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 ТЕОРИИ НЕЛОКАЛЬНОЙ УПРУГОСТИ И ИХ ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ И АНАЛИЗ

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 58 с., 47 рис., 4 табл., 23 источника.

В качестве объекта исследования в 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 уравнений равновесия в нелокальной постановке с граничными условиями первого и второго родов, анализ полученных результатов и сравнение их с решением аналогичных уравнений в классической постановке.

В работе изложена постановка задачи и описан параллельный и распределённый алгоритм сборки матрицы жёсткости с использованием технологий OpenMP и MPI. Также были предложены различные варианты балансировки данных между процессами. Проанализирована эффективность и масштабируемость полученного алгоритма на примере результатов расчётов на гибридном вычислительном кластере К-10. В работе описаны особенности связанные с выполнением принципа Сен-Венана на примере вытянутой прямоугольной пластины. Проведено исследование поведения решений в областях со ступенчатыми переходами, а также в областях с эллиптическими вырезами. Полученные результаты были сравнены с результатами решения уравнения в классической постановке. Результаты для областей со ступенчатыми переходами также были сравнены с результатами экспериментов. Для областей с эллиптическими вырезами была проанализирована зависимость максимальных напряжений от соотношения полуосей эллипсов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Современное машиностроение ставит высокие требования к физико-механическим свойствам материалов из которых должны быть изготовлены детали конструкций или их покрытия. Как правило, такими материалами становятся различные структурно-чувствительные материалы, которые получают при помощи компактирования нанопорошков, осаждением на подложку, кристализацией аморфных сплавов и другими способами [1].

Использование классических математических моделей для структурно-чувствительных материалов некорректно, так как классические модели не учитывают масштабных эффектов, которые в телах малых размеров становятся сопоставимыми с размерами самого тела и начинают оказывать существенное влияние. Существует множество различных моделей, которые описывают подобные эффекты, однако наибольший интерес представляют модели, в которых основные соотношения имеют ту же формулировку, что и классические. Одной из таких моделей является нелокальной модель Эрингена [2], где определяющие соотношения имеют тот же вид, что и в классическом подходе, но представлены в интегро-дифференциальном форме. Такая формулировка открывает возможности для использования хорошо изученных численных методов, в частности метода конечных элементов.

В данной работе рассмотрено решение двумерного уравнения равновесия в нелокальной постановке. Ранее, подобные уравнения в двумерных постановка были рассмотрены в работах [3], [4], [5] и многих других. Основной сложностью, с которой сталкиваются исследователи при решении подобных задач, это аппроксимация интегро-дифференциальных уравнений. В частности, в методе конечных элементов, который применительно к данным задачам принято ещё называть методом нелокальных конечных элементов, приходится выполнять поиск ближайших соседей, что приводит к некоторым дополнительным расходам при расчётах и усложняет структуру вычислительной

программы.

Стоит отметить, что схемы решения такого рода уравнений отличаются высокой вычислительной сложностью, так как итоговые СЛАУ имеют высокую плотность заполненности и эта плотность стремится к константе при дроблении сетки. То есть, даже несмотря на использование разреженных и симметрических матриц, не всегда удаётся достичь существенной экономии ресурсов. Прямые методы решения такого рода СЛАУ сталкиваются с проблемами нехватки оперативной памяти, а итерационные имеют медленную сходимость в виду большого количества ненулевых элементов в матрице. Однако скорость сходимости итерационных методов можно ускорить если воспользоваться альтернативными базисами конечных элементов [6], [7], подробности исследования которых описаны в приложении А. Вместе с колоссальными объёмами требуемой оперативной памяти также существует проблема большого объёма вычислений, которая частично решается современными параллельными и распределёнными вычислительными системами.

Особым интересом исследования является проверка классических принципов и гипотез. В частности, одним из таких принципов является принцип Сен-Венана [8], проверка которого представлена в разделе с результатами рассчётов. Также особый интерес представляют области с концентраторами, так как они являются типичными проблемами механики разрушения из-за сингулярных деформаций и напряжений, которые в них возникают. В работе пойдёт речь об областях со ступенчатыми переходами и эллиптическими вырезами, так как они имеют достаточно простую структуру, а также различные теоретические и экспериментальные оценки и результаты.

1. Основные соотношения

1.1. Постановка задачи

Дано двумерное евклидово пространство $\mathbb{R}^2$ с произвольно выбранной прямоугольной декартовой системой координат $O\mathrm{x}_1\mathrm{x}_2$, в которой положение точки фиксировано радиус-вектором $\boldsymbol{x} = x_i\boldsymbol{e}_i$, где $\boldsymbol{e}_i$, $i = \overline{1,2}$ — единичные орты координатных осей; x_i, $i = \overline{1,2}$ — компоненты вектора $\boldsymbol{x}$. В произвольной замкнутой области $S \subset \mathbb{R}^2$ с кусочно-гладкой границей ∂S уравнение равновесия сплошной среды имеет вид [8]

$$\nabla \cdot \widehat{\boldsymbol{\sigma}} + \boldsymbol{b} = \boldsymbol{0}, \tag{1}$$

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

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

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

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

$$\widehat{\boldsymbol{\sigma}} = p_1\widehat{\mathrm{C}} \cdot\cdot \widehat{\boldsymbol{\varepsilon}} + p_2 \iint\limits_{S'(\boldsymbol{x})\cap S} \varphi(\boldsymbol{x},\boldsymbol{x}')\widehat{\mathrm{C}} \cdot\cdot \widehat{\boldsymbol{\varepsilon}} dS'(\boldsymbol{x}), \quad \boldsymbol{x'} \in S'(\boldsymbol{x}), \tag{2}$$

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

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

$p_1 > 0$ и $p_2 \geqslant 0$ — весовые доли классического и нелокального законов, соответственно, такие, что $p_1 + p_2 = 1$;

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

$S'(\boldsymbol{x})$ — область нелокального влияния.

Положим, что деформации малы, поэтому для определения компонент

тензора деформации $\widehat{\boldsymbol{\varepsilon}}$ воспользуемся соотношением Коши [8]

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

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

В случае линейного упругого изотропного тела для задачи в плоском напряжённом состоянии компоненты тензора упругости $\widehat{\mathrm{C}}$ определяют 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 [8],

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

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

$\boldsymbol{d}(\boldsymbol{x}) = d_i(\boldsymbol{x})\boldsymbol{e}_i$, $i = \overline{1,2}$ — некоторая функция, задающая перемещения на границе Γ_1;

$\boldsymbol{p}(\boldsymbol{x}) = p_i(\boldsymbol{x})\boldsymbol{e}_i$, $i = \overline{1,2}$ — некоторая функция, задающая вектор плотности поверхностной нагрузки на границе Γ_2;

$\boldsymbol{n}$ — внешняя нормаль области S.

1.2. Выбор функций нелокального влияния

В определении тензора напряжений (2) не конкретизирована геометрия области нелокального влияния $S'(\boldsymbol{x})$ 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 приводит к большим затратам используемой оперативной памяти. Вместе с тем, вычисление экспонент является весьма трудоёмкой процедурой, поэтому возникает предложение использовать весьма широкий класс полиномиальных функций нелокального влияния с фиксированными областями этого влияния $S'(\boldsymbol{x})$, которые можно представить в виде

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

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

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

Тогда нормировочный множитель A примет значение

$$A = \frac{pn}{4r_1r_2\mathrm{B}(1/n,1/n)\mathrm{B}(2/p,q+1)},$$

где $r_1, r_2 > 0$ — длины полуосей области $S'(\boldsymbol{x})$;

$n > 0$ — параметр;

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

$\mathrm{B}(x, y) = \int\limits_0^1 t^{x-1}(1-t)^{y-1}dt$ — бета-функция Эйлера [9].

Форма зоны нелокального влияния $S'(\boldsymbol{x})$, при различных параметрах n и фиксированных длинах полуосей r_1 и r_2, проиллюстрированы на рис. 1.

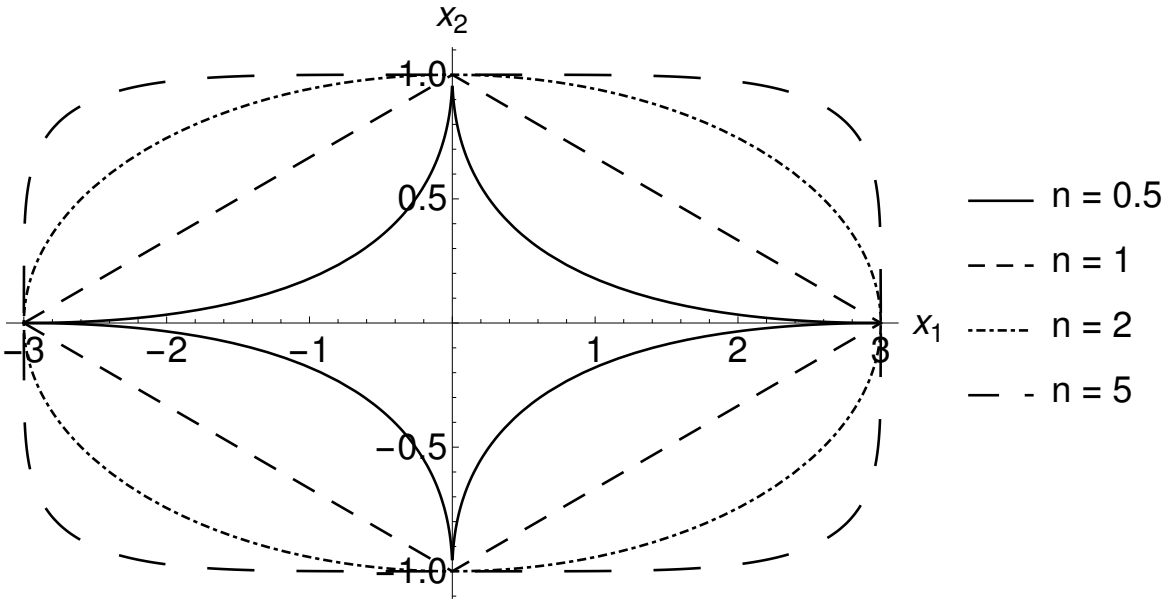


Рис. 1. Формы зоны нелокального влияния при различных параметрах n и радиусах $r_1 = 3$, $r_2 = 1$

Предположим, что рассматриваемые материалы изотропны, поэтому будем полагать, что область нелокального влияния является кругом, то есть $r_1 = r_2 = r$, а параметр $n = 2$. Далее в работе мы будем варьировать параметры плотности распределения влияния p и q. Будем считать, что эти параметры целые, поэтому, для простоты изложения, суженный класс функций нелокального влияния будем обозначить как $\varphi_{p,q}$. На рис. 2 и 3 представлены примеры функций нелокального влияния в разрезе по оси симметрии при различных параметрах p и q. На рисунках видно, что при увеличении параметра p, распределение влияния становится равномернее и стремится к константе $(\pi r^2)^{-1}$, что в теории должно приводить к увеличению отклонения решения нелокальной задачи от аналогичного решения классической. При увеличении параметра q распределение концентрируется в центре области и стремится к

дельта-функции Дирака [12], которая, при постановке в (2), даст нам классический закон Гука. Смешанное варьирование параметро p и q рассматривать не будем.

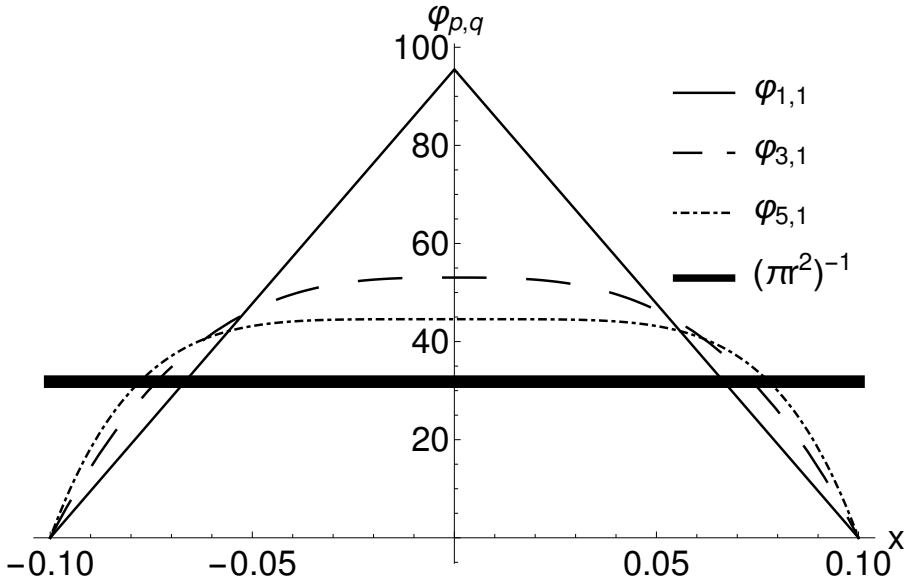


Рис. 2. Портреты функций влияния при вариации p

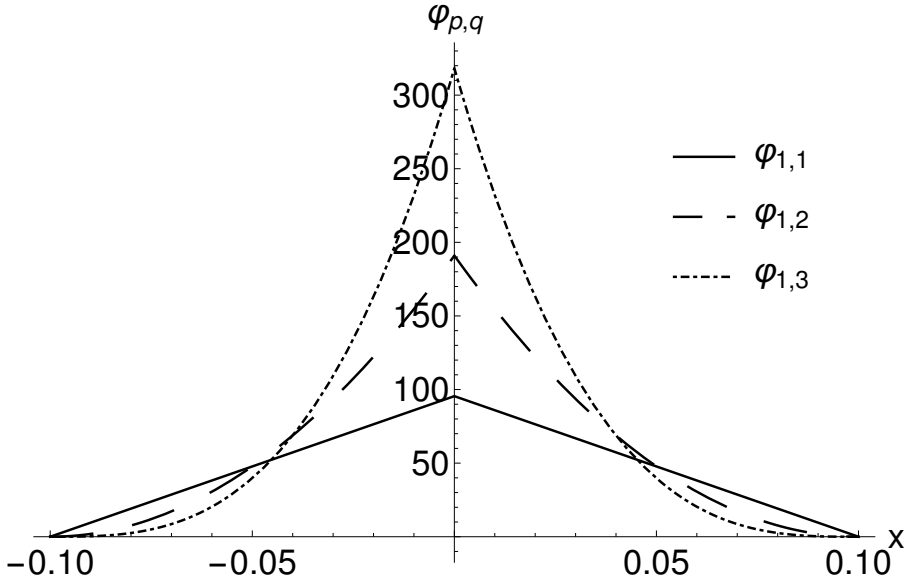


Рис. 3. Портреты функций влияния при вариации q

Для практических расчётов следует использовать функции с наименьшими вычислительными затратами, но в то же время обеспечивающими уменьшение влияния на расстоянии. В классе полиномиальных функций это функция $\varphi_{2,1}$, которую мы будем использовать во всех расчётах, если не сказано иного.

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 [10], [11]. 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 $\{\boldsymbol{x}_i\}_{i \in I^{(e)}}$ и базисных функций $\left\{N_i^{(e)}\right\}_{i \in I^{(e)}}$ таких, что

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

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

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

δ_{ij} — дельта-функция Кронекера [12];

$S^{(e)}$ — 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 $O\mathrm{x}_1\mathrm{x}_2$ будем строить следующим образом:

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

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

следующим образом

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

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

$$\iint\limits_S N_n^{(e)}(\nabla \cdot \widehat{\boldsymbol{\sigma}} + \boldsymbol{b})dS = \mathbf{0}, \quad n \in I^{(e)},\ (e) \in S_h.$$

Проинтегрируем первое слагаемое по частям, тогда по формуле Грина приходим к уравнению вида

$$\iint\limits_S (\nabla N_n^{(e)}) \cdot \widehat{\boldsymbol{\sigma}} dS = \oint\limits_{\Gamma_2} N_n^{(e)} \boldsymbol{p} d\Gamma - \iint\limits_S N_n^{(e)} \boldsymbol{b} dS, \quad n \in I^{(e)},\ (e) \in S_h.$$

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

$$p_1 \iint\limits_S \left(\nabla N_n^{(e)}\right) \cdot \left(\widehat{\mathbf{C}} \cdot\cdot \widehat{\boldsymbol{\varepsilon}}\right) dS +$$

$$+ p_2 \iint\limits_S \left(\nabla N_n^{(e)}\right) \cdot \iint\limits_{S'(\boldsymbol{x}) \cap S} \varphi(\boldsymbol{x}, \boldsymbol{x}') \widehat{\mathbf{C}} \cdot\cdot \widehat{\boldsymbol{\varepsilon}} dS'(\boldsymbol{x}) dS =$$

$$= \oint\limits_{\Gamma_2} N_n^{(e)} \boldsymbol{p} d\Gamma - \iint\limits_S N_n^{(e)} \boldsymbol{b} dS, \quad n \in I^{(e)},\ (e) \in S_h.$$

Перейдём к индексной форме записи

$$p_1 \iint\limits_S N_{n,i}^{(e)} C_{ijkl}\varepsilon_{kl} dS + p_2 \iint\limits_S N_{n,i}^{(e)} \iint\limits_{S'(\boldsymbol{x})\cap S} \varphi(\boldsymbol{x},\boldsymbol{x}') C_{ijkl}\varepsilon_{kl} dS'(\boldsymbol{x}) dS =$$

$$= \oint\limits_{\Gamma_2} N_n^{(e)} p_j d\Gamma - \iint\limits_S N_n^{(e)} b_j dS, \quad i,j,k,l = \overline{1,2},\ n \in I^{(e)},\ (e) \in S_h.$$

Итоговое выражение, которое мы хотим получить можно записать в следующем виде

$$\left(p_1 \widehat{\mathbf{K}}^{Loc} + p_2 \widehat{\mathbf{K}}^{NonLoc}\right) \cdot\cdot \widehat{\mathbf{U}} = \widehat{\mathbf{P}} - \widehat{\mathbf{B}},$$

где $\widehat{\mathbf{K}}^{Loc}$ и $\widehat{\mathbf{K}}^{NonLoc}$ — тензоры четвёртого ранга, соответствующие матрицам жёсткости для локальной и нелокальной постановок соответственно;

$\widehat{\mathbf{U}}$ — тензор второго ранга, соответствующий вектору искомых узловых перемещений;

$\widehat{\mathbf{P}}$ — тензор второго ранга, дискретизация вектора плотности поверхностных сил;

$\widehat{\mathbf{B}}$ — тензор второго ранга, дискретизация вектора плотности объёмных сил.

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

следующим образом

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

$$i,j,k,l = \overline{1,2}, \quad n,m,p,q = \overline{1,L},$$

где L — количество узлов сетки; $\boldsymbol{E}_i$ — единичный вектор, размерности L. Тогда аппроксимацию матрицы $\widehat{\mathbf{K}}^{Loc}$ можно записать в следующем виде:

$$\widehat{\mathbf{K}}^{Loc} = \sum_{(e)\in S_h} \sum_{n,m\in I^{(e)}} \sum_{q\in Q^{(e)}} w_q \widehat{\mathbf{K}}_{nm}^{(e)(e)}(\boldsymbol{x}_q,\boldsymbol{x}_q)J_q^{(e)}, \tag{5}$$

где $Q^{(e)}$ — множество квадратурных узлов на элементе (e);

$\boldsymbol{x}_q$ — координаната квадратурного узла q;

w_q — квадратурный вес в квадратурном узле q;

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

Зону нелокального влияния $S'(\boldsymbol{x})$ аппроксимируем относительно каждого квадратурного узла сетки в отдельности [3]. Тогда, те элементы (e), квадратурные узлы $Q^{(e)}$ которых хотя бы частично попали в область влияния, будут участвовать в расчёте. Таким образом, аппроксимацию матрицы $\widehat{\mathbf{K}}^{NonLoc}$ можно записать в виде:

$$\widehat{\mathbf{K}}^{NonLoc} = \sum_{(e)\in S_h} \sum_{n\in I^{(e)}} \sum_{q\in Q^{(e)}} w_q J_q^{(e)} \times$$

$$\times \sum_{(e')\in S_h^q} \sum_{m'\in I^{(e')}} \sum_{q'\in Q^{(e')}} w_{q'}\varphi(\boldsymbol{x}_q,\boldsymbol{x}_{q'})\widehat{\mathbf{K}}_{nm'}^{(e)(e')}(\boldsymbol{x}_q,\boldsymbol{x}_{q'})J_{q'}^{(e')}. \tag{6}$$

Иллюстрация такого способа аппроксимации наглядно проиллюстрирована на рис. 4, где квадратурные узлы помечены точками, а квадратурный узел относительного которого происходит аппроксимация помечен крестом. Аппроксимированная зона нелокального влияния $S_h^q \subset S_h$ выделена серым цветом.

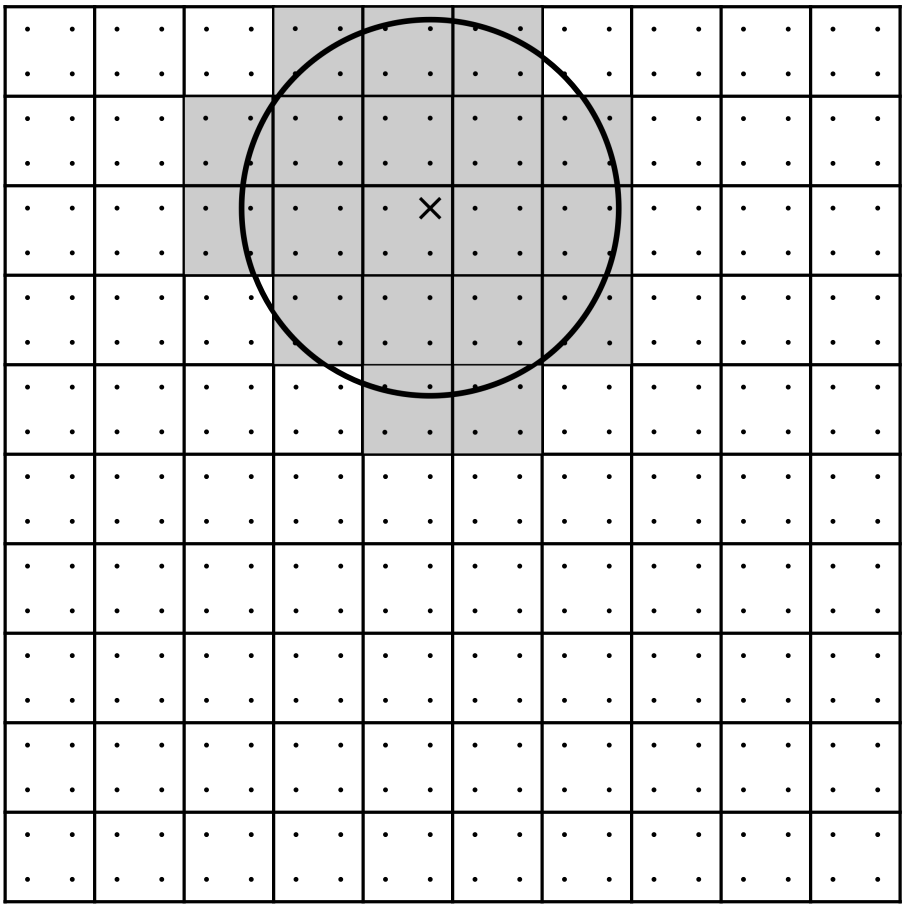


Рис. 4. Аппроксимация зоны нелокального влияния относительно квадратурного узла сетки

Аппроксимацию вектора плотности поверхностных сил $\widehat{\mathbf{P}}$ и вектора плотности объёмных сил $\widehat{\mathbf{B}}$ запишем в следующем виде

$$\widehat{\mathbf{P}} = \sum_{(be)\in\Gamma_h} \sum_{n\in I^{(be)}} \boldsymbol{E}_n \otimes \sum_{q\in Q^{(be)}} w_q N_n^{(be)}(\boldsymbol{x}_q)\boldsymbol{p}(\boldsymbol{x}_q)J_q^{(be)}, \tag{7}$$

$$\widehat{\mathbf{B}} = \sum_{(e)\in S_h} \sum_{n\in I^{(e)}} \boldsymbol{E}_n \otimes \sum_{q\in Q^{(e)}} w_q N_n^{(e)}(\boldsymbol{x}_q)\boldsymbol{b}(\boldsymbol{x}_q)J_q^{(e)}, \tag{8}$$

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

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

Вычисление матриц (5) и (6) можно упростить записав производные от функций форм в локальной системе координат элементов $(e) \in S_h$

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

Якобиан на границе аппроксимируем следующим образом

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

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

2. Программная реализация

2.1. Аппроксимация зоны нелокального влияния

Основная сложность, 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 (6) порядок сумм заставляет 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 [13]. Тогда формула (6) принимает вид:

$$\widehat{\mathbf{K}}^{NonLoc} = \sum_{(e)\in S_h} \sum_{n\in I^{(e)}} \sum_{(e')\in S_h^{(e)}} \sum_{m'\in I^{(e')}} \sum_{q\in Q^{(e)}} w_q J_q^{(e)} \times$$

$$\times \sum_{q'\in Q^{(e')}} w_{q'} \varphi(\boldsymbol{x}_q, \boldsymbol{x}_{q'}) \widehat{\mathbf{K}}_{nm'}^{(e)(e')}(\boldsymbol{x}_q, \boldsymbol{x}_{q'}) J_{q'}^{(e')}. \quad (9)$$

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

Такой способ аппроксимации проиллюстрирован на рис. 5, где центры элементов помечены точками, текущий элемент относительно которого проводится аппроксимация помечен крестом, а аппроксимированная зона закра-

шена серым цветом.

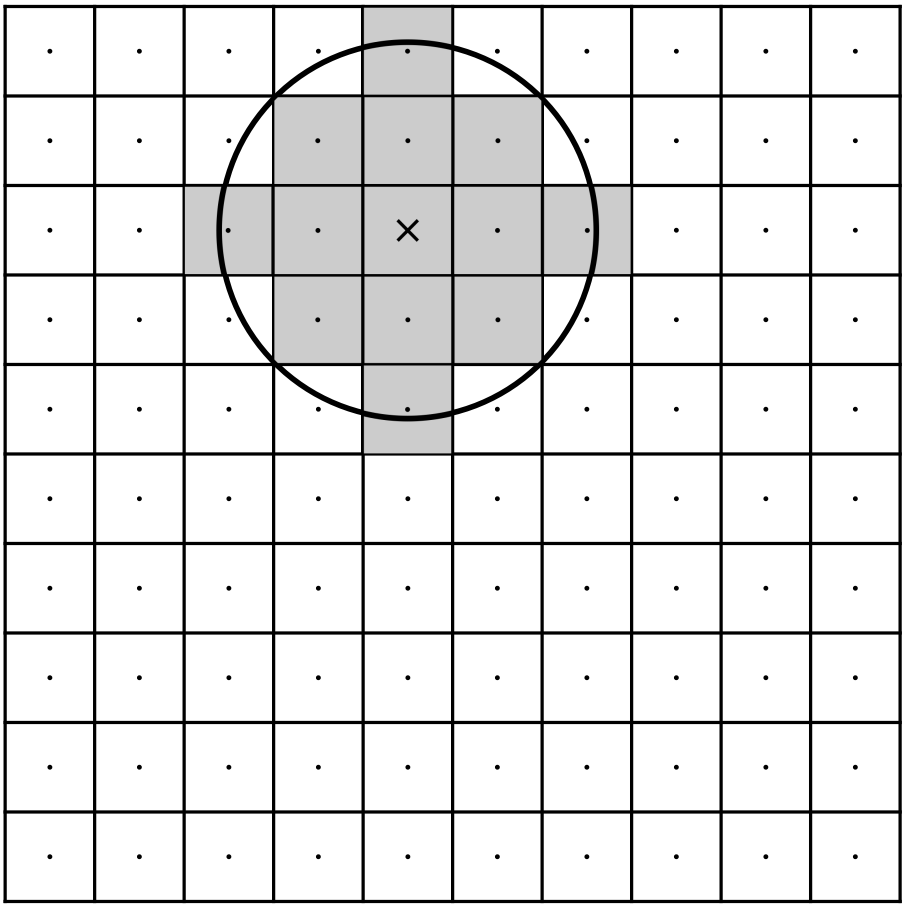


Рис. 5. Аппроксимация зоны нелокального влияния относительно центра конечного элемента

Заметим, что в формуле (9) изменён 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 r, таким образом возникает шанс того, что все квадратурные узлы, которые попали под зону влияния $S'(\boldsymbol{x})$, будут учтены в расчёте. Естественно, 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 (5) и (9), 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 $n \in S_h$ 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. Таким образом формулу (5) можно представить в следующей форме:

$$\widehat{\mathbf{K}}^{Loc} = \sum_{n \in I^{(e)}} \sum_{(e) \in E^n} \sum_{m \in I^{(e)}} \sum_{q \in Q^{(e)}} w_q \widehat{\mathbf{K}}^{(e)(e)}_{nm}(\boldsymbol{x}_q, \boldsymbol{x}_q) J^{(e)}_q. \tag{10}$$

Для матрицы жёсткости $\widehat{\mathbf{K}}^{NonLoc}$ формула (9) приобретает вид

$$\widehat{\mathbf{K}}^{NonLoc} = \sum_{n \in I^{(e)}} \sum_{(e) \in E^n} \sum_{(e') \in S^{(e)}_h} \sum_{m' \in I^{(e')}} \sum_{q \in Q^{(e)}} w_q J^{(e)}_q \times$$

$$\times \sum_{q' \in Q^{(e')}} w_{q'} \varphi(\boldsymbol{x}_q, \boldsymbol{x}_{q'}) \widehat{\mathbf{K}}^{(e)(e')}_{nm'}(\boldsymbol{x}_q, \boldsymbol{x}_{q'}) J^{(e')}_{q'}. \tag{11}$$

Так же преобразуем и формулу (8)

$$\widehat{\mathbf{B}} = \sum_{n \in I^{(e)}} \boldsymbol{E}_n \otimes \sum_{(e) \in E^n} \sum_{q \in Q^{(e)}} w_q N_n^{(e)}(\boldsymbol{x}_q) \boldsymbol{b}(\boldsymbol{x}_q) J_q^{(e)}. \tag{12}$$

Для общности, сделаем аналогичную процедуру и для аппроксимации граничных условий второго рода (7)

$$\widehat{\mathbf{P}} = \sum_{n \in \Gamma_h} \boldsymbol{E}_n \otimes \sum_{(be) \in E^n} \sum_{q \in Q^{(be)}} w_q N_n^{(be)}(\boldsymbol{x}_q) \boldsymbol{p}(\boldsymbol{x}_q) J_q^{(be)}. \tag{13}$$

Стоит добавить, что для аппроксимации граничных условий второго рода менять порядок циклов не обязательно, так как данная операция, как правило, не является трудозатратной и поэтому её проще всего реализовать согласно классическому представлению, добавив лишь дополнительные условия по заполнению массивов в самой программе.

2.3. Балансировка данных

Как уже было сказано, поузловой обход сетки (10)-(12) 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 неравномерное 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 элементов матрицы между процессами $p \in P$. Чтобы этого добиться, необходимо подсчитать количество элементов матрицы M_p, которыми владеет каждый из процессов. Затем сложить все эти суммы $M = \sum\limits_{p \in P} M_p$ 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 гибридном вычислительном кластере К-10 [14], 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 состоящей 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, получаемые в расчётах, имеют симметричную и разреженную структуру, поэтому для удобства будем хранить их в CSR формате [15]. Для индексации будем использовать 64-х битные числа, а для значений коэффициентов будем использовать числа с плавающей точкой двойной точности.

В табл. 1 представлены данные об объёмах затрачиваемой оперативной памяти и времени счёта на одном и 16 потоках. Из таблицы видно, что рост требований вычислительных ресурсов квадратичный относительно количества конечных элементов и радиуса поиска ближайших соседей для нелокальной постановки задачи, в то время как для классической постановки рост требований линейный. Связано это с тем, что среднее число ближайших соседей, приходящихся на каждый конечный элемент, растёт квадратично. Однако, стоит заметить, что эффективность распараллеливания для нелокальных задач намного выше, чем для классических, что более наглядно проиллюстрировано на рис. 6.

При использовании параллельных и распределённых вычислений одновременно удаётся достичь ещё большего ускорения сборки матрицы жёсткости. Для 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 конечных элементов	Радиус поиска соседей	Среднее число соседей	Требуемый объём памяти	Время расчёта 1 поток	Время, расчёта 16 потоков
2500	0	—	3.7 Мб	0.031 с	0.017 с
10000	0	—	15 Мб	0.152 с	0.046 с
40000	0	—	59 Мб	1.280 с	0.207 с
2500	0.1	66	63 Мб	4.385 с	0.307 с
10000	0.1	281	916 Мб	76.65 с	5.102 с
40000	0.1	1143	13.3 Гб	1265 с	82.32 с
2500	0.2	257	210 Мб	17.40 с	1.166 с
10000	0.2	1042	3.1 Гб	288.4 с	18.71 с
40000	0.2	4194	47 Гб	4704 с	303.2 с

Таблица 1. Затрачиваемые вычислительные ресурсы на различных сетках и при различных радиусах поиска ближайших соседей

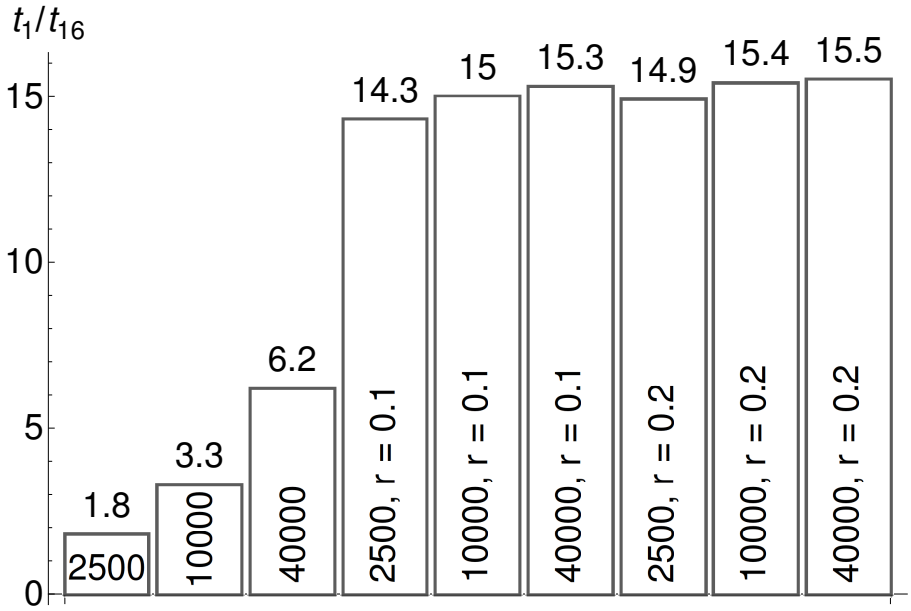
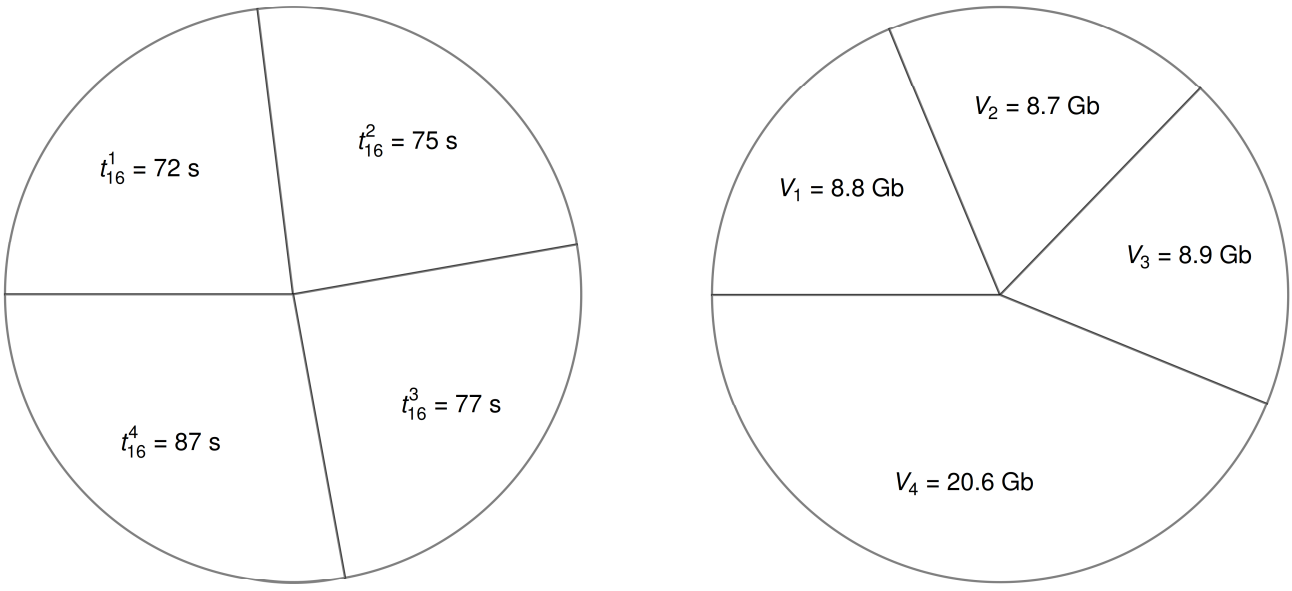


Рис. 6. Ускорение времени сборки матрицы жёсткости при использовании технологии OpenMP на 16 потоках в сравнении со временем счёта на 1 потоке

При балансировке объёмов вычислений получаем равномерное распределение времени счёта, но при этом наблюдаем сильный дисбаланс в распределении объёмов затрачиваемой оперативной памяти. Исходя из результатов, представленных в табл. 2, на четвёртом процессе объёмы используемой оперативной памяти в два раза выше, чем у всех остальных процессов, что объясняется количеством самопересекающихся индексов приходящихся на каждый из процессов. На рис. 7, на примере задачи на сетке состоящей из 40000 элементов и радиусом поиска $\tilde{r} = 0.2$, наглядно проиллюстрированы распределения времени счёта и объёмы затрачиваемой оперативной памяти.

N	$\tilde{r}$	V_1	V_2	V_3	V_4	t_{16}^1, с	t_{16}^2, с	t_{16}^3, с	t_{16}^4, с
10000	0.1	181 Мб	173 Мб	174 Мб	388 Мб	1.269	1.312	1.222	1.482
10000	0.2	607 Мб	582 Мб	587 Мб	1.3 Гб	4.736	4.508	4.474	5.376
40000	0.1	2.5 Гб	2.5 Гб	2.5 Гб	5.8 Гб	20.34	20.93	19.54	23.48
40000	0.2	8.8 Гб	8.7 Гб	8.9 Гб	20.6 Гб	71.63	75.31	77.42	86.76

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



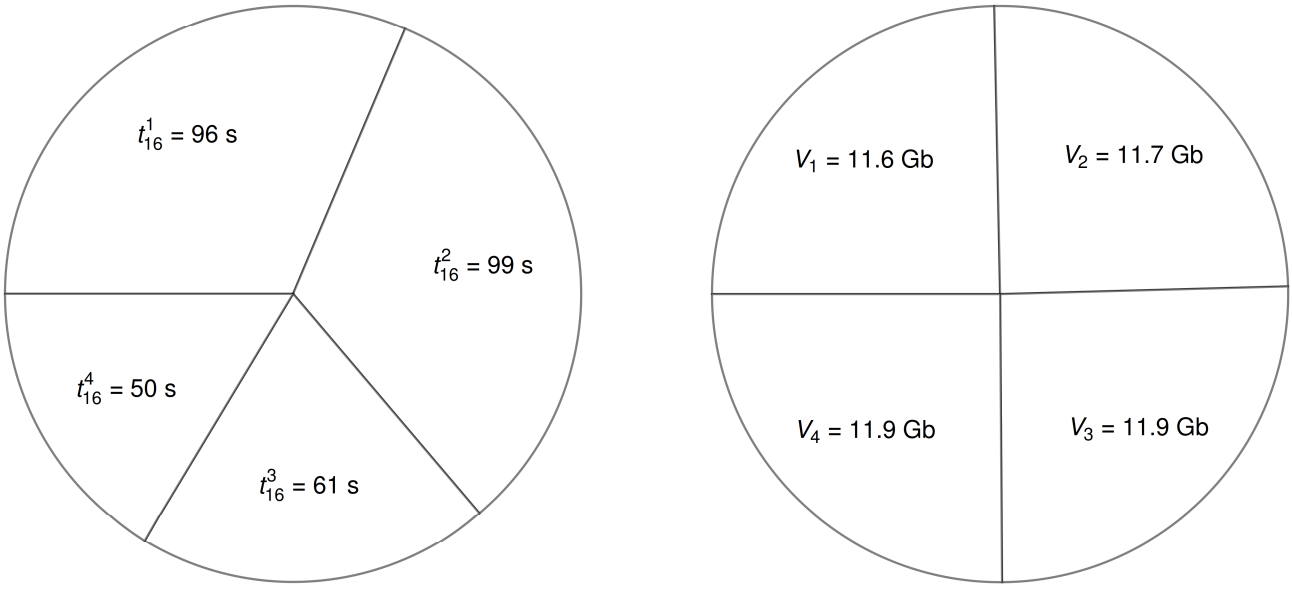
(а) (б)

Рис. 7. Распределение времени счёта (а) и объёмов используемой оперативной памяти (б) при балансировке объёмов вычислений

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. Результаты расчётов представлены в табл. 3. Как и в предыдущем случае, на примере задачи на сетке состоящей из 40000 элементов и радиусом поиска ближайших соседей $\tilde{r} = 0.2$, проиллюстрируем на рис. 8 диаграммы распределения времени счёта и объёмов потребляемой памяти.

N	$\tilde{r}$	V_1	V_2	V_3	V_4	t_{16}^1, с	t_{16}^2, с	t_{16}^3, с	t_{16}^4, с
10000	0.1	225 Мб	229 Мб	233 Мб	229 Мб	1.675	1.786	1.123	0.887
10000	0.2	768 Мб	782 Мб	797 Мб	782 Мб	5.704	6.029	3.918	3.159
40000	0.1	3.3 Гб	3.3 Гб	3.4 Мб	3.3 Гб	26.74	27.96	16.91	13.70
40000	0.2	11.6 Гб	11.7 Гб	11.9 Гб	11.8 Гб	95.76	99.08	60.50	50.03

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



(а) (б)

Рис. 8. Распределение времени счёта (а) и объёмов используемой оперативной памяти (б) при балансировке количества элементов матрицы

Подытожим результаты использования распределённых вычислений диаграммой представленной на рис. 9. Из данной диаграммы видно, что при использовании распределённых вычислений наблюдается существенное ускорение времени счёта независимо от используемой балансировки. 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.

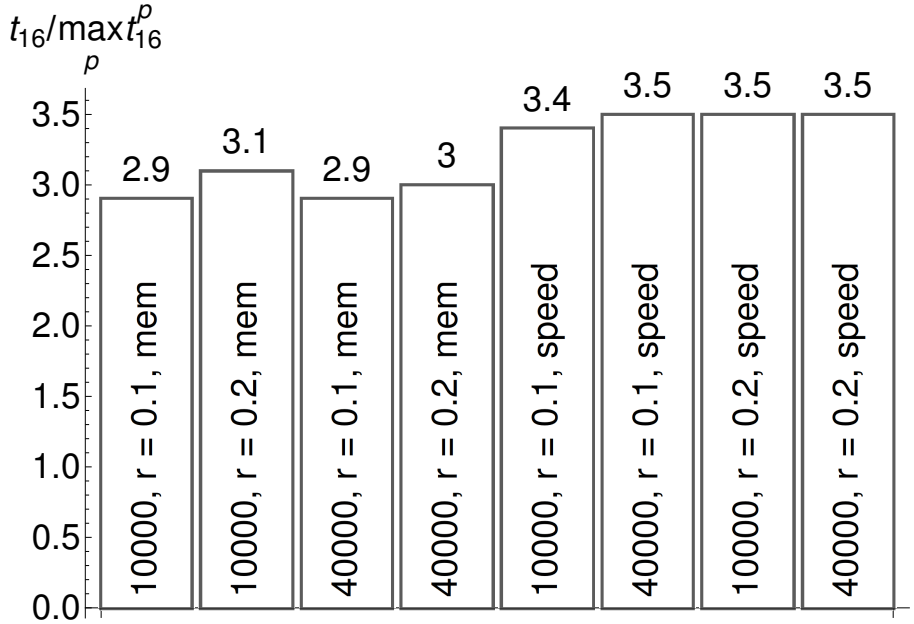


Рис. 9. Ускорение времени сборки матрицы жёсткости при использовании технологии MPI, где speed означает балансировку объёмов вычислений, а mem балансировку объёмов памяти

ЗАКЛЮЧЕНИЕ

В работе было рассмотрено двумерное уравнение равновесия в нелокальной постановке. Исследования показали, что несмотря на такую постановку задачи, здесь, как и в классическом случае, выполняется принцип Сен-Венана, который, в отличие от классического решения, не даёт константы вдали от приложенных нагружений, но в тоже время интегрально совпадает с ними. В областях со ступенчатыми переходами или эллиптическими вырезами наблюдается смягчение концентраторов, которое характеризуется увеличением деформаций, но снижением напряжений. Также в окрестностях концентраторов в деформациях наблюдаются отрицательные величины, которые тем больше и тем шире, чем больше вклад и радиус нелокального влияния, соответственно.

Был предложен и реализован параллельный и распределённый поузловой алгоритм сборки матрицы жёсткости на основе метода конечных элементов и написана программа на языке программирования C++. Были также предложены и реализованы различные варианты балансировки данных между процессами. Распараллеливание программы было произведено при помощи технологии параллельного программирования OpenMP [18]. Распределённые вычисления были реализованы на основе технологии MPI [19]. Для хранения и решения разреженных симметричных СЛАУ использовалась библиотека линейной алгебры Eigen [20]. В качестве распределённых решателей СЛАУ были использованы библиотеки PETSc [21] и Intel MKL PARDISO [22]. Для реализации обобщённой конечно-элементной модели была использована библиотека символьного дифференцирования на этапе компиляции symdiff [23]. Моделирование геометрии и генерация конечно-элементной сетки была сделана в SALOME [24]. Обработка результатов и выведение графиков были сделаны при помощи Paraview [25] и Wolfram Mathematica [26].

Список использованных источников

1. Thangadurai T.D., Manjubaashini N., Thomas S., Maria H.J. Nanostructured Materials. Springer International Publishing, 2020. 210 p.

2. Eringen A.C. Nonlocal continuum field teories. New York-Berlin-Heidelberg: Springer-Verlag, 2002.

3. Pisano A.A., Sofi A., Fuschi P., 2009. Nonlocal integral elasticity: 2D finite element based solutions. Int. J. Solids Struct. 46, 3836–3849.

4. Wen P.H., Huang X.J., Aliabadi M.H. Two Dimensional Nonlocal Elasticity Analysis by Local Integral Equation Method. CMES, vol.96, no.3, pp.199-225, 2013

5. Abdollahi R. Benchmarks in nonlocal elasticity defined by Eringen's integral model / Abdollahi R., Boroomand B. – Int. J. Solids Struct, 2013, 50, p. 2758–2771.

6. Астионенко И.А., Гучек П.И., Литвиненко Е.И., Хомченко А.Н. Применение альтернативных серендиповых моделей при решении задач о кручении призматических стержней. Вестник ХНТУ №1(46), 2013.

7. Астионенко И.А., Литвиненко Е.И., Хомченко А.Н. Конструирование многопараметрических полиномов на бикубическом элементе серендипова семейства. Научные ведомости БелГУ №5(60), 2009.

8. Зарубин В.С., Кувыркин Г.Н. Математические модели механики и электродинамики сплошной среды. М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. 512 с.

9. Абрамовица М., Стиган И. Справочник по специальным функциям. С формулами, графиками и математическими таблицами. М.: Наука, 1979. 832 с.

10. Zienkiewicz O. The Finite Element Method: Its Basis and Fundamentals. Seventh edition. / O. Zienkiewicz, R. Taylor, J.Z. Zhu — 2013. — 756 p.

11. Bathe K-J. Finite Element Procedures. Second edition. — 2014. — 1065 p.

12. Мейз Дж. Теория и задачи механики сплошных сред. М.: Изд-во «Мир». 1974. 320 с.

13. Kuvyrkin G.N., Savelyeva I. Yu., Sokolov A.A. Features of the software implementation of the numerical solution of stationary heat equation taking into account the effects of nonlocal finite element method. Journal of Physics: Conference Series, Volume 1479, Applied Mathematics, Computational Science and Mechanics: Current Problems 11-13 November 2019, Voronezh, Russian Federation

14. https://www.kiam.ru/MVS/resourses/k10.html (Дата обращения 31.03.2021)

15. Писсанецки С. Технология разреженных матриц: Пер. с англ. — М.: Мир. 1988. — 410 с. ил.

16. Андреев А.В. Инженерные методы определения концентрации напряжений в деталях машин. М.: Машиностроение, 1976, 72 с.

17. Биргер И.А., Шорр Б.Ф., Иосилевич Г.Б. Расчет на прочность деталей машин: Справочник. 4-е изд., перераб. и доп. — М.: Машиностроение, 1993. — 640 с: ил.

18. Антонов А.С. Параллельное программирование с использованием технологии OpenMP — М.: Изд-во Московского Университета, 2009. — 78 с.

19. Абрамян, М.Э. Параллельное программирование на основе технологии MPI 2.0 : учебник — Ростов-на-Дону; Таганрог: Изд-во ЮФУ, 2018. — 357 с.

20. Eigen: https://eigen.tuxfamily.org/ (Дата обращения 31.03.2021)

21. PETSc: https://www.mcs.anl.gov/petsc (Дата обращения 31.03.2021)

22. Intel MKL PARDISO: https://software.intel.com/content/www/us/en/develop/documentation/onemkl-developer-reference-c/top.html (Дата обращения 31.03.2021)

23. Краснов М.М. Метапрограммирование шаблонов C++ в задачах математической физики. М.: ИПМ им. М.В. Келдыша, 2017, 84 с.

24. SALOME: https://www.salome-platform.org/ (Дата обращения 31.03.2021)

25. Paraview: https://www.paraview.org/ (Дата обращения 31.03.2021)

26. Mangano S. Mathematica Cookbook. — O'Reilly Media: Sebastopol, 2010. — 800 p.

ПРИЛОЖЕНИЕ А

При решении задач методом конечных элементов возникает потребность в эффективном решении СЛАУ. Прямые методы требуют больших затрат оперативной памяти, поэтому возникает спрос на использование итерационных методов решения. Однако итерационные методы могут иметь слишком медленную сходимость, которая в первую очередь обусловлена самой системой. Таким образом требуется «смягчить» получаемую матрицу жёсткости, чтобы увеличить эффективность итерационных методов. Одним из способов «смягчения» является выбор альтернативного базиса, для которого скорость сходимости итерационного метода будет наискорейшей.

Для задач в двумерных постановках, в целях экономии вычислительных ресурсов, разумнее всего использовать элементы серендипового семейства. Однако базисы таких элементов, которые предложил Зенкевич [10], обладают рядом дефектов, которые повышают жёсткость итоговой системы уравнений. Но были предложены альтернативные базисы [6], [7], которые лишены данных дефектов. Так, например, для квадратичного серендипового элемента, изображённого на рис. 10, можно ввести один свободный параметр s, благодаря которому базисные функции приобретают вид [6]

$$N_i = \frac{1}{16}(1+\xi_i\xi)(1+\eta_i\eta)((9s-1)(1-\xi_i\xi-\eta_i\eta)+(9s+3)\xi_i\xi\eta_i\eta),$$

$$i = 1,3,5,7, \quad \xi_i, \eta_i = \pm 1,$$

$$N_i = \frac{1}{16}(1-\xi^2)(1+\eta_i\eta)((5-9s)+(9s+3)\eta_i\eta), \quad i = 2,6, \quad \eta_i = \pm 1,$$

$$N_i = \frac{1}{16}(1-\eta^2)(1+\xi_i\xi)((5-9s)+(9s+3)\xi_i\xi), \quad i = 4,8, \quad \xi_i = \pm 1.$$

Выбор параметра s был основан на следующих предположениях

$$\int_1^1 \int_1^1 N_i d\xi d\eta = s, \quad i = 1, 3, 5, 7,$$

$$\int_1^1 \int_1^1 N_i d\xi d\eta = 1 - s, \quad i = 2, 4, 6, 8.$$

Таким образом, стандартный базис, который предложил Зенкевич [10], будет получен при $s = -1/3$.

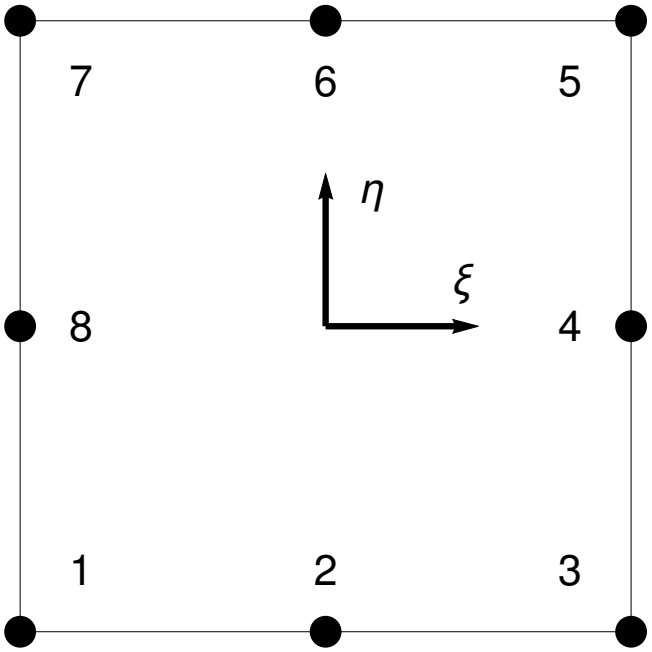


Рис. 10. Нумерация узлов на квадратичном серендиповом элементе

Теперь же проверим зависимость количества итераций метода сопряжённых градиентов от параметра s, а также при различных параметрах p_1. В качестве тестовой задачи возьмём задачу Неймана на области $S = [0, 5] \times [0, 1]$ с введённой на ней равномерной сеткой S_h, состоящей из 18000 квадратичных

серендиповых элементов. Поставим следующие граничные условия

$$\overline{\boldsymbol{\sigma}} \cdot \boldsymbol{n}|_{\overline{x}_1=0} = -f(\overline{x}_2), \quad \overline{\boldsymbol{\sigma}} \cdot \boldsymbol{n}|_{\overline{x}_1=5} = f(\overline{x}_2),$$

где f определяется следующим образом

$$f(x) = \begin{cases} 4x, & x \leqslant 0.5, \\ \\ 4 - 4x, & x > 0.5. \end{cases}$$

Значение параметра s	Количество итераций при			
	$p_1 = 1$	$p_1 = 2/3$	$p_1 = 1/2$	$p_1 = 1/3$
-1/3	2410	2014	1755	1543
-1/4	2085	1810	1577	1387
-1/6	1938	1620	1411	1241
-1/12	1736	1451	1345	1112
0	1514	1314	1217	1006
1/12	1558	1227	1136	938
1/6	1485	1240	1079	948
1/4	1591	1252	1160	958
1/3	1511	1262	1170	966
5/12	1618	1274	1181	974
1/2	1560	1306	1210	999

Таблица 4. Количество итераций метода сопряжённых градиентов в классическом и нелокальном случаях при различных параметрах s и p_1 при $r = 0.1$

Исходя из результатов, представленных в табл. 4 и на рис. 11, видно, что при отрицательных значениях параметра s количество итераций значительно

больше, чем при равных по модулю положительных значениях. Однако при достаточно большом увеличении параметра s, количество итераций снова начинает увеличиваться и минимум находится в окрестности точки $s = 1/6$. Также отметим, что в нелокальном случае, при уменьшении параметра p_1, наблюдается уменьшение количества итераций, но несмотря меньшее количество итераций по сравнению с классическим подходом, время вычислений значительно больше, так как каждая итерация обходится дороже из-за более плотной матрицы жёсткости.

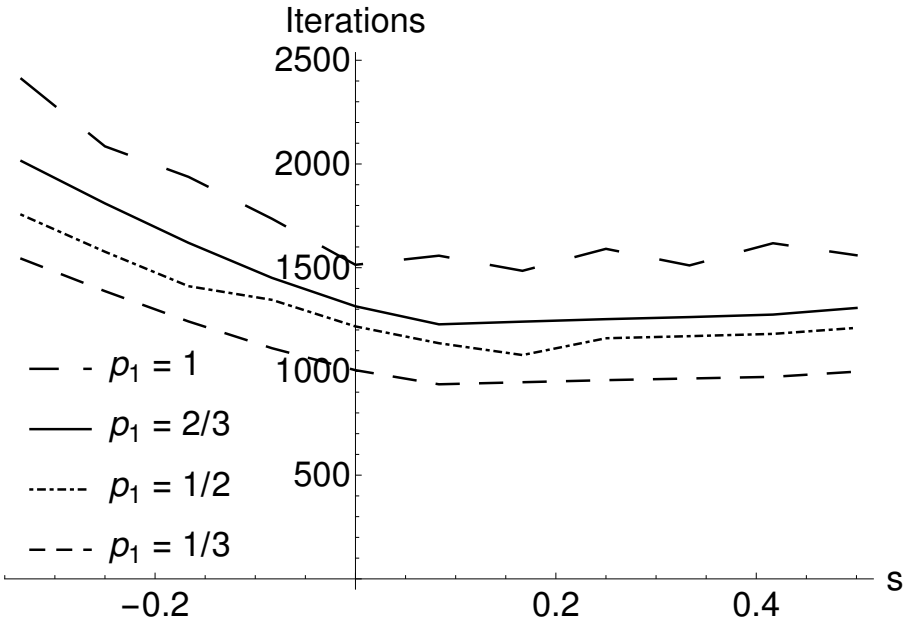


Рис. 11. Зависимость количества итераций от параметра s в локальном и нелокальном случаях

Таким образом, использование нестандартных базисов, хоть и не значительно, но всё же способно ускорить скорость решения СЛАУ при использовании итерационных методов. Что касается точности, то практика показала, что для задач рассматриваемых в данной работе, разница между решениями порядка 10^{-5}, что не очень существенно.