

УДК 004.021

Снижение неравномерности объемов хранения данных при разделении конечно-элементных моделей на суперэлементы

Берчун Ю. В.¹, Бирюкова М. М.¹,
Киселёв И. А.¹, Соколов С. С.¹,
Шевченко А. С.^{1,*}

* artem_sh@icloud.com

¹МГТУ им. Н.Э. Баумана, Москва, Россия

Настоящая работа посвящена рассмотрению двух методов автоматического разделения конечно-элементной модели на суперэлементы. Первый метод представляет собой алгоритм разделения сетки на односвязные подобласти по характерному заданному числу узлов в суперэлементе. Второй метод основывается на формировании суперэлементов с заданным характерным объемом хранения матрицы коэффициентов системы уравнений равновесия внутренних узлов, исключаемых в ходе суперэлементного преобразования. На тестовых примерах представлено сравнение результатов работы указанных методов с точки зрения неравномерности генерируемого суперэлементного разбиения по числу узлов и объему хранения матрицы коэффициентов системы уравнений равновесия внутренних узлов.

Ключевые слова: метод конечных элементов, метод суперэлементов, разделение на суперэлементы, формирование суперэлементов, объем хранения

Введение

Для решения линейно-упругих статических задач механики деформируемого твердого тела широко применяют метод конечных элементов (МКЭ). В соответствии с данным методом требуется осуществить дискретизацию расчетной области на неперекрывающиеся конечные элементы (КЭ) простой формы, взаимодействующие друг с другом в узлах конечно-элементной сетки. Решение задачи сводится к определению перемещений узлов сетки по степеням свободы (направлениям независимых перемещений, полностью определяющих положение узла) за счет решения системы уравнений равновесия для дискретной модели [1]

$$Kq = P. \quad (*)$$

Здесь K – матрица жесткости полной модели, составляется из матриц жесткости отдельных КЭ в процессе ансамблирования; q – вектор узловых перемещений в состоянии статического равновесия; P – вектор внешних узловых сил полной конечно-элементной модели.

СЛАУ (*) в силу топологии модели является существенно разреженной (заполнение ненулевыми элементами менее 1 %), симметричной и положительно определенной. Для ее решения используются итерационные (например, метод сопряженных градиентов [2]) или прямые методы (в первую очередь, метод Холецкого [3]).

Одной из основных проблем применения прямых методов является существенное увеличение числа ненулевых элементов в процессе разложения матрицы K . С вычислительной точки зрения это приводит, главным образом, к избыточной потребности в оперативной памяти. Число ненулевых элементов, получаемых в процессе разложения, определяется не только числом ненулевых элементов в исходной матрице, но и характером их расположения. В этой связи используют понятие ширины профиля матрицы, характеризующее максимальное отдаление ненулевых элементов от главной диагонали. Для снижения ширины профиля матрицы (и, как следствие, уменьшения числа ненулевых элементов) применяют методы перестановки строк и столбцов, разработанные на основе теории графов, что фактически означает перенумерацию узлов исходной КЭ сетки. Широко применение находит алгоритм Катхилла-Макки (Cuthill-McKee) [3].

Однако в случае, если объект моделирования имеет сложную геометрическую форму и требует подробной КЭ сетки, число уравнений СЛАУ (*) может составлять десятки миллионов и более. Вслед за размерностью задачи растут и требования к объему оперативной памяти. Кроме того, упоминавшийся выше алгоритм перенумерации сам по себе обладает вычислительной сложностью, что негативно сказывается на производительности алгоритма в целом по мере увеличения размерности.

Преодоление проблемы размерности возможно за счет декомпозиции задачи. Одним из эффективных подходов к решению задач МКЭ высокой размерности является применение метода суперэлементов (МСЭ) [4]. Задача выделения суперэлементов (СЭ) сводится к задаче разбиения графов. Для этого СЭ сетку представляют в виде графа, узлы которого соответствуют узлам сетки КЭ, а ребрами попарно связаны все вершины, относящиеся к одному КЭ.

Задача выделения СЭ представляет собой проблему разделения графа на k частей и описывается следующим образом. Дан граф $G = (V, E)$, где $|V| = n$. Требуется разделить V на подмножества $V_1 \dots V_k$, такие, что $V_i \cap V_j = \emptyset$ в случае $i \neq j$, $|V_i| \approx n/k$, $\cup_{i=1}^k V_i = V$ [5]. В случае разделения конечно-элементной сетки на СЭ необходимо также учитывать, что КЭ должен входить в СЭ целиком даже если его отдельные узлы после разделения принадлежат разным СЭ.

Задачи разделения (разбиения) графов можно классифицировать на две группы: условные и безусловные.

Целью условного разделения графа является, в первую очередь, выделение частей, имеющих идентичные (или близкие) характеристики (в некоторых статьях используется термин вес). Примером такой характеристики можно считать число вершин или число ребер в выделенной части. Данный тип разделения чаще всего используют для решения инженерных задач, требующих трудоемких вычислений [6].

Цель безусловного разделения графа – минимизация числа ребер, пересекаемых границей между частями. Разделение данного типа широко используют при фрагментации изображений [7, 8], классификации текстов [9, 10], а также для решения некоторых специфических задач, таких как задача разделения воздушного пространства [11].

Задача о разделении КЭ модели на СЭ может быть поставлена, исходя как из целей безусловного, так и условного разбиений. Это зависит от приоритетов, которыми руководствуется разработчик КЭ комплекса. Как характеристики получаемых отдельных СЭ, так и степень их связанности оказывают влияние на трудоемкость всего расчета. В рамках данной статьи задача разделения трактуется как условная, так как это позволяет в явном виде оценить неравномерность разбиения, построенного на основе предлагаемой характеристики. При классическом подходе, характеристикой является либо число узлов в СЭ, либо число КЭ в СЭ. Очевидно, что это всегда величины одного порядка, поэтому далее будем говорить лишь о первом варианте.

Для решения задачи разделения графа разработано большое число алгоритмов, от достаточно простых, основанных на регулярном обходе графа (например, поиск в ширину [12]), до сложных, таких как спектральные алгоритмы [13], геометрические алгоритмы [14], алгоритмы многоуровневого деления графа [5, 15].

Упомянутые выше алгоритмы изначально построены для разделения графа на близкие по числу вершин части. Это хорошо согласуется с рассмотренным классическим подходом к выбору характеристики, но при этом возникает вопрос об эффективности полученных разбиений с точки зрения трудоемкости последующих вычислений, так как в силу применяемых численных методов трудоемкость решения СЛАУ для каждого СЭ зависит не только от размерности и заполнения ненулевыми элементами матрицы жесткости СЭ, но и от ширины ее профиля. В силу симметрии указанной матрицы в памяти компьютера ее целесообразно хранить в профильном виде [16], поэтому в качестве характеристики предлагается выбрать объем памяти, необходимый для хранения соответствующей структуры данных (для простоты определим его как «объем хранения матрицы»). Подчеркнем отличие понятия объема хранения матрицы от понятия ширины профиля матрицы. Последний оценивает максимальное отклонение ненулевого элемента от главной диагонали, в то время как объем хранения матрицы представляет собой

интегральную характеристику всей матрицы. Следует отметить, что предложенная характеристика резко меняет сам подход к выделению СЭ в МСЭ – вместо безразмерных величин (или физических величин, выражаемых в традиционных для механики единицах измерения) при наборе узлов в СЭ предлагается применять величину, измеряемую в мегабайтах. Авторам неизвестны работы в области МСЭ, в которых характеристики, используемые при выделении СЭ, формулируются на основе понятий информационных технологий.

В статье ставится задача исследования влияния предложенной характеристики на качество работы алгоритма разбиения КЭ модели на СЭ. В качестве критерий оценки качества разделения используем неравномерность полученных СЭ, которую рассчитываем как с точки зрения числа узлов, так и с точки зрения объемов хранения матриц.

В первом разделе работы приводим основные соотношения метода суперэлементов и выполняем анализ структуры вычислительных затрат при его реализации, позволяющий сформулировать требования к алгоритмам разбиения на СЭ. Второй раздел посвящен описанию простейших алгоритмов разбиения на СЭ, использующих различные характеристики для построения разбиения на СЭ. В третьем разделе приведены результаты применения рассмотренных алгоритмов к двум тестовым моделям для сравнения их эффективности. Показано, что оценка объема хранения матриц СЭ на этапе разбиения позволяет сбалансировать неравномерность получаемых СЭ по различным критериям.

1. Вычислительные аспекты метода суперэлементов

1.1 Основные соотношения метода суперэлементов

Ключевой особенностью МСЭ является возможность рассмотрения анализируемой модели в виде совокупности вложенных подмоделей (суперэлементов). СЭ представляют собой непересекающиеся объединения конечных элементов или СЭ более низкого уровня, взаимодействующих между собой только в граничных узлах. Это подразумевает необходимость описания поведения каждого СЭ в отдельности через параметры только в граничных узлах. Равновесие СЭ определенного уровня, как самостоятельной единицы, описывается через узловые параметры системой (1) вида аналогичного (*):

$$K^{\alpha,\delta} q^{\alpha,\delta} = P^{\alpha,\delta}. \quad (1)$$

Здесь α – номер уровня СЭ;

δ – порядковый номер СЭ для рассматриваемого уровня;

$K^{\alpha,\delta}$ – матрица жесткости СЭ;

$q^{\alpha,\delta}$ – вектор узловых перемещений СЭ;

$P^{\alpha,\delta}$ – вектор узловых усилий СЭ [4].

Размерность системы (1) существенно меньше размерности полной системы (*) и зависит от числа СЭ и уровней суперэлементного разбиения, на которые разделена исходная модель. За счет разделения узлов СЭ на внутренние и граничные (входящие более чем в один СЭ), его уравнение равновесия (1) может быть представлено в блочном виде

$$\begin{pmatrix} K_{ii} & K_{is} \\ K_{si} & K_{ss} \end{pmatrix} \begin{pmatrix} q_i \\ q_s \end{pmatrix} = \begin{pmatrix} P_i \\ P_s \end{pmatrix},$$

где i – индекс внутренних узлов;

s – индекс граничных узлов;

q_i – вектор перемещений внутренних узлов;

q_s – вектор перемещений граничных узлов;

P_i – вектор нагрузок, приложенных к внутренним узлам;

P_s – вектор нагрузок, приложенных к граничным узлам.

Для описания взаимодействия СЭ между собой и формирования следующих уровней суперэлементного разбиения (за счет процедуры ансамблирования [1]) используют модифицированные матрицы жесткости для граничных узлов k , которые могут быть определены для каждого СЭ выражением

$$k = K_{ss} - K_{si} \cdot K_{ii}^{-1} \cdot K_{is}. \quad (2)$$

Подобное преобразование применяется и для вектора узловых сил

$$p = P_s - K_{si} \cdot K_{ii}^{-1} P_i. \quad (3)$$

Смысл преобразований (2), (3) состоит в исключении из СЛАУ (1) уравнений для внутренних узлов (называемых иногда по этой причине «исключаемыми»). Самый верхний уровень СЭ разбиения характеризуется отсутствием граничных узлов. На этом уровне решение задачи сводится к решению СЛАУ, аналогичной (*), но существенно меньшей размерности. Число уровней СЭ выбирают таким образом, чтобы на верхнем уровне получить СЛАУ приемлемого размера, решение которой является возможным с учетом имеющихся ограничений по оперативной памяти ЭВМ. После решения задачи на верхнем уровне необходимо последовательно определить перемещения внутренних узлов СЭ всех уровней через перемещения их граничных узлов, проходя по структуре уровней сверху вниз и используя соотношение

$$q_i = K_{ii}^{-1} (P_i - K_{is} q_s). \quad (4)$$

Выражения (2)-(4) представляют собой основные соотношения СЭ преобразования. Основные вычислительные затраты сосредоточены на этапе построения матриц жесткости для граничных узлов СЭ (2) в операции

$$K_{is}^{NEW} = K_{ii}^{-1} \cdot K_{is}. \quad (5)$$

1.2 Анализ структуры вычислительных затрат при реализации МСЭ

Столбец j матрицы K_{is}^{NEW} является результатом решения СЛАУ с матрицей K_{ii} и правой частью в виде j -ого столбца матрицы K_{is} . Для отыскания подобных решений целесообразно использовать метод Холецкого. Прямой ход, т.е. нахождение такой верхней треугольной матрицы L , что $K_{ii} = L^T \cdot L$, выполняется однократно. Обратный ход (подстановка правой части) выполняется многократно, по числу S столбцов матрицы K_{is} , то есть столько раз, сколько степеней свободы имеет СЭ в граничных узлах. Потребность в вычислительных затратах для выполнения операции (5) зависит от размерности матрицы K_{ii} , то есть числа внутренних узлов СЭ, и числа столбцов матрицы K_{is} , равного числу граничных узлов. Для повышения производительности расчет для больших СЭ-моделей целесообразно проводить параллельно, с использованием распределенных вычислений. Поскольку каждый из узлов вычислительной сети обрабатывает в каждый момент времени отдельный СЭ, для выравнивания вычислительной нагрузки необходимо выделять СЭ, матрицы которых имеют примерно одинаковый размер. СЭ модель может быть сформирована в режиме, когда выделение отдельных СЭ производится пользователем в явной форме по конструктивным соображениям. В этом случае, как правило, разбиение получается очень неравномерным по объему матриц СЭ, кроме того, число граничных узлов может оказаться не оптимальным. Другим подходом является применение алгоритмов автоматического разделения исходной сетки на СЭ по некоторому критерию.

2. Сравнимых алгоритмы разбиения на СЭ

Сравним два алгоритма автоматического разделения исходной сетки КЭ, в основе каждого из которых лежит формирование СЭ методом поиска в ширину (breadth-first search, BFS [12]), по критериям оценки качества разбиения на СЭ (п. 3.3). Этот метод был выбран как свободный от эвристик, что наиболее подходит для анализа на данном этапе исследований. Отличие алгоритмов состоит в использовании разных критериев окончания формирования СЭ.

В первом алгоритме критерием окончания формирования СЭ является заданное пользователем характерное число узлов в СЭ, во втором алгоритме – заданный пользователем характерный объем хранения соответствующих СЭ матриц внутренних (исключаемых) степеней свободы.

С помощью каждого из анализируемых алгоритмов на первом этапе осуществляем формирование первоначальной СЭ-модели. После этого проверяется характерный объем каждого сформированного на первой стадии СЭ (по числу узлов или объему матрицы исключаемых узлов, в зависимости от алгоритма). При обнаружении СЭ, у которого

объем меньше некоторого наперед заданного порогового значения, производим его слияние с одним из соседних СЭ «нормального» размера. Таким образом, удается исключить из модели заведомо малые СЭ. В качестве порогового значения в настоящей работе была выбрана $1/3$ характерного заданного объема СЭ.

Блок схема алгоритма разделения по числу вершин приведена на рис. 1.

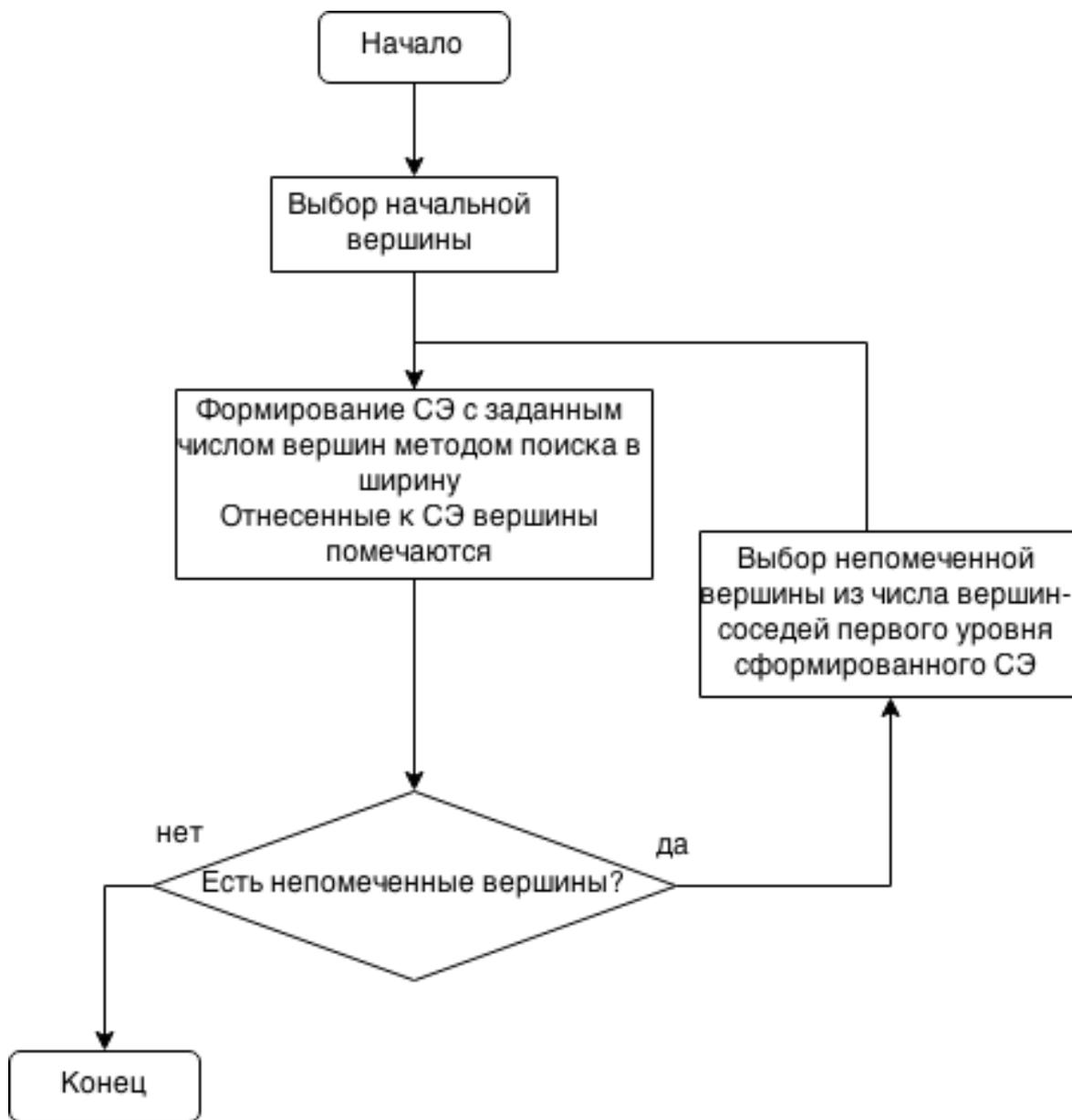


Рис. 1. Блок-схема алгоритма разделения по числу вершин

Блок схема алгоритма разделения по объему хранения матриц СЭ приведена на рис. 2. Выделенная часть алгоритма описывает формирование одного СЭ.

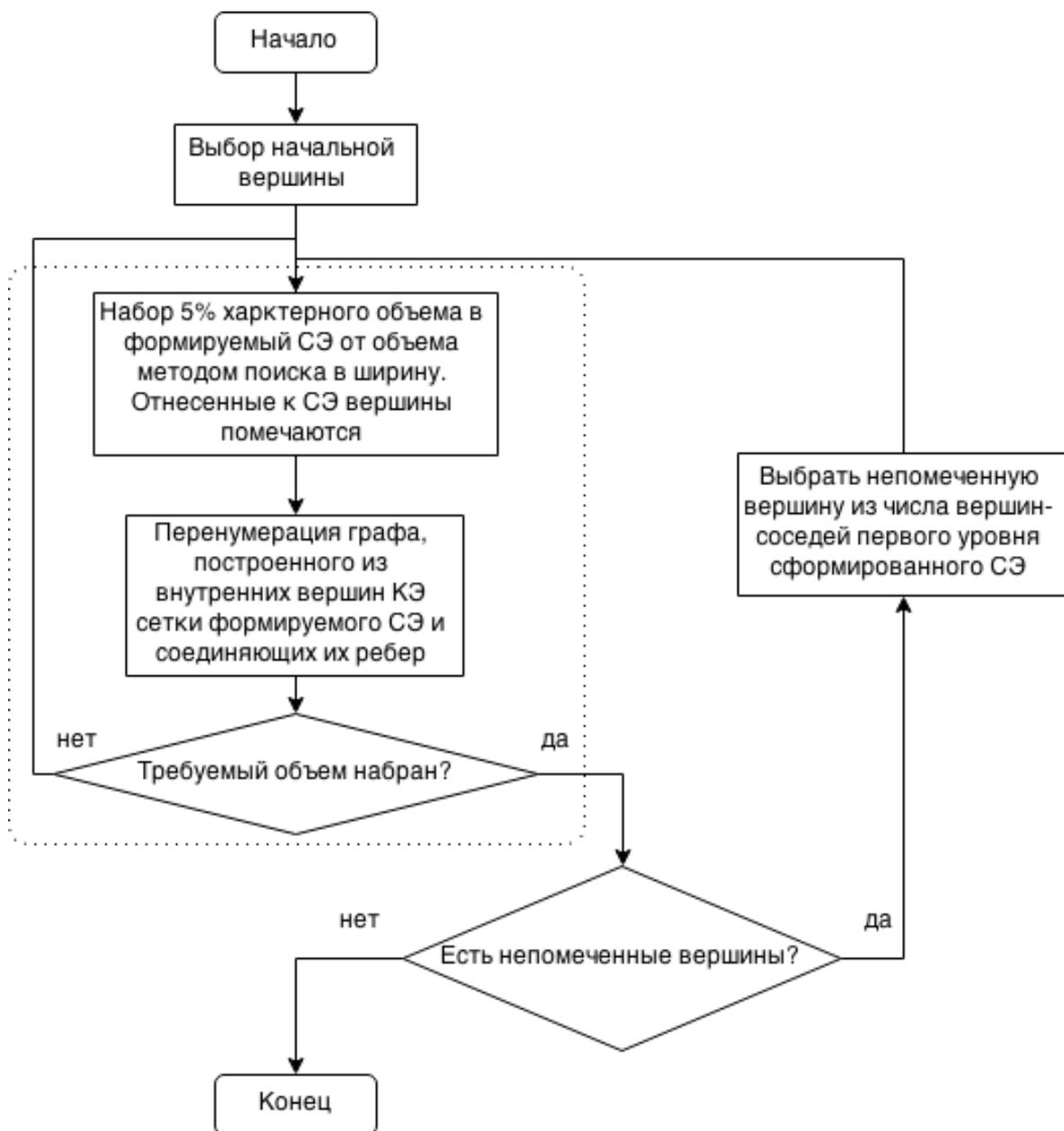


Рис. 2. Блок-схема алгоритма разделения по объему

Перенумерация выполняется по алгоритму Катхилла-Макки (Cuthill–McKee [3]) для минимизации ширины ленты, объема хранения матрицы и времени решения СЛАУ методом Холецкого в процессе последующей обработки СЭ.

3. Исследование эффективности разбиения

3.1. Программная реализация

Рассмотренные алгоритмы реализованы на языке C++ (компилятор Clang x86_64 версия 2.6 [17]) с использованием средств библиотек Qt (версия 5.1.1) [18] и STL (стандарт C++98 [19]). Визуализация результата проведена в программе ParaView (версия 4.2.0) [20].

3.2. Модели для исследования эффективности разбиения

Для исследования эффективности разработанных алгоритмов были использованы две конечно-элементные модели различной размерности, показанные на рис. 3, 4. Характеристики моделей представлены в табл. 1.

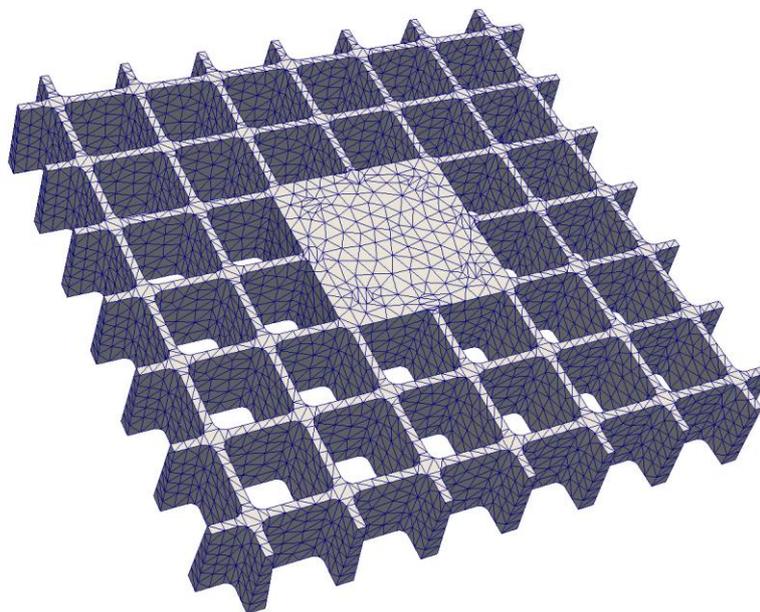


Рис. 3. Конечно-элементная модель расширительной оснастки для виброиспытаний (модель № 1)

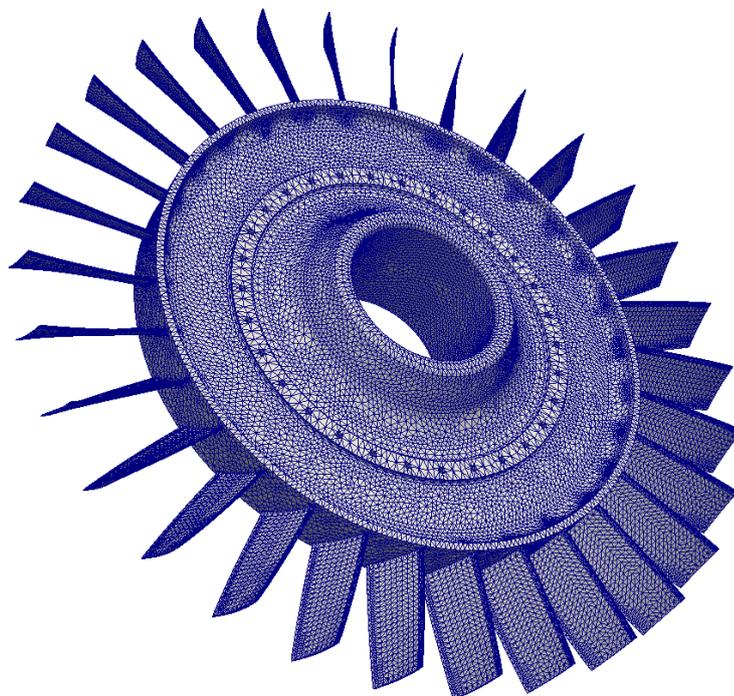


Рис. 4. Конечно-элементная модель диска компрессора с лопатками (модель №2)

Таблица 1. Характеристики моделей

	Модель №1	Модель №2
Число вершин.	51 319	502 116
Число конечных элементов	27 509	278 807
Тип конечных элементов	10-ти узловой квадратичный тетраэдр	

3.3. Методика исследования эффективности разбиения

Ниже для каждого алгоритма приведены примеры разделения на СЭ с указанием параметров алгоритмов и числа полученных СЭ. Также представлены таблицы, в которых показано распределение числа узлов и объемов матриц исключаемых узлов СЭ, а также неравномерность такого распределения. Неравномерность A_{MB} (для объема) и A_N (для числа узлов) вычисляем как отношение соответствующей характеристики СЭ к заданному или среднему ее значению. Разброс характеристики по СЭ можно оценить, найдя отношение максимального и минимального коэффициентов неравномерности этой характеристики: $\Omega = \max(A) / \min(A)$. Для модели №2 приведена только часть таблицы, в силу ее большого размера.

3.4. Результаты исследования эффективности разбиения

В результате разделения модели № 1 по числу вершин (заданное характерное число – 5000 вершин на СЭ) получено 10 СЭ (рис. 5). Параметры полученных СЭ представлены в табл. 2.

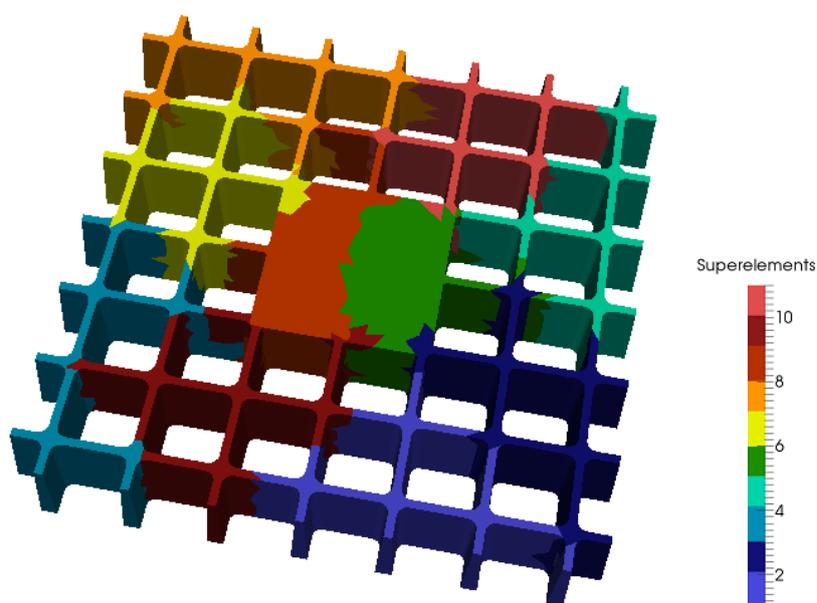


Рис. 5. Результат выделения суперэлементов в модели №1 с использованием алгоритма разделения по числу вершин

Таблица 2. Результаты обработки модели №1 алгоритмом разделения по числу вершин

№ СЭ	Объем матрицы внутренних узлов в мегабайтах	Λ_{MB} (отношение к среднему объему)	Число узлов	Λ_N (отношение к заданному характерному числу узлов)
1	17,64	1,282	5041	1,008
2	9,11	0,661	5185	1,037
3	12,33	0,895	5157	1,031
4	13,02	0,946	5004	1,001
5	8,37	0,608	5002	1,000
6	20,47	1,487	5000	1,000
7	5,28	0,384	5000	1,000
8	25,98	1,887	5085	1,017
9	14,61	1,062	5845	1,169
10	10,78	0,783	5000	1,000
$\Omega_{MB} = 4,914$			$\Omega_N = 1,169$	

В результате разделения модели № 1 по объему хранения матрицы (заданный характерный объем – 14 мегабайт на СЭ) получено 9 СЭ (рис. 6). Параметры полученных СЭ представлены в табл. 3.

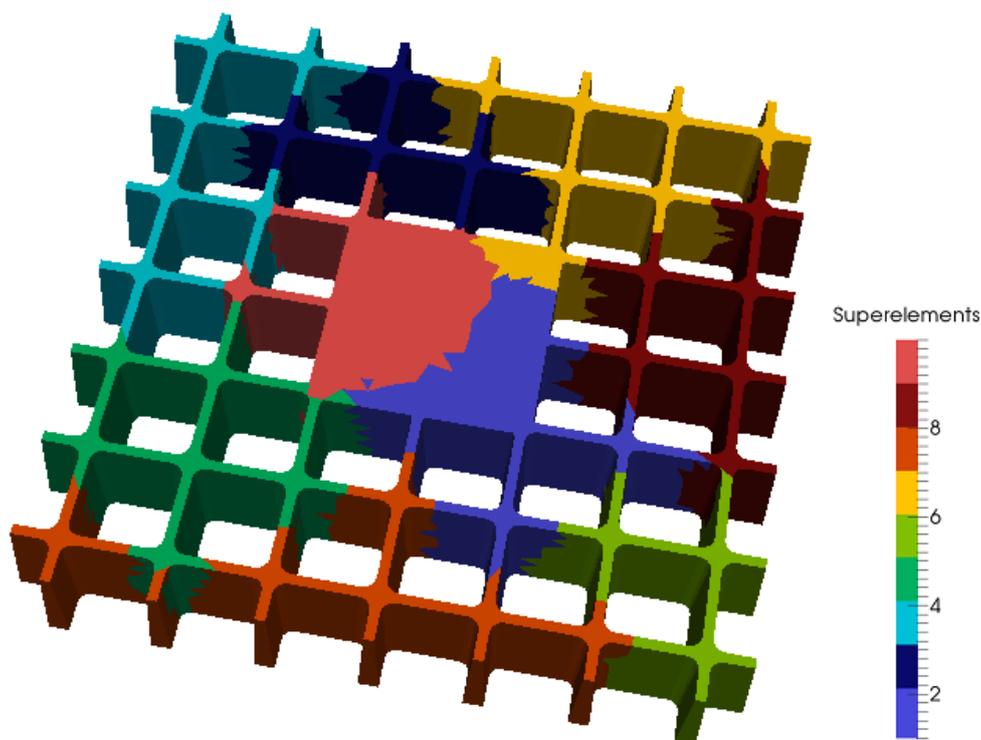


Рис. 6. Результат выделения суперэлементов в модели №1 с использованием алгоритма разделения по объему

Таблица 3. Результаты обработки модели №1 алгоритмом разделения по объему хранения матрицы

№ СЭ	Объем матрицы внутренних узлов в мегабайтах	Λ_{MB} (отношение к заданному характерному объему)	Число узлов	Λ_N (отношение к среднему числу узлов)
1	18,05	1,289	4164	0,730
2	16,67	1,191	6651	1,166
3	16,55	1,182	6189	1,085
4	15,30	1,093	6356	1,114
5	14,38	1,027	8238	1,444
6	16,81	1,201	3909	0,685
7	16,61	1,186	5894	1,033
8	14,07	1,005	5370	0,941
9	23,86	1,704	4548	0,797
$\Omega_{MB} = 1,696$			$\Omega_N = 1,970$	

В результате разделения модели № 2 по числу вершин (заданное характерное число – 10000 вершин на СЭ) получено 58 СЭ (рис. 7). Параметры полученных СЭ представлены в табл. 4.

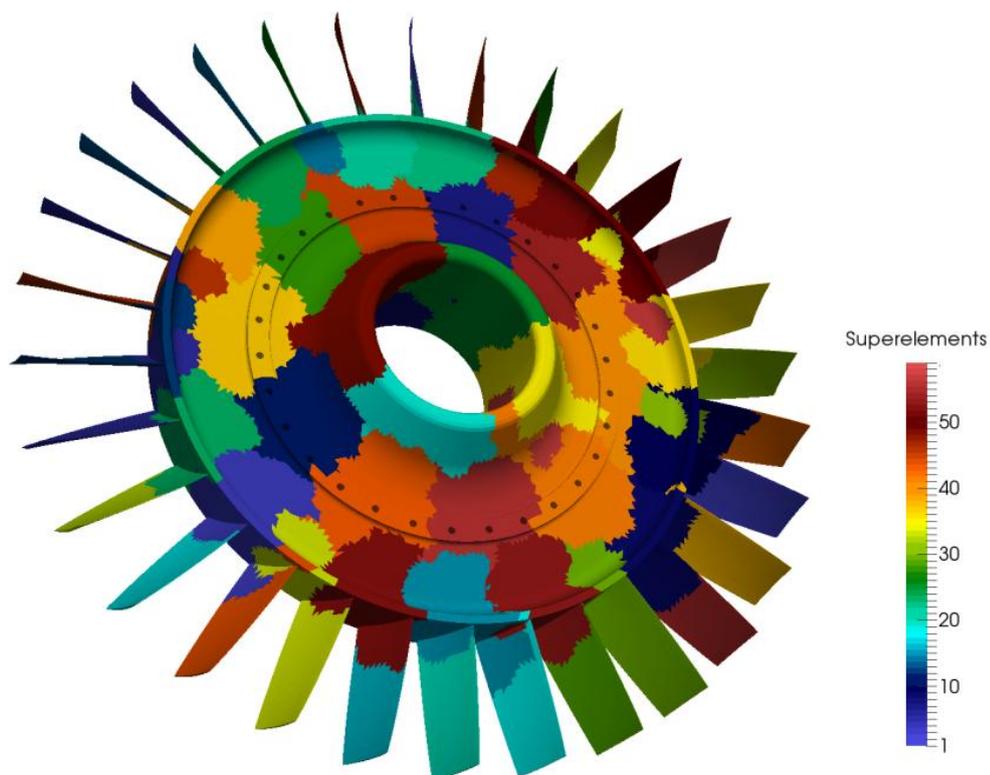


Рис. 7. Результат выделения суперэлементов в модели №2 с использованием алгоритма разделения по числу вершин

Таблица 4. Результаты обработки модели №2 алгоритмом разделения по числу вершин

№ СЭ	Объем матрицы внутренних узлов в мегабайтах	Λ_{MB} (отношение к среднему объему)	Число узлов	Λ_N (отношение к заданному характерному числу узлов)
1	71,95	2,045	10039	1,003
2	76,97	2,188	10006	1,001
3	21,29	0,605	12137	1,213
4	8,47	0,240	5615	0,561

55	63,47	1,804	13051	1,305
56	8,43	0,239	5366	0,536
57	0,96	0,027	3653	0,365
58	10,12	0,287	6827	0,682
	$\Omega_{MB} = 96,235$		$\Omega_N = 3,633$	

В результате разделения модели № 2 по объему хранения матрицы (заданный характерный объем – 34 мегабайта на СЭ) получено 42 СЭ (рис. 8). Параметры полученных СЭ представлены в табл. 5.

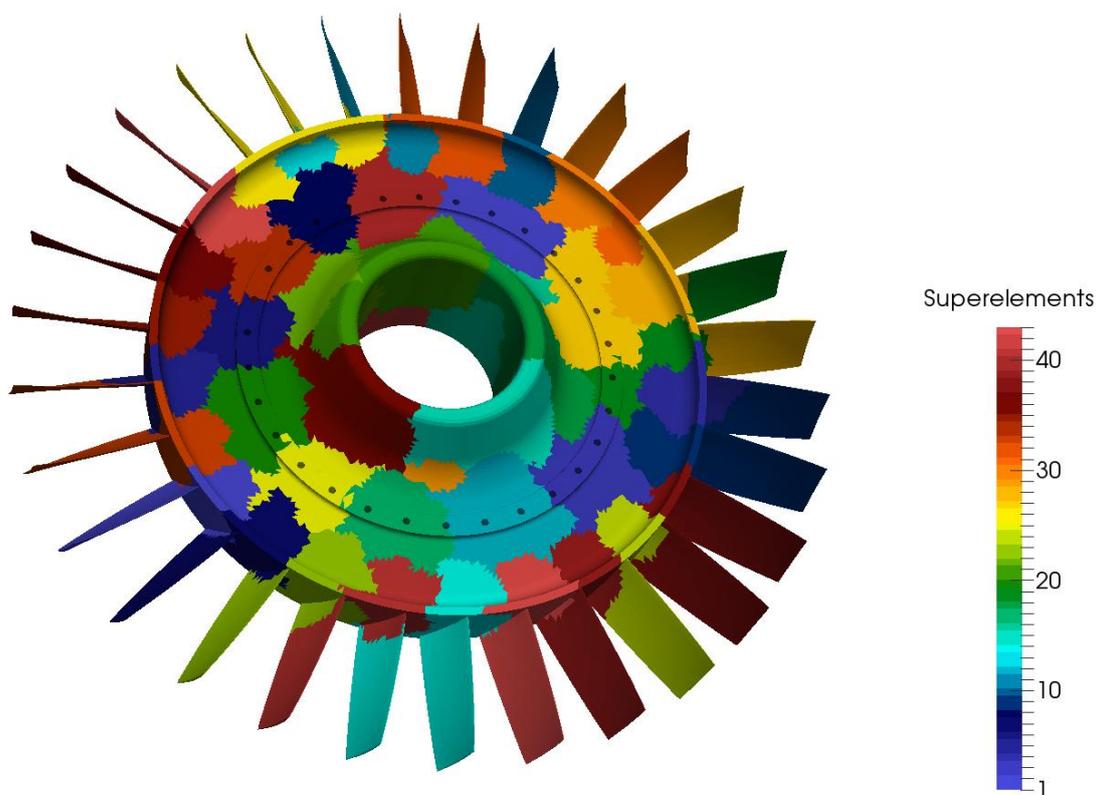


Рис. 8. Результат выделения суперэлементов в модели №2 с использованием алгоритма разделения по объему

Таблица 5. Результаты обработки модели №2 алгоритмом разделения по объему хранения матрицы

№ СЭ	Объем матрицы внутренних узлов в мегабайтах	Λ_{MB} (отношение к заданному характерному объему)	Число узлов	Λ_N (отношение к среднему числу узлов)
1	29,05	0,854	6016	0,503
2	35,89	1,055	10620	0,888
3	54,08	1,590	11977	1,001
4	50,97	1,499	11477	0,960
- - - -				
39	49,19	1,446	12076	1,010
40	39,00	1,147	10982	0,918
41	92,95	2,733	11115	0,929
42	65,53	1,927	30848	2,580
$\Omega_{MB} = 3,484$			$\Omega_N = 5,164$	

Заключение

В результате выявлено, что при применении алгоритма разделения с традиционной оценкой числа вершин как критерия окончания формирования СЭ, неравномерность полученного разбиения с точки зрения объема хранения матриц оказывается существенно более высокой (для второй модели – более, чем на порядок) по сравнению с неравномерностью по числу вершин. В то же время, алгоритм разделения использующий предложенную характеристику (объем хранения матрицы), на обеих тестовых моделях дал такие разбиения на СЭ, для которых неравномерности обеих характеристик представляют собой величины одного порядка (табл.6).

Таблица 6. Сравнительная таблица результатов тестирования

	Модель №1	Модель №2
Алгоритм разделения по числу вершин (I)	$\Omega_N = 1,169$ $\Omega_{MB} = 4,914$	$\Omega_N = 3,633$ $\Omega_{MB} = 96,235$
Алгоритм разделения по объему хранения матриц (II)	$\Omega_N = 1,970$ $\Omega_{MB} = 1,696$	$\Omega_N = 5,164$ $\Omega_{MB} = 3,484$

Результаты исследования показывают, что с точки зрения объема хранения матриц, алгоритм разделения, использующий этот параметр в качестве критерия, дает существенно меньшую неравномерность СЭ, чем алгоритм разделения по числу вершин. При этом неравномерность по числу вершин возрастает незначительно, то есть имеет место неравенство

$$\frac{\Omega_{MB}^I}{\Omega_{MB}^{II}} > \frac{\Omega_N^{II}}{\Omega_N^I}.$$

Результат, полученный при решении задачи в упрощенной постановке (только условного разбиения), позволяет сделать вывод о целесообразности оценок объемов хранения матриц при разработке алгоритмов решения задачи разбиения КЭ модели на СЭ

в комплексной постановке. Однако, предложенный алгоритм выделения СЭ по критерию объема хранения матрицы обладает заведомо меньшей производительностью, так как на каждой итерации требуется выполнять перенумерацию вершин графа. Кроме того, производительность при решении задачи (*) методом СЭ зависит не только от внутренней структуры каждого из СЭ, но и от связей между ними. Данный алгоритм не минимизирует число граничных узлов СЭ, так как задача безусловного разделения графа в данной статье сознательно не рассматривалась. Поэтому по данному критерию результат заведомо не оптимален.

Список литературы

1. Zienkiewicz O.C., Taylor R.L. The finite element method. Vol. 1: The basis. Butterworth-Heinemann, 2000. 689 p.
2. Васильев Ф.П. Методы оптимизации. М.: Факториал Пресс, 2002. 824 с.
3. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений: пер. с англ. М.: Мир, 1984. 333 с.
4. Qu Z. Model Order Reduction Techniques: with Applications in Finite Element Analysis. Springer London, 2004. P. 257-262.
5. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. Philadelphia // SIAM Journal on Scientific Computing (SISC). 1999. Vol. 20, no. 1. P. 359-392.
6. Bichot C.E., Siarry P. Graph Partitioning: Optimisation and Applications. ISTE – Wiley, 2011. P. 13-16.
7. Gdalyahu Y., Weinshall D., Werman M. Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization // IEEE Transaction on Pattern Analysis and Machine Intelligence. 2001. Vol. 23, no. 10. P. 1053-1074. DOI: [10.1109/34.954598](https://doi.org/10.1109/34.954598)
8. Martínez A.M., Mittrapiyanuruk P., Kak A.C. On combining graph-partitioning with non-parametric clustering for image segmentation // Computer Vision and Image Understanding. 2004. Vol. 95, no. 1. P. 72-85. DOI: [10.1016/j.cviu.2004.01.003](https://doi.org/10.1016/j.cviu.2004.01.003)
9. Zhao Y., Karypis G. Empirical and theoretical comparisons of selected criterion functions for document clustering // Machine Learning. 2004. Vol. 55, no. 3. P. 311-331. DOI: [10.1023/B:MACH.0000027785.44527.d6](https://doi.org/10.1023/B:MACH.0000027785.44527.d6)
10. Bichot C.E. Co-clustering documents and words by minimizing the normalized cut objective function // Journal of Mathematical Modeling and Algorithms (JMMA). 2009. Vol. 9, no. 2. P. 131-147. DOI: [10.1007/s10852-010-9126-0](https://doi.org/10.1007/s10852-010-9126-0)
11. Bichot C.E. Metaheuristics versus spectral and multilevel methods applied on an air traffic control problem // Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM), May 2006. P. 493-498.
12. Sedgewick R. Algorithms. 4th ed. Boston: Addison-Wesley Professional, 2011. 976 p.

13. Ng A.Y., Jordan M., Weiss Y. On Spectral Clustering: Analysis and an Algorithm // Proc. 14th Advances in Neural Information Processing Systems, 2001. P. 849-856.
14. Rosenberg A., Heath L. Graph Separators, with Applications. Springer US, 2002. 270 p. DOI: [10.1007/b115747](https://doi.org/10.1007/b115747)
15. Dhillon I.S., Guan Y., Kulis B. Weighted graph cuts without eigenvectors: a multilevel approach // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2007. Vol. 29, no. 11. P. 1944-1957. DOI: [10.1109/TPAMI.2007.1115](https://doi.org/10.1109/TPAMI.2007.1115)
16. Watkins D.S. Fundamentals of matrix computations. 2nd ed. New York: John Wiley & Sons, Inc., 2002. P. 60-61.
17. clang: a C language family frontend for LLVM: website. Режим доступа: <http://clang.llvm.org> (дата обращения 10.03.2015).
18. Qt: Cross-platform application & UI development framework: website. Режим доступа: <http://www.qt.io> (дата обращения 11.03.2015).
19. JTC1/SC22/WG21 - The C++ Standards Committee – ISO C++: official website. Режим доступа: <http://www.open-std.org/jtc1/sc22/wg21/> (дата обращения 10.03.2015).
20. ParaView: website. Режим доступа: <http://www.paraview.org> (дата обращения 11.04.2015).

Reducing Data Size Inequality during Finite Element Model Separation into Superelements

Yu.V. Berchun¹, M.M. Biryukova¹,

[*artem_sh@icloud.com](mailto:artem_sh@icloud.com)

I.A. Kiselev¹, A.S. Shevchenko^{1,*},

S.S. Sokolov¹

¹Bauman Moscow State Technical University, Moscow, Russia

Keywords: finite element method, superelement method, separation into superelements, superelements forming, data size

This paper is devoted to elaboration of two methods of automatic finite element model separation into superelements in order to reduce computing resources requirements (per node) for solving linear-elastic problems of solid mechanics. The first method represents an algorithm of finite element grid separation into connected subdomains by a specific preassigned number of nodes in a superelement. The second method is based on forming superelements by a specific preassigned structure size of internal nodes equilibrium equations system coefficients matrix. The considered algorithms are based on the graph theory. The evaluation of the structure size of the internal nodes equilibrium equations system coefficients matrix is based on the assumption that the Cholesky decomposition will be used for further solution. The renumbering of the internal nodes of the superelements by the Cuthill-McKee algorithm for reducing the corresponding equilibrium equations system matrix profile width and the number of non-zero elements is performed before the evaluation of the matrix structure size. The comparison of the algorithms results was made on the test samples, taking into consideration the inequality of the superelement partition in the number of nodes and in the equilibrium equations system coefficients matrix structure size. The results show that the proposed approach provides less inequality in the superelement matrix structure size whereas the inequality in the number of nodes increases low.

References

1. Zienkiewicz O.C., Taylor R.L. *The finite element method. Vol. 1: The basis*. Butterworth-Heinemann, 2000. 689 p.
2. Vasil'ev F.P. *Metody optimizatsii* [Optimization methods]. Moscow, Faktorial Press, 2002. 824 p. (in Russian).
3. George J.A., Liu J.W.-H. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, 1981. (Russ. ed.: George J.A., Liu J.W.-H. *Chislennoe reshenie bol'shikh razrezhennykh sistem uravnenii*. Moscow, Mir Publ., 1984. 333 p.).

4. Qu Z. *Model Order Reduction Techniques: with Applications in Finite Element Analysis*. Springer London, 2004, pp. 257-262.
5. Karypis G., Kumar V. A fast and high quality multilevel scheme for partitioning irregular graphs. Philadelphia. *SIAM Journal on Scientific Computing (SISC)*, 1999, vol. 20, no. 1, pp. 359-392.
6. Bichot C.E., Siarry P. *Graph Partitioning: Optimisation and Applications*. ISTE – Wiley, 2011, pp. 13-16.
7. Gdalyahu Y., Weinshall D., Werman M. Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2001, vol. 23, no. 10, pp. 1053-1074. DOI: [10.1109/34.954598](https://doi.org/10.1109/34.954598)
8. Martínez A.M., Mittrapiyanuruk P., Kak A.C. On combining graph-partitioning with non-parametric clustering for image segmentation. *Computer Vision and Image Understanding*, 2004, vol. 95, no. 1, pp. 72-85. DOI: [10.1016/j.cviu.2004.01.003](https://doi.org/10.1016/j.cviu.2004.01.003)
9. Zhao Y., Karypis G. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 2004, vol. 55, no. 3, pp. 311-331. DOI: [10.1023/B:MACH.0000027785.44527.d6](https://doi.org/10.1023/B:MACH.0000027785.44527.d6)
10. Bichot C.E. Co-clustering documents and words by minimizing the normalized cut objective function. *Journal of Mathematical Modeling and Algorithms (JMMA)*, 2009, vol. 9, no. 2, pp. 131-147. DOI: [10.1007/s10852-010-9126-0](https://doi.org/10.1007/s10852-010-9126-0)
11. Bichot C.E. Metaheuristics versus spectral and multilevel methods applied on an air traffic control problem. *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM)*, May 2006, pp. 493-498.
12. Sedgewick R. *Algorithms*. 4th ed. Boston: Addison-Wesley Professional, 2011. 976 p.
13. Ng A.Y., Jordan M., Weiss Y. On Spectral Clustering: Analysis and an Algorithm. *Proc. 14th Advances in Neural Information Processing Systems*, 2001, pp. 849-856.
14. Rosenberg A., Heath L. *Graph Separators, with Applications*. Springer US, 2002. 270 p. DOI: [10.1007/b115747](https://doi.org/10.1007/b115747)
15. Dhillon I.S., Guan Y., Kulis B. Weighted graph cuts without eigenvectors: a multilevel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, vol. 29, no. 11, pp. 1944-1957. DOI: [10.1109/TPAMI.2007.1115](https://doi.org/10.1109/TPAMI.2007.1115)
16. Watkins D.S. *Fundamentals of matrix computations*. 2nd ed. New York, John Wiley & Sons, Inc., 2002, pp. 60-61.
17. clang: a C language family frontend for LLVM: website. Available at: <http://clang.llvm.org>, accessed 10.03.2015.
18. Qt: Cross-platform application & UI development framework: website. Available at: <http://www.qt.io>, accessed 11.03.2015.
19. JTC1/SC22/WG21 - The C++ Standards Committee – ISO C++: official website. Available at: <http://www.open-std.org/jtc1/sc22/wg21/>, accessed 10.03.2015.
20. ParaView: website. Available at: <http://www.paraview.org>, accessed 11.04.2015.