S_h.$$

Домножим уравнение (1) на функцию формы $N_n^{(e)}$ и проинтегрируем по области S

$$\iint_S N_n^{(e)} (\nabla_{\mathbf{x}} \cdot \hat{\boldsymbol{\sigma}} + \mathbf{b}) dS = \mathbf{0}, \quad n \in I^{(e)}, \quad (e) \in S_h.$$

Проинтегрируем первое слагаемое по частям, тогда по формуле Грина получим

$$\iint_S (\nabla_{\mathbf{x}} N_n^{(e)}) \cdot \hat{\boldsymbol{\sigma}} dS = \oint_{\Gamma_2} N_n^{(e)} \mathbf{p} d\Gamma - \iint_S N_n^{(e)} \mathbf{b} dS, \quad n \in I^{(e)}, \quad (e) \in S_h.$$

На место $\hat{\boldsymbol{\sigma}}$ подставим соотношение (2)

$$\begin{aligned} p_1 \iint_S (\nabla_{\mathbf{x}} N_n^{(e)}) \cdot (\hat{\mathbf{C}} \cdot \cdot \hat{\boldsymbol{\varepsilon}}) dS + \\ + p_2 \iint_S (\nabla_{\mathbf{x}} N_n^{(e)}) \cdot \iint_{S'(\mathbf{x}) \cap S} \varphi(\mathbf{x}, \mathbf{x}') \hat{\mathbf{C}} \cdot \cdot \hat{\boldsymbol{\varepsilon}} dS'(\mathbf{x}) dS = \\ = \oint_{\Gamma_2} N_n^{(e)} \mathbf{p} d\Gamma - \iint_S N_n^{(e)} \mathbf{b} dS, \quad n \in I^{(e)}, \quad (e) \in S_h. \end{aligned}$$

Перейдём к индексной форме записи

$$\begin{aligned}
& p_1 \iint_S N_{n,i}^{(e)} C_{ijkl} \varepsilon_{kl} dS + p_2 \iint_S N_{n,i}^{(e)} \iint_{S'(\mathbf{x}) \cap S} \varphi(\mathbf{x}, \mathbf{x}') C_{ijkl} \varepsilon_{kl} dS'(\mathbf{x}) dS = \\
& = \oint_{\Gamma_2} N_n^{(e)} p_j d\Gamma - \iint_S N_n^{(e)} b_j dS, \quad i, j, k, l = \overline{1, 2}, \quad n \in I^{(e)}, \quad (e) \in S_h.
\end{aligned}$$

Воспользуемся соотношением Коши (3) и заменим перемещения \mathbf{u} интерполяционными соотношениями, то есть представим в виде конечной суммы $\mathbf{u}(\mathbf{x}) \approx \mathbf{u}_m N_m^{(e)}(\mathbf{x})$, $m \in I^{(e)}$, где \mathbf{u}_m – искомые перемещения в m -ом узле. Также для простоты дальнейшего изложения, введём тензор $\mathbf{K}_{nm}^{(e)(e')}$, который по сути представляет из себя блок 2×2 в n -ой строке и m -ом столбце матрицы жёсткости $\hat{\mathbf{K}}$. Компоненты тензора $\mathbf{K}_{nm}^{(e)(e')}$ можно определить следующим образом

$$K_{nmi,j}^{(e)(e')}(\mathbf{x}, \mathbf{y}) = \delta_{np} \delta_{mq} C_{ikjl} N_{n,k}^{(e)}(\mathbf{x}) N_{m,l}^{(e')}(\mathbf{y}) \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{E}_p \otimes \mathbf{E}_q,$$

где \mathbf{E}_i — единичный вектор, размерность которого совпадает с количеством узлов сетки.

Зону нелокального влияния $S'(\mathbf{x})$ аппроксимируем относительно каждого квадратурного узла сетки и будем считать, что элементы (e) , квадратурные узлы $Q^{(e)}$ которых хотя бы частично попали в область влияния, будут участвовать в расчёте. Иллюстрация такого способа аппроксимации наглядно продемонстрирована на рис. 4, где квадратурные узлы помечены точками, а квадратурный узел относительного которого происходит аппроксимация помечен крестом. Аппроксимированная зона выделена серым цветом.

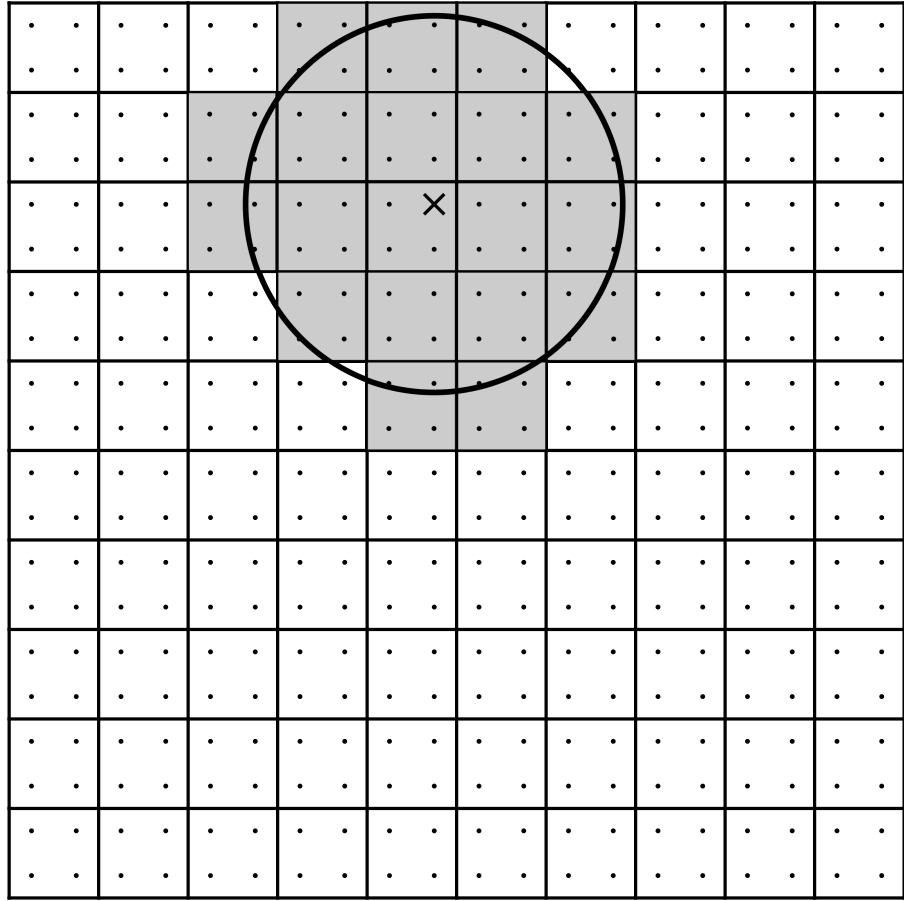


Рис. 4. Аппроксимация зоны нелокального влияния относительно квадратурного узла сетки

Таким образом аппроксимируем интегралы конечными суммами и наконец приходим к равенству с ассамблированной матрицей жёсткости и правой частью

$$\begin{aligned}
 & \sum_{(e) \in S_h} \sum_{n \in I^{(e)}} \left(p_1 \sum_{m \in I^{(e)}} \sum_{q \in Q^{(e)}} w_q \mathbf{K}_{nm}^{(e)(e)}(\mathbf{x}_q, \mathbf{x}_q) J_q^{(e)} + \right. \\
 & \left. + p_2 \sum_{q \in Q^{(e)}} w_q J_q^{(e)} \sum_{(e') \in S_h^{tq}} \sum_{m' \in I^{(e')}} \sum_{q' \in Q^{(e')}} w_{q'} \varphi(\mathbf{x}_q, \mathbf{x}_{q'}) \mathbf{K}_{nm'}^{(e)(e')}(\mathbf{x}_q, \mathbf{x}_{q'}) J_{q'}^{(e')} \right) \cdot \mathbf{U} =
 \end{aligned}$$

$$\begin{aligned}
&= \sum_{(be) \in \Gamma_h} \sum_{n \in I^{(be)}} \mathbf{E}_n \otimes \sum_{q \in Q^{(be)}} w_q N_n^{(be)}(\mathbf{x}_q) \mathbf{p}(\mathbf{x}_q) J_q^{(be)} - \\
&\quad - \sum_{(e) \in S_h} \sum_{n \in I^{(e)}} \mathbf{E}_n \otimes \sum_{q \in Q^{(e)}} w_q N_n^{(e)}(\mathbf{x}_q) \mathbf{b}(\mathbf{x}_q) J_q^{(e)}, \quad (5)
\end{aligned}$$

где $\mathbf{U} = \delta_{nj} \mathbf{E}_n \otimes \mathbf{u}_j$ — тензор неизвестных;

$S_h^{(q)} \subset S_h$ — аппроксимированная зона нелокального влияния на квадратурном узле q ;

$\Gamma_h \subset S_h$ — конечно-элементная модель границы области S ;

$(be) \in \Gamma_h$ — одномерные элементы заданные на границе;

w_q — квадратурные веса;

$J_q^{(e)} = \left| \det \hat{\mathbf{J}}^{(e)}(\mathbf{x}_q) \right|$ — якобиан вычисленный в квадратурной точке \mathbf{x}_q .

Вычисление системы (5) можно упростить записав производные от функций форм в локальной системе координат элементов

$$\frac{\partial N_i^{(e)}}{\partial x_k} = \frac{\partial N_i^{(e)}}{\partial \xi_j^{(e)}} \frac{\partial \xi_j^{(e)}}{\partial x_k}, \quad j, k = \overline{1, 2}, \quad i \in I^{(e)}, \quad (e) \in S_h.$$

Якобиан на границе аппроксимируем следующим образом

$$J^{(be)} = \sqrt{\left(x_{1i} \frac{\partial N_i^{(e)}}{\partial \xi^{(e)}} \right)^2 + \left(x_{2i} \frac{\partial N_i^{(e)}}{\partial \xi^{(e)}} \right)^2}, \quad i \in I^{(be)}, \quad (be) \in \Gamma_h.$$

Заметим, что индекс оси $\xi^{(e)}$ не ставится, так как элемент одномерный.

2. Программная реализация

2.1. Аппроксимация зоны нелокального влияния

Основной сложностью, которая возникает при решении задачи (1), является аппроксимация интегрального слагаемого в (2), так как в матрицах элементов необходимо учитывать влияние, которое приходится на соседние элементы. Действуя по правилам аппроксимации вложенных интегралов мы приходим к тому, что в уравнении (5) порядок сумм заставляет нас аппроксимировать зону нелокального влияния для каждого квадратурного узла сетки, что не очень практично, так как выполнять поиск ближайших соседей относительно квадратурных узлов может быть весьма затратной процедурой, а хранение индексов элементов, которые попадают в зону влияния, очень дорогим. Более того, такой подход попросту сложно реализовать программно, поэтому возникает предложение аппроксимировать зону нелокального влияния относительно центров элементов и учитывать в расчётах те элементы, центры которых попали в зону влияния. Такой способ аппроксимации проиллюстрирован на рис. 5, где центры элементов помечены точками, текущий элемент относительно которого проводится аппроксимация помечен крестом, а аппроксимированная зона закрашена серым цветом. Тогда формулу (5) можно переписать в следующем виде

$$\begin{aligned} & \sum_{(e) \in S_h} \sum_{n \in I^{(e)}} \left(p_1 \sum_{m \in I^{(e)}} \sum_{q \in Q^{(e)}} w_q \mathbf{K}_{nm}^{(e)(e)}(\mathbf{x}_q, \mathbf{x}_q) J_q^{(e)} + \right. \\ & \left. + p_2 \sum_{(e') \in S_h^{tq}} \sum_{m' \in I^{(e')}} \sum_{q \in Q^{(e)}} w_q J_q^{(e)} \sum_{q' \in Q^{(e')}} w_{q'} \varphi(\mathbf{x}_q, \mathbf{x}_{q'}) \mathbf{K}_{nm'}^{(e)(e')}(\mathbf{x}_q, \mathbf{x}_{q'}) J_{q'}^{(e')} \right) \cdot \mathbf{U} = \end{aligned}$$

$$\begin{aligned}
&= \sum_{(be) \in \Gamma_h} \sum_{n \in I^{(be)}} \mathbf{E}_n \otimes \sum_{q \in Q^{(be)}} w_q N_n^{(be)}(\mathbf{x}_q) \mathbf{p}(\mathbf{x}_q) J_q^{(be)} - \\
&\quad - \sum_{(e) \in S_h} \sum_{n \in I^{(e)}} \mathbf{E}_n \otimes \sum_{q \in Q^{(e)}} w_q N_n^{(e)}(\mathbf{x}_q) \mathbf{b}(\mathbf{x}_q) J_q^{(e)}, \quad (6)
\end{aligned}$$

где $S_h'^{(e)}$ — зона нелокального влияния аппроксимированная на элементе.

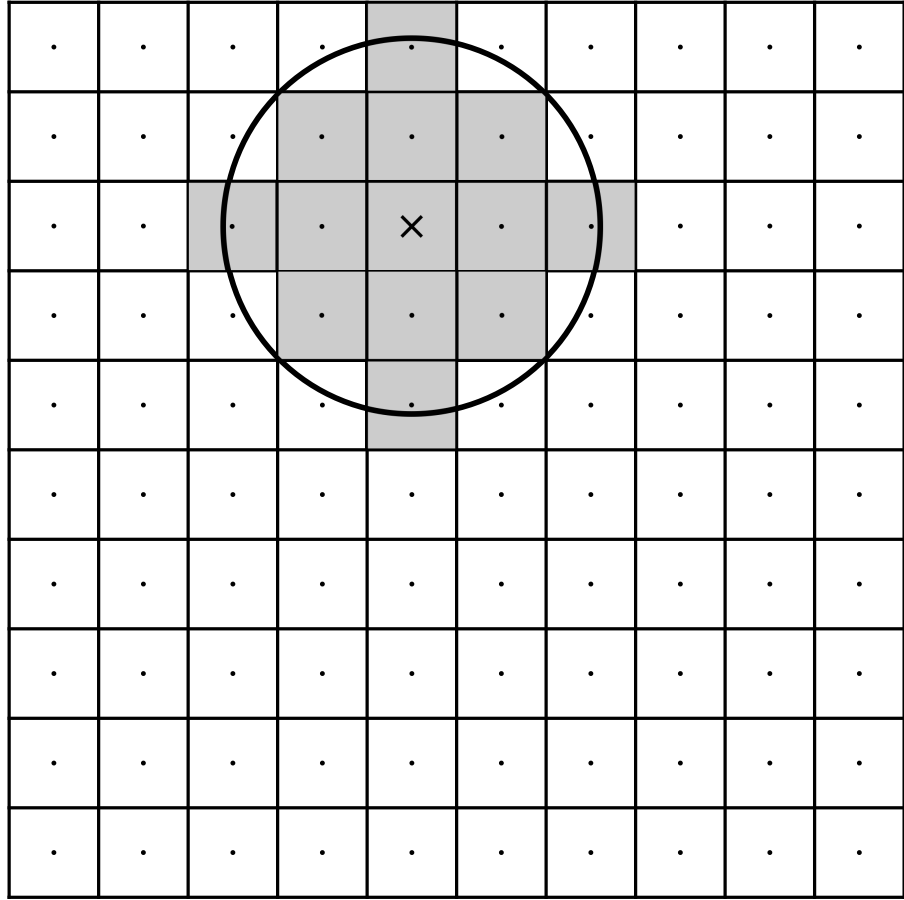


Рис. 5. Аппроксимация зоны нелокального влияния

Заметим, что в формуле (6) поменялся порядок знаков суммирования, а значит и порядок действий. С точки зрения написания вычислительной программы это означает лишь перестановку циклов и введение дополнительных массивов индексов. Для успешного интегрирования, зону аппроксимации лучше всего брать несколько больше, нежели реальная зона нелокального влияния, таким образом возникает шанс того, что все квадратурные узлы, которые попали под зону влияния $S'(\mathbf{x})$ будут учтены в расчёте. Естествен-

но, радиус поиска следует определять исходя из грубости сетки, что наглядно продемонстрировано на рис. 5.

2.2. Распараллеливание алгоритма

Учитывая высокую вычислительную сложность подобных задач разумно использовать возможности современных вычислительных систем, а именно параллельные и распределённые вычисления. К сожалению, обход сетки, который задаётся порядком суммирования в (6) не позволяет эффективно использовать такие возможности, поэтому возникает предложение определить другой порядок обхода сетки, в котором не будет проблем связанных с гонкой данных и который позволит собирать части матрицы независимо сразу в нескольких процессах. Этого можно достичь определив для каждого узла сетки множество элементов $E^{(n)}$, которым он принадлежит. Такой подход позволяет эффективно использовать возможности параллельных и распределённых систем, так как каждая строка матрицы вычисляется независимо от всех остальных. Заметим, что порядок суммирования при аппроксимации граничных условий второго рода менять не обязательно, так как данная операция не является трудозатратной и поэтому её проще всего реализовать согласно классическому представлению. Таким образом, формула (6) принимает вид

$$\begin{aligned} & \sum_{n \in S_h} \sum_{(e) \in E^{(n)}} \left(p_1 \sum_{m \in I^{(e)}} \sum_{q \in Q^{(e)}} w_q \mathbf{K}_{nm}^{(e)(e)}(\mathbf{x}_q, \mathbf{x}_q) J_q^{(e)} + \right. \\ & \left. + p_2 \sum_{(e') \in S_h'^q} \sum_{m' \in I^{(e')}} \sum_{q \in Q^{(e)}} w_q J_q^{(e)} \sum_{q' \in Q^{(e')}} w_{q'} \varphi(\mathbf{x}_q, \mathbf{x}_{q'}) \mathbf{K}_{nm'}^{(e)(e')}(\mathbf{x}_q, \mathbf{x}_{q'}) J_{q'}^{(e')} \right) \cdot \mathbf{U} = \end{aligned}$$

$$\begin{aligned}
&= \sum_{(be) \in \Gamma_h} \sum_{n \in I^{(be)}} \mathbf{E}_n \otimes \sum_{q \in Q^{(be)}} w_q N_n^{(be)}(\mathbf{x}_q) \mathbf{p}(\mathbf{x}_q) J_q^{(be)} - \\
&\quad - \sum_{n \in S_h} \mathbf{E}_n \otimes \sum_{(e) \in E^{(n)}} \sum_{q \in Q^{(e)}} w_q N_n^{(e)}(\mathbf{x}_q) \mathbf{b}(\mathbf{x}_q) J_q^{(e)}, \quad (7)
\end{aligned}$$

2.3. Балансировка данных

Как уже было сказано, поузловой обход сетки (7) позволяет достаточно эффективно использовать параллельные и распределённые вычисления. Однако при распределённых вычислениях возникает проблема балансировки данных между процессами, так как если раздать обработку узлов сетки между процессами равномерно, то данные могут иметь произвольное распределение, что может привести к слишком высоким затратам вычислительных ресурсов на некоторых узлах кластера, в то время как остальным придётся проводить время в ожидании. Такой дисбаланс также может привести к тому, что на некоторых узлах кластера может не хватить оперативной памяти, хотя потенциально такая задача могла бы поместиться при равномерном распределении данных. В таком случае необходимо делать балансировку данных между процессами.

Идеальная балансировка подразумевает распределение данных таким образом, чтобы потребляемые объёмы оперативной памяти и объёмы вычислений на всех процессах были одинаковыми, но практика показывает, что добиться такой балансировки в общем случае не удаётся, поэтому опишем два возможных варианта балансировки.

Первый вариант позволяет добиться равномерного распределения объёмов потребляемой оперативной памяти между процессами P . Этот вариант балансировки подразумевает под собой осреднение количества элемен-

тов матрицы между процессами $p \in P$. Чтобы этого добиться, необходимо подсчитать количество элементов матрицы M_p , которым владеет каждый из процессов. Затем сложить все эти суммы $M = \sum_{p \in P} M_p$ и взять среднее по количеству процессов $M_m = M/|P|$. После этого распределить узлы таким образом, чтобы количество элементов матрицы на каждом из процессов \widetilde{M}_p было примерно одинаковым, то есть $\widetilde{M}_p \approx M_m$. Для подсчёта количества элементов в матрице не требуется формировать полный портрет матрицы на каждом процессе, это можно делать построчно, что гораздо эффективнее и не приводит к высоким затратам оперативной памяти.

Второй вариант балансировки позволяет добиться равномерного распределения объёмов вычислений. Такой вариант балансировки достигается аналогичными методами, но в качестве осредняемого параметра выбирается количество вызовов функции интегрирования.

3. Результаты расчётов

3.1. Результаты распараллеливания

Проведём серию расчётов на системе K-10 [4], где на каждом узле кластера установлено по два процессора Intel Xeon E5-2660 и по 128 Гб оперативной памяти. В качестве тестовой задачи возьмём задачу на области $S = [0, 1] \times [0, 1]$, с введённой на ней равномерной сеткой S_h из квадратичных серендиповых элементов. В качестве функции нелокального влияния выберем $\varphi = \varphi_{2,1}$, так как данная функция требует наименьшее количество вычислений, но при этом обеспечивает уменьшение влияния на расстоянии. Матрицы, которые получаются в расчётах, имеют симметричную и разреженную структуру, поэтому для удобства будем хранить их в CSR формате. Для индексации будем использовать 32-х битные числа, а для значений коэффициентов будем использовать числа с плавающей точкой двойной точности.

Расчёты показывают, что объёмы требуемой оперативной памяти и время вычислений t_1 растут линейно относительно количества элементов сетки в классической постановке задачи и квадратично в нелокальной. Причём в нелокальной также наблюдается квадратичный рост относительно радиуса поиска ближайших соседей, так как среднее число соседей растёт в тех же пропорциях. Результаты расчётов представлены в табл. 1. В таблице также представлено время счёта на 16 потоках. Из таблицы видно, что для задач в классической постановке не удаётся достичь существенного ускорения времени счёта, но для нелокальных задач время счёта ускорилось примерно в 13.5 раз. Результаты ускорения наглядно продемонстрированы на рис. 6.

При использовании параллельных и распределённых вычислений одновременно удаётся достичь ещё большего ускорения сборки матрицы жёсткости. В качестве демонстрации возможностей распределённых вычислений будем использовать 4 узла кластера и запустим на каждом процессы с 16

потоками. Случай с равномерным распределением узлов между процессами опустим и сразу рассмотрим случаи с балансировками, которые были описаны ранее.

Количество элементов	Радиус поиска соседей	Среднее число соседей	Требуемый объём памяти	Время расчёта 1 поток	Время, расчёта 16 потоков
2500	0	—	3 Мб	0.065 с	0.019 с
10000	0	—	12 Мб	0.296 с	0.065 с
40000	0	—	46 Мб	1.490 с	0.277 с
2500	0.05	20	19 Мб	3.583 с	0.321 с
10000	0.05	69	203 Мб	48.82 с	3.773 с
40000	0.05	295	2.9 Гб	815.5 с	60.45 с
2500	0.1	66	48 Мб	11.45 с	0.854 с
10000	0.1	281	687 Мб	193.3 с	13.90 с
40000	0.1	1143	10.2 Гб	3161 с	222.5 с

Таблица 1. Сравнение последовательного и параллельного режимов работы

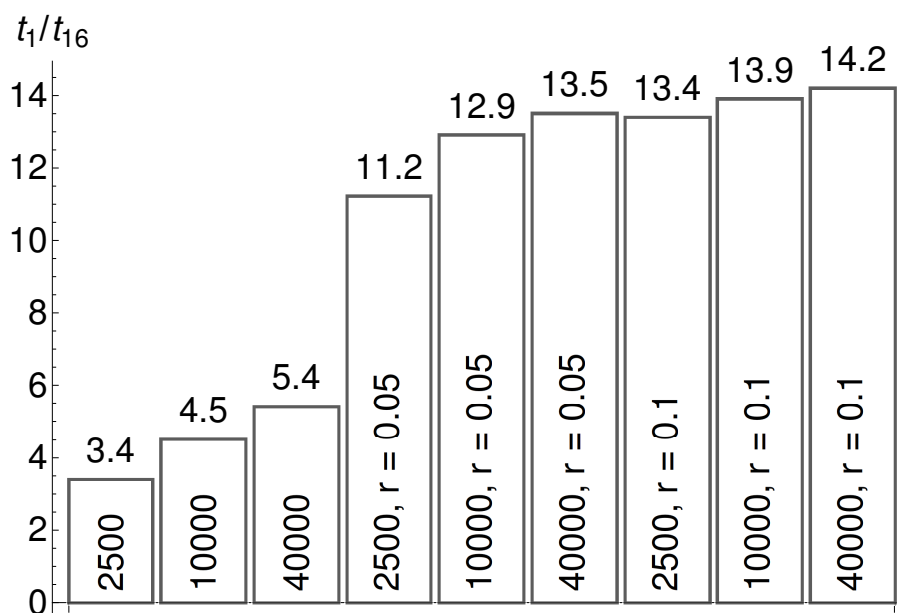


Рис. 6. Ускорение времени сборки матрицы жёсткости при использовании технологии OpenMP

В случае балансировки объёмов вычислений получаем заметное ускорение времени счёта, но при этом в распределении объёмов затрачиваемой оперативной памяти наблюдается сильный дисбаланс. Исходя из результатов, которые представлены в табл. 2, на четвёртом процессе объёмы используемой оперативной памяти в два раза выше, чем у всех остальных процессов, хотя при этом, время счёта у всех процессов примерно одинаковое. Такое, на первый взгляд, странное распределение можно объяснить тем, что в процессах, где объёмы используемой оперативной памяти меньше, количество самопересекающихся индексов в расчёте коэффициентов больше. Распределение

N	r	V_1	$t_{16}^1, \text{с}$	V_2	$t_{16}^2, \text{с}$	V_3	$t_{16}^3, \text{с}$	V_4	$t_{16}^4, \text{с}$
10000	0.05	42 Мб	0.934	40 Мб	0.868	39 Мб	0.890	82 Мб	0.974
10000	0.1	136 Мб	3.775	130 Мб	3.396	130 Мб	3.670	291 Мб	3.775
40000	0.05	558 Мб	15.16	545 Мб	14.19	544 Мб	14.55	1.2 Гб	15.74
40000	0.1	1.9 Гб	55.41	1.9 Гб	54.80	1.9 Гб	57.47	4.5 Гб	59.91

Таблица 2. Распределение объёмов данных и времени выполнения между процессами при балансировке объёмов вычислений

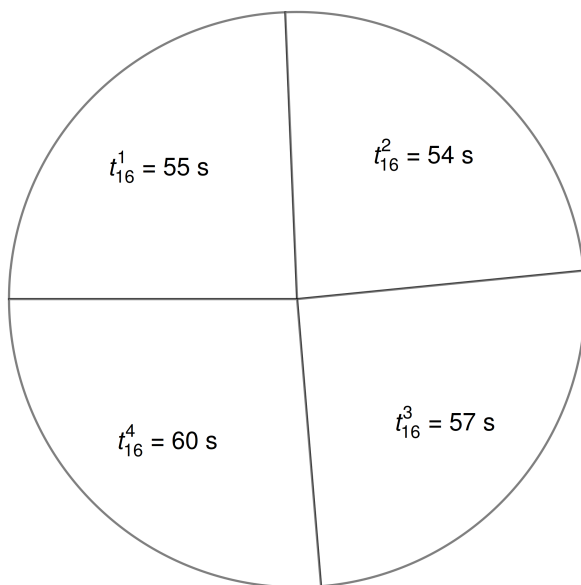


Рис. 7. Распределение времени счёта между процессами при балансировке объёмов вычислений

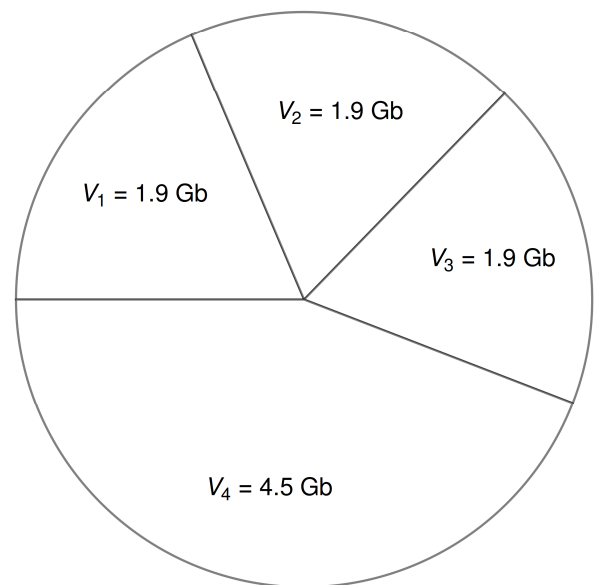


Рис. 8. Распределение объёмов памяти между процессами при балансировке объёмов вычислений

объёмов оперативной памяти и времени счёта наглядно продемонстрировано на рис. 7 и 8 на примере задачи с 40000 элементами и радиусом нелокального влияния $r = 0.1$.

N	r	V_1	$t_{16}^1, \text{с}$	V_2	$t_{16}^2, \text{с}$	V_3	$t_{16}^3, \text{с}$	V_4	$t_{16}^4, \text{с}$
10000	0.05	50 Мб	1.133	51 Мб	1.119	51 Мб	0.917	51 Мб	0.603
10000	0.1	168 Мб	4.545	172 Мб	4.505	175 Мб	3.212	172 Мб	2.236
40000	0.05	711 Мб	19.12	716 Мб	18.68	723 Мб	12.40	716 Мб	9.311
40000	0.1	2.5 Гб	72.66	2.6 Гб	72.46	2.5 Гб	45.91	2.6 Гб	34.72

Таблица 3. Распределение объёмов данных и времени выполнения между процессами при балансировке количества элементов матрицы

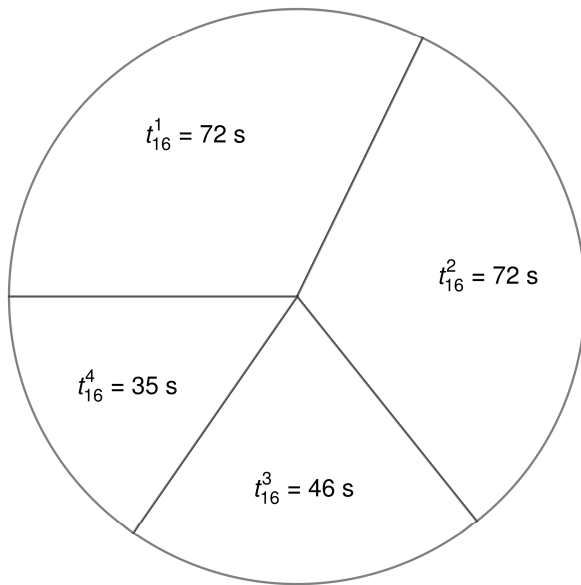


Рис. 9. Распределение времени счёта между процессами при балансировке количества элементов матрицы

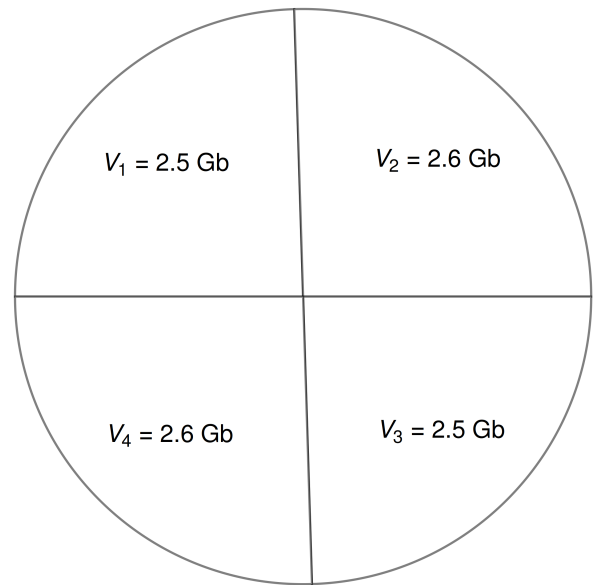


Рис. 10. Распределение объёмов памяти между процессами при балансировке количества элементов матрицы

Решением проблемы дисбаланса в использовании оперативной памяти является балансировка количества элементов матрицы между процессами. При такой балансировке, данные распределяются равномерно, но при этом неравномерным становится время вычисления коэффициентов, что так же

объясняется количеством самопересекающихся индексов в расчёте коэффициентов матрицы. Результаты расчётов представлены в табл. 3. Как и в случае балансировки объёмов вычислений на примере задачи с 40000 элементов и радиусом нелокального влияния $r = 0.1$ покажем в виде диаграмм распределение времени вычислений и объёмов данных между процессами. Результаты представлены на рис. 9 и 10.

Подытожим результаты использования распределённых вычислений диаграммой представленной на рис. 11. Из данной диаграммы видно, что в целом, при использовании любой из балансировок наблюдается ускорение времени счёта при использовании нескольких процессов. Однако стоит отметить, что при балансировке объёмов вычислений время счёта ускоряется сильнее. Данный разрыв может быть больше, например, при другой нумерации узлов или принципиально другой геометрии области S . В любом случае, результаты показывают высокую эффективность и целесообразность использования такого вида вычислений.

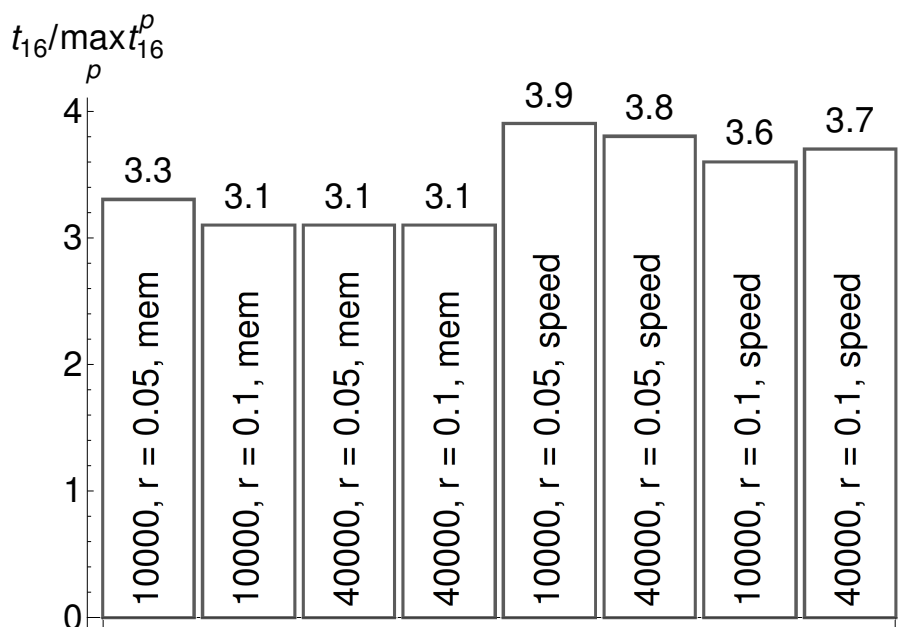


Рис. 11. Ускорение времени сборки матрицы жёсткости при использовании технологии MPI

3.2. Задача Неймана и проверка принципа Сен-Венана

Далее, во всех расчётах будем использовать безразмерные параметры и переменные

$$\bar{\sigma} = \frac{\sigma}{p}, \quad \bar{x} = \frac{x}{L}, \quad \bar{u} = \frac{Eu}{(1 - \nu^2)p},$$

где L — характерный размер области; p — нормировочный множитель для нагружений; черта над переменными означает, что они безразмерны. Примем коэффициент Пуассона $\nu = 0.3$, а безразмерный модуль Юнга $\bar{E} = 21$. Будем считать, что если не указано иного, функция нелокального влияния $\varphi = \varphi_{2,1}$.

При исследовании нелокальных моделей упругости важно опираться на принципы классической теории упругости. Одним из таких принципов является принцип Сен-Венана, согласно которому различные, но статически эквивалентные нагрузки вызывают в стержне одинаковые напряженные состояния вдали от точек приложения [1]. Данное явление не доказано в общем случае, но подтверждается многими теоретическими и экспериментальными исследованиями. Для проверки применимости принципа Сен-Венана к нелокальным задачам упругости рассмотрим задачу Неймана на области $S = [0, 10] \times [0, 1]$, с введённой на ней равномерной сеткой S_h , состоящей из 64000 квадратичных серендиповых элементов. Поставим следующие граничные условия

$$\bar{\sigma} \cdot \mathbf{n}|_{\bar{x}_1=0} = -f(\bar{x}_2), \quad \bar{\sigma} \cdot \mathbf{n}|_{\bar{x}_1=10} = f(\bar{x}_2), \quad \bar{\sigma} \cdot \mathbf{n}|_{\bar{x}_2=0,1} = 0,$$

где f — некоторая функция нагружения. Для получения единственности решения поставленной задачи, добавим два геометрических условия, которые

так же нам дадут симметрию решения и зафиксируют его относительно центра области по каждой из осей

$$\bar{u}_1|_{\bar{x}_1=5} = 0, \quad \bar{u}_2|_{\bar{x}_2=0.5} = 0.$$

В классической постановке задачи, в силу симметрии, можно было бы обойтись рассмотрением лишь четверти области, но в нелокальной постановке приходится учитывать поиск ближайших соседей, который не допускает рассмотрения части области S .

Рассмотрим три случая нагружения с одинаковой нормировкой на области $A = [0, 1]$

$$f_1(x) = 1,$$

$$f_2(x) = \begin{cases} 4x, & x \leq 0.5, \\ 4 - 4x, & x > 0.5, \end{cases}$$

$$f_3(x) = \begin{cases} 2 - 4x, & x \leq 0.5, \\ 4x - 2, & x > 0.5. \end{cases}$$

На рис. 12 и 13 изображены напряжения $\bar{\sigma}_{11}$ вдоль оси нагружения в различных сечениях. Несмотря на различный характер нагружений, в центре области напряжения принимают одинаковую форму, которая в нелокальной

постановке, в отличие от классической, не является константой. Расчётные параметры и выбранные сечения указаны на рисунках.

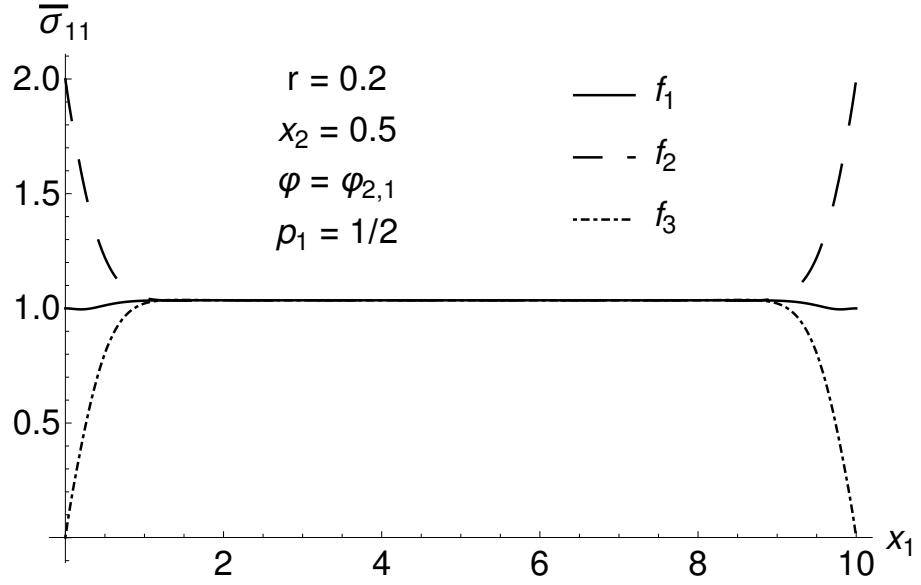


Рис. 12. Распределение напряжений $\bar{\sigma}_{11}$ в сечении $\bar{x}_2 = 0.5$ при различных нагружениях.

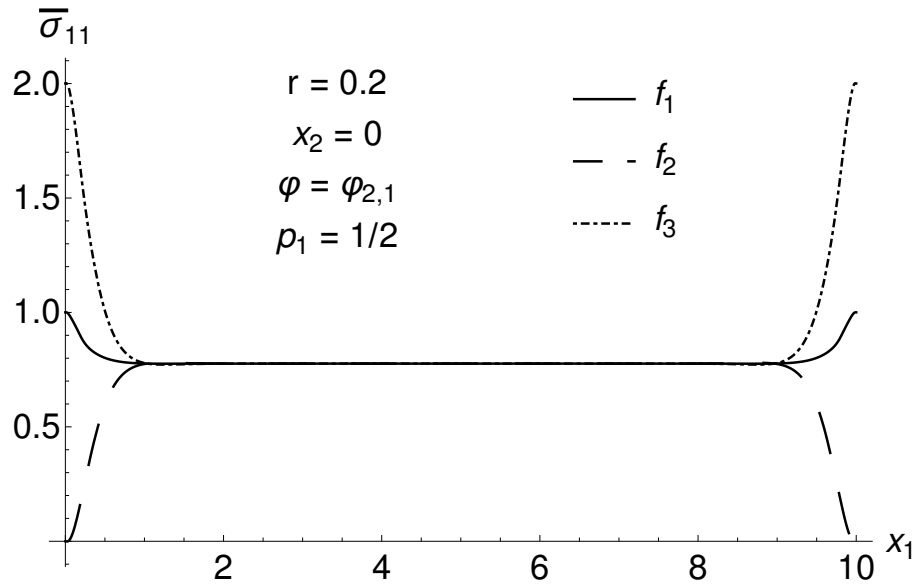


Рис. 13. Распределение напряжений $\bar{\sigma}_{11}$ в сечении $\bar{x}_2 = 0$ при различных нагружениях.

Рассматривая центральное сечение вдоль другой оси, можем убедиться в том, что в нелокальной постановке поверхность напряжений имеет более сложную форму. Вариация весового параметра p_1 влияет на отклонение

напряжений на границе, в то время как вариация радиуса нелокального влияния r влияет на размах краевого эффекта. Результаты расчётов представлена на рис. 14 и 15.

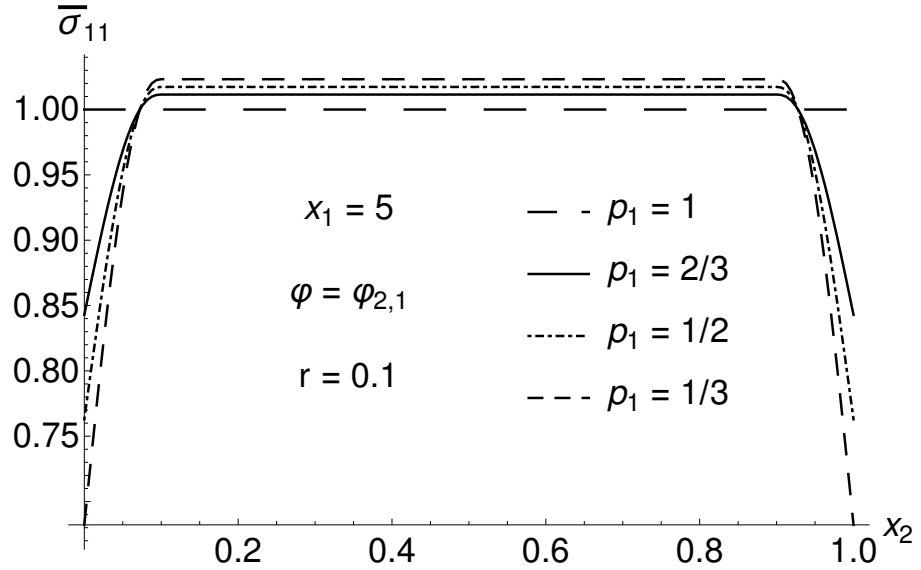


Рис. 14. Распределение напряжений $\bar{\sigma}_{11}$ в сечении $\bar{x}_1 = 5$ при вариации p_1

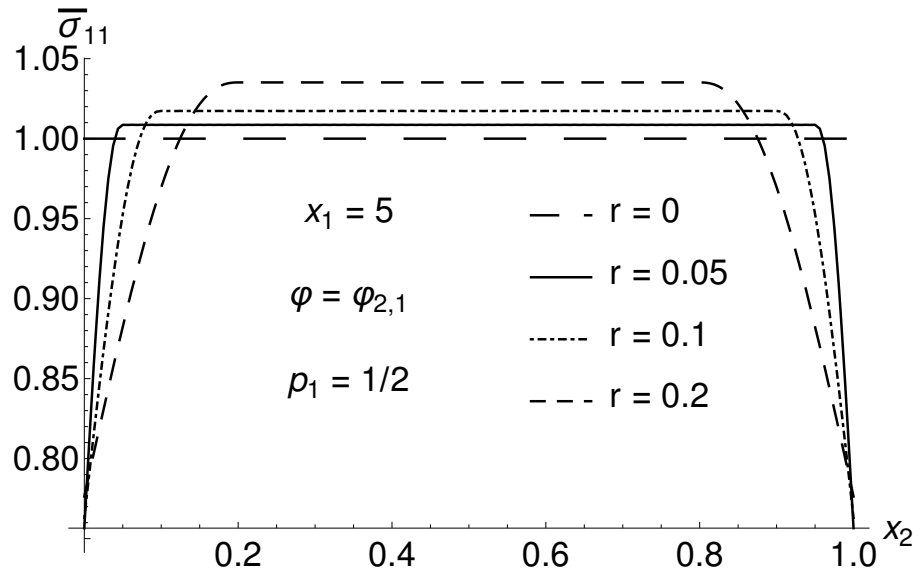


Рис. 15. Распределение напряжений $\bar{\sigma}_{11}$ в сечении $\bar{x}_1 = 5$ при вариации r

Стоит отметить, что в отличие от классической постановки, в нелокальной также возникают касательные напряжения $\bar{\sigma}_{12}$. Их характер имеет весьма сложную, но симметричную структуру. На рис. 16 и 17 изображены кривые в сечениях отстоящих от границы на величину радиуса нелокального влияния при различных весовых параметрах.

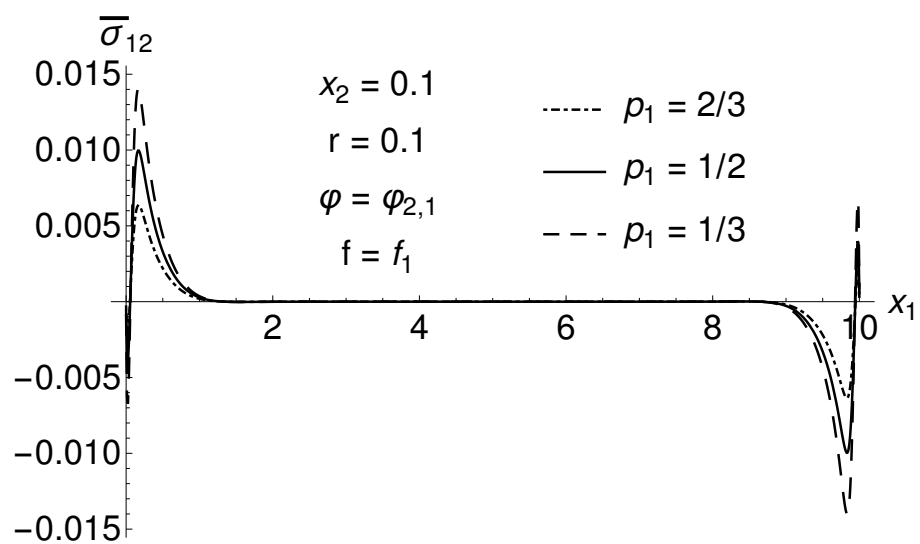


Рис. 16. Распределение напряжений $\bar{\sigma}_{12}$ в сечении $\bar{x}_2 = 0.1$

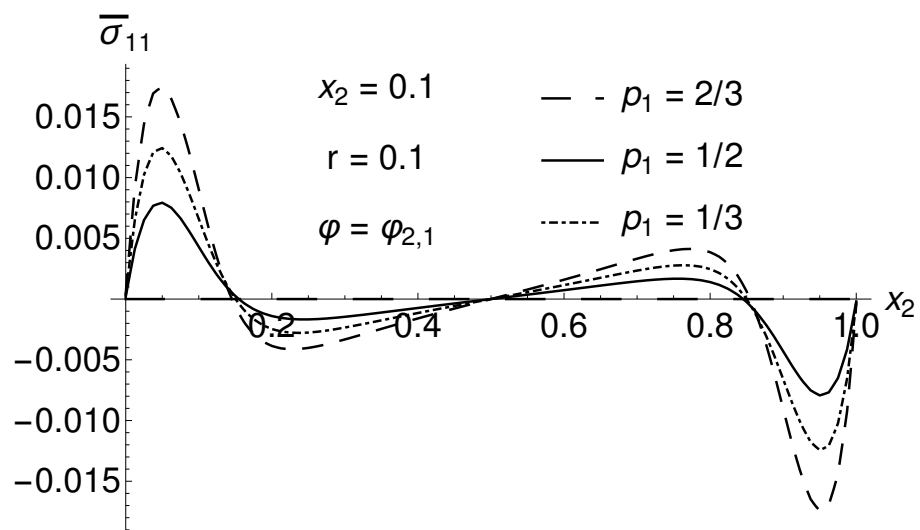


Рис. 17. Распределение напряжений $\bar{\sigma}_{12}$ в сечении $\bar{x}_1 = 0.1$

3.3. Расчёт на Т-образной области

3.4. Задача Кирша

Список использованных источников

1. Зарубин В.С., Кувыркин Г.Н. Математические модели механики и электродинамики сплошной среды. М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. 512 с.
2. Zienkiewicz O. The Finite Element Method: Its Basis and Fundamentals. Seventh edition. / O. Zienkiewicz, R. Taylor, J.Z. Zhu – 2013. – 756 p.
3. Bathe K-J. Finite Element Procedures. Second edition. — 2014. — 1065 p.
4. <https://www.kiam.ru/MVS/resources/k10.html> (Дата обращения 31.03.2021)