

## Анализ вариантов метода Холесского для решения больших разреженных систем линейных алгебраических уравнений

к.т.н. доц. Азметов Х.Х.

ЦИАМ им. П.И. Баранова, МГТУ им. Н.Э. Баумана  
8(495)362-90-71, hakim.azmetov@mail.ru

**Аннотация.** В системе анализа напряженно-деформированного состояния при циклическом упругопластическом нагружении методом конечных элементов реализованы алгоритмы решения больших разреженных систем линейных алгебраических уравнений, основанные на методе Холесского. Проведено сравнение эффективности методов по использованию памяти и вычислительных ресурсов.

**Ключевые слова:** метод конечных элементов, СЛАУ, метод Холесского

Работа выполнена при финансовой поддержке РФФИ (грант № 15-01-03073\_А).

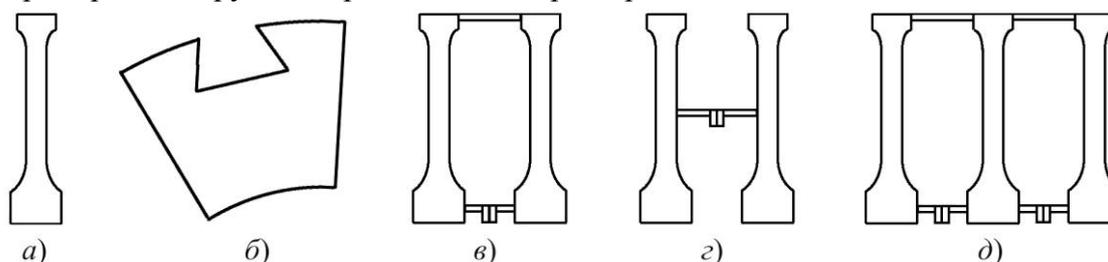
Анализ напряженно-деформированного состояния конструкций методом конечных элементов (МКЭ) с математической точки зрения сводится к решению системы линейных алгебраических уравнений (СЛАУ) [1, 2]. Из-за большой размерности получаемой СЛАУ, зависящей от количества степеней свободы решаемой задачи, именно хранение ее матрицы и собственно решение самой СЛАУ является наиболее ресурсоемкой в смысле объема требуемой памяти и процессорного времени частью решения задачи МКЭ.

Ранее в модуле решателя МКЭ «Cycle2D» [3] использовался ленточный метод хранения и решения системы линейных уравнений, который стал помехой для решения задач большой размерности и большом количестве полуциклов расчета при моделировании циклического нагружения. Если время счета на тестовом компьютере одного полуцикла расчета составляет около пяти минут при количестве уравнений порядка двадцати тысяч, то расчет тысячи полуциклов – несколько суток, что значительно затрудняет анализ конструкции. Поэтому встала необходимость модернизации модуля решателя и реализации более эффективных методов хранения матрицы жесткости конструкции и решения СЛАУ.

При модернизации модуля решателя на основе метода Холесского (см. Приложение) реализованы несколько алгоритмов решения разреженных СЛАУ [4]:

- профильный;
- минимальной степени;
- дерево факторов;
- параллельных сечений;
- вложенных сечений.

В зависимости от рассчитываемой конструкции сетки конечных элементов имеют различную структуру. Соответственно матрицы получаемой СЛАУ имеют различные характеристики (полуширину ленты, структуру заполненности и общую заполненность, т.е. количество ненулевых элементов), в предположении, что нумерация узлов сетки оптимизирована для получения диагональной СЛАУ с минимальной полушириной ленты. На рисунке 1 приведены примеры конструкций с различными характеристиками.



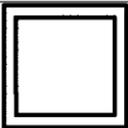
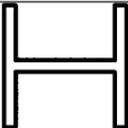
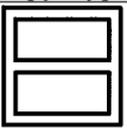
**Рисунок 1. Варианты конструкций:** а – сечение диска; б – сектор замкового соединения; в – два диска, соединенные по ободу и ступице; г – два диска, соединенные по полотну; д – три диска, соединенные по ободам и ступицам

Конструкция на рисунке 1,а обладает свойствами балочной структуры, и, соответственно, СЛАУ имеет малую полуширину и большую заполненность. Конструкция на рисунке 1,б имеет квадратную структуру, и, соответственно, большую полуширину ленты СЛАУ и малую заполненность. Остальные конструкции обладают промежуточными характеристиками.

Для тестирования алгоритмов хранения матриц СЛАУ и их решения были использованы несколько сеток конечных элементов с различной конфигурацией и, соответственно, различной полушириной и степенью заполненности матрицы. Основные характеристики сеток представлены в таблице 1. Из таблицы видно, что при практически одинаковом количестве уравнений матрицы жесткости задачи наименьшей полушириной и наибольшей заполненностью ленты обладает балочная структура, и, наоборот, наибольшей полушириной и наименьшей заполненностью – квадратная структура. Остальные структуры имеют промежуточные характеристики.

Таблица 1

**Характеристики сеток конечных элементов**

Тип задачи	1	2	3	4	5
	Балочная структура	Квадратная структура	Кольцевая структура	Тавровая структура	Замкнутая тавровая структура
					
Количество узлов	10000	9917	9918	9984	9993
Количество элементов	19356	19468	18744	18968	18810
Число уравнений	20000	19834	19836	19968	19986
Полуширина ленты	100	322	142	144	144
Заполненность ленты, %	7,37	2,30	5,12	5,06	5,04

В таблицах 2–5 и на рисунках 2–3 представлены результаты решения тестовых задач различными методами. В таблице 2 представлены объемы требуемой памяти для накладных расходов (НР), т.е. массивов индексации систем хранения матрицы жесткости, самой матрицы жесткости и всей задачи в целом для каждого из методов в единицах ячеек памяти программы (I4 – целое число длиной 32 байта, R8 – число с плавающей точкой двойной точности размером 64 байта), в таблице 3 – в процентах относительно объема всей задачи при решении ленточным методом. На рисунке 2 приведены диаграммы объема требуемой памяти, построенные по таблице 3.

Из таблиц и диаграмм видно, что наилучшие результаты по использованию памяти дает метод параллельных сечений. При этом требуемый объем памяти уменьшился от 40% на балочной структуре до 19% на квадратной структуре.

В таблице 4 представлены времена счета отдельных этапов расчета и всей задачи в целом для каждого из методов в процентах относительно этапов при решении ленточным методом. В таблице 5 представлены времена счета отдельных этапов итерации относительно времени итерации при решении ленточным методом. На рисунке 3 приведены диаграммы времен счета, построенные по данным таблицы 5. В этап подготовки входит нахождение упорядочения матрицы в соответствии с системой хранения и распределение памяти с созданием массивов индексации, в этап формирования глобальной матрицы (ГМ) – вычисление

матриц и правых частей элементов и вставка их в глобальную матрицу и правую часть, в этап решения СЛАУ – нахождение разложения матрицы и вычисление вектора неизвестных, в этап итерации – суммарное время одной итерации при решении нелинейной задачи, состоящей из формирования матрицы и решения СЛАУ, в этап задачи – суммарное время решения задачи, состоящей из подготовки и нескольких итераций.

Таблица 2

**Требуемый объем памяти (в ячейках памяти)**

№№ задач		Метод решения					
		Ленточный	Профильный	Минимальной степени	Дерево Факторов	Параллельных сечений	Вложенных сечений
1	НР [I4]	0	80 001	276 472	238 540	173 887	270 272
	Матрица [R8]	2 000 000	1 431 489	1 026 602	795 814	627 570	1 015 170
	Задача [R8]	2 211 168	1 682 657	1 376 006	1 126 252	925 681	1 361 474
2	НР [I4]	0	79 337	282 825	236 950	168 411	277 247
	Матрица [R8]	6 386 548	4 065 004	1 384 663	2 100 015	977 671	1 234 435
	Задача [R8]	6 597 661	4 315 786	1 737 189	2 429 603	1 272 990	1 584 172
3	НР [I4]	0	79 345	269 131	235 724	172 115	259 568
	Матрица [R8]	2 816 712	1 509 479	855 284	831 316	550 680	800 928
	Задача [R8]	3 023 552	1 755 991	1 196 689	1 156 018	843 577	1 137 552
4	НР [I4]	0	79 873	269 548	237 456	173 727	262 509
	Матрица [R8]	2 875 392	1 475 705	788 878	576 154	567 334	850 366
	Задача [R8]	3 084 222	1 724 471	1 132 482	903 712	863 027	1 190 450
5	НР [I4]	0	79 945	270 523	237 304	172 555	261 737
	Матрица [R8]	2 877 984	1 782 348	844 963	969 091	598 763	792 419
	Задача [R8]	3 085 918	2 030 255	1 188 159	1 295 677	892 975	1 131 222

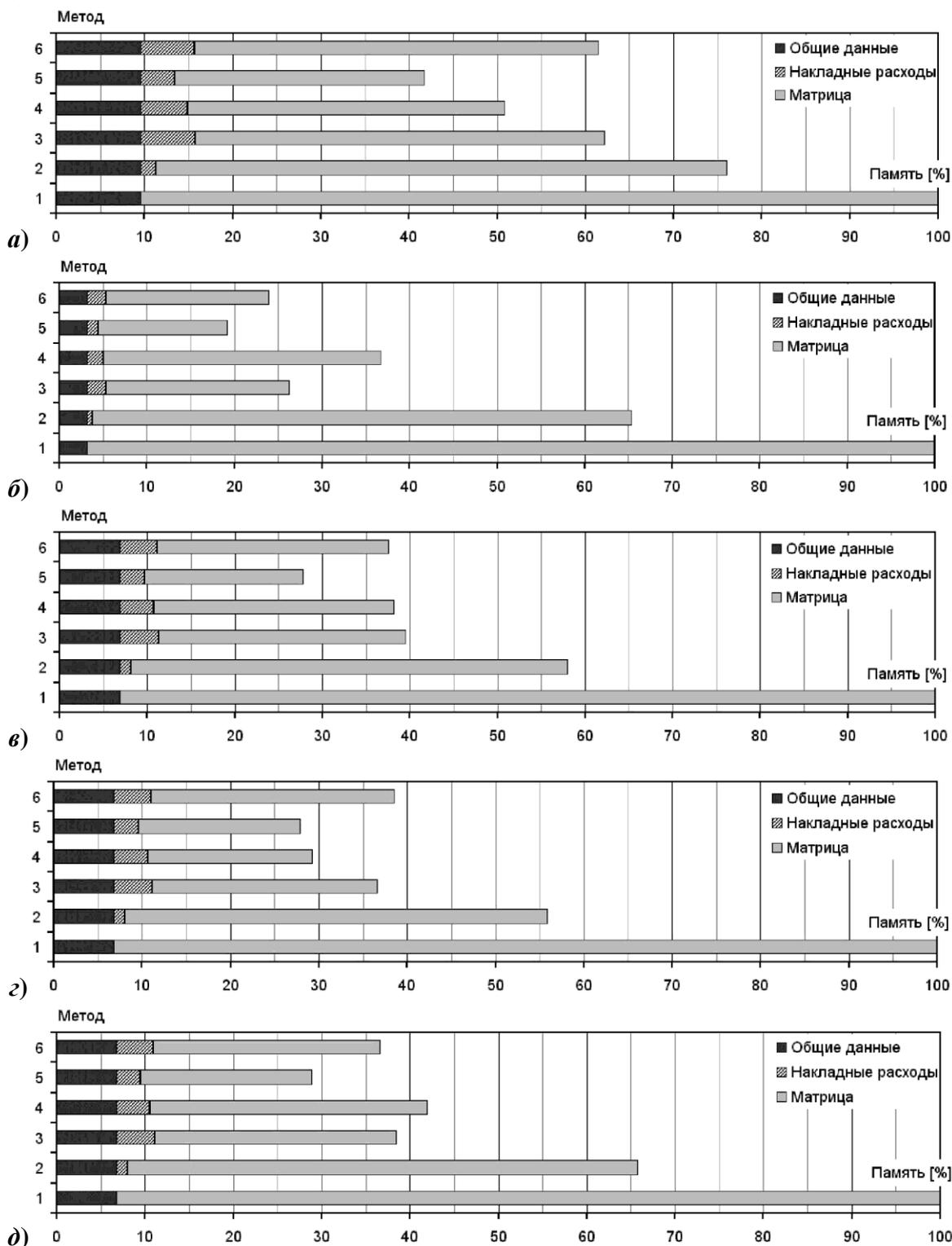
Таблица 3

**Требуемый объем памяти (в % от ленточного метода)**

№№ задач		Метод решения					
		Ленточный	Профильный	Минимальной степени	Дерево Факторов	Параллельных сечений	Вложенных сечений
1	НР	0	2	6	5	4	6
	Матрица	90	65	46	36	28	46
	Задача	100	76	62	51	42	62
2	НР	0	1	2	2	1	2
	Матрица	97	62	21	32	15	19
	Задача	100	65	26	37	19	24
3	НР	0	1	4	4	3	4
	Матрица	93	50	28	27	18	26
	Задача	100	58	40	38	28	38
4	НР	0	1	4	4	3	4
	Матрица	93	48	26	19	18	28
	Задача	100	56	37	29	28	39
5	НР	0	1	4	4	3	4
	Матрица	93	58	27	31	19	26
	Задача	100	66	39	42	29	37

Из таблиц видно, что хотя все методы проигрывают ленточному методу при подготовке решения и формировании матрицы жесткости конструкции, время решения самой системы уравнений существенно выше. Это дает большой выигрыш по времени на одной итерации

(рисунке 3), а также при решении всей задачи, особенно при нескольких итерациях, не говоря уже о нескольких сотнях и тысячах циклов. Наибольший выигрыш времени на одной итерации дает метод вложенных сечений – уменьшение времени итерации на сложных структурах до 9–29%.



**Рисунок 2. Объем памяти для различных задач: а – балочная; б – квадратная; в – кольцевая; г – тавровая; д – замкнутая тавровая.**

**Методы решения: 1 – ленточный; 2 – профильный; 3 – минимальной степени; 4 – дерево факторов; 5 – параллельных сечений; 6 – вложенных сечений**

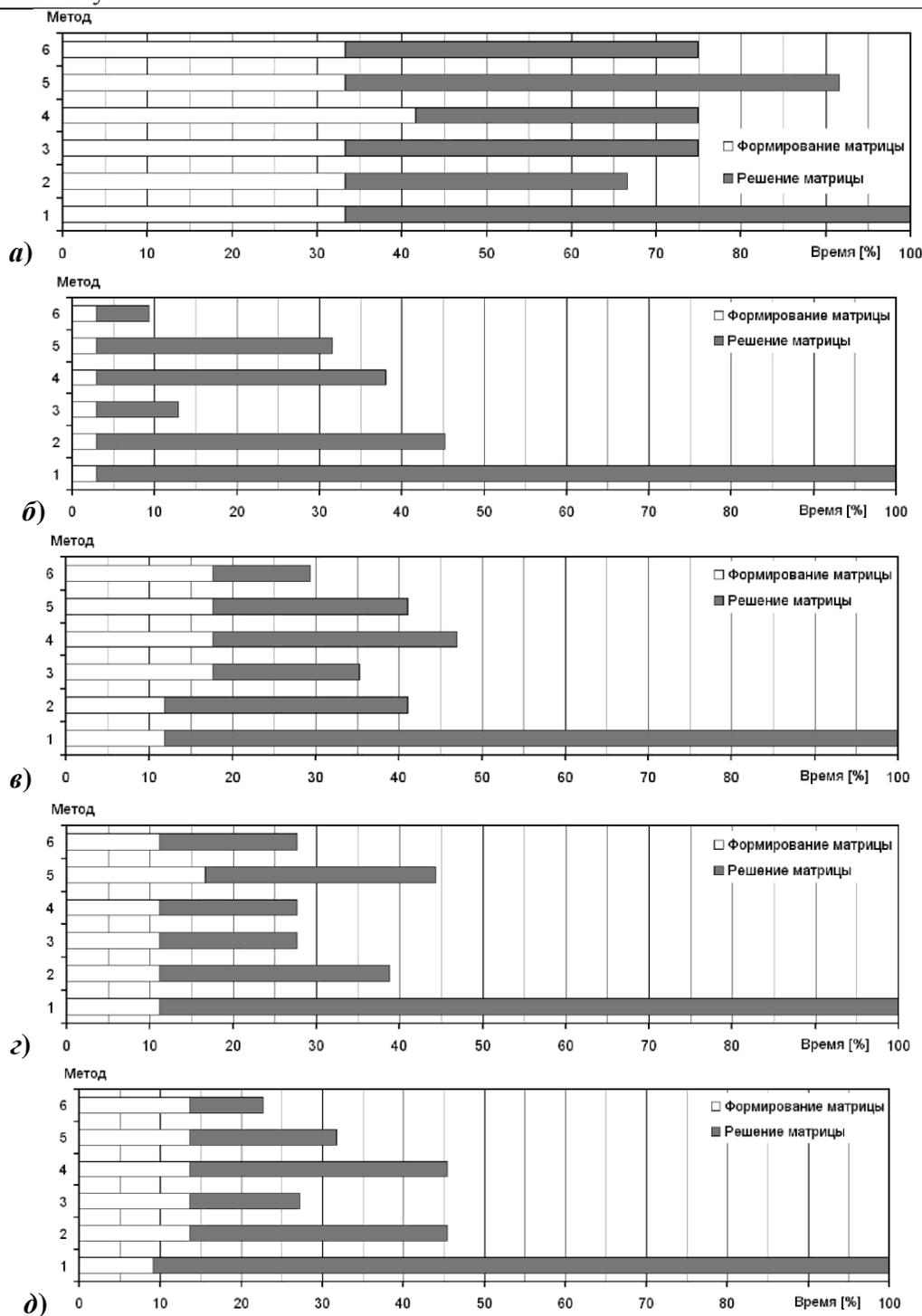
Время счета отдельных этапов (в % от ленточного метода)

№№ задач	Метод решения						
	Ленточный	Профильный	Минимальной степени	Дерево Факторов	Параллельных сечений	Вложенных сечений	
1	Подготовка	100	300	1400	200	200	200
	Формирование ГМ	100	100	100	125	100	100
	Решение СЛАУ	100	50	63	50	88	63
	Итерация	100	67	75	75	92	75
	Решение задачи	100	74	92	78	94	80
2	Подготовка	100	350	900	200	200	200
	Формирование ГМ	100	100	100	100	100	100
	Решение СЛАУ	100	44	10	36	30	7
	Итерация	100	45	13	38	32	9
	Решение задачи	100	48	17	39	33	13
3	Подготовка	100	400	1500	200	300	300
	Формирование ГМ	100	100	150	150	150	150
	Решение СЛАУ	100	33	20	33	27	13
	Итерация	100	41	35	47	41	29
	Решение задачи	100	48	41	51	41	34
4	Подготовка	100	400	1400	300	200	300
	Формирование ГМ	100	100	100	100	150	100
	Решение СЛАУ	100	31	19	19	31	19
	Итерация	100	39	28	28	44	28
	Решение задачи	100	46	35	35	45	37
5	Подготовка	100	200	750	100	100	100
	Формирование ГМ	100	150	150	150	150	150
	Решение СЛАУ	100	35	15	35	20	10
	Итерация	100	45	27	45	32	23
	Решение задачи	100	54	38	57	40	32

Таблица 5

Время одной итерации (в % от ленточного метода)

№№ задач	Метод решения						
	Ленточный	Профильный	Минимальной степени	Дерево Факторов	Параллельных сечений	Вложенных сечений	
1	Формирование ГМ	33	33	33	42	33	33
	Решение СЛАУ	67	33	42	33	58	42
	Итерация	100	67	75	75	92	75
2	Формирование ГМ	3	3	3	3	3	3
	Решение СЛАУ	97	42	10	35	29	6
	Итерация	100	45	13	38	32	9
3	Формирование ГМ	12	12	18	18	18	18
	Решение СЛАУ	88	29	18	29	24	12
	Итерация	100	41	35	47	41	29
4	Формирование ГМ	11	11	11	11	17	11
	Решение СЛАУ	89	28	17	17	28	17
	Итерация	100	39	28	28	44	28
5	Формирование ГМ	9	14	14	14	14	14
	Решение СЛАУ	91	32	14	32	18	9
	Итерация	100	45	27	45	32	23



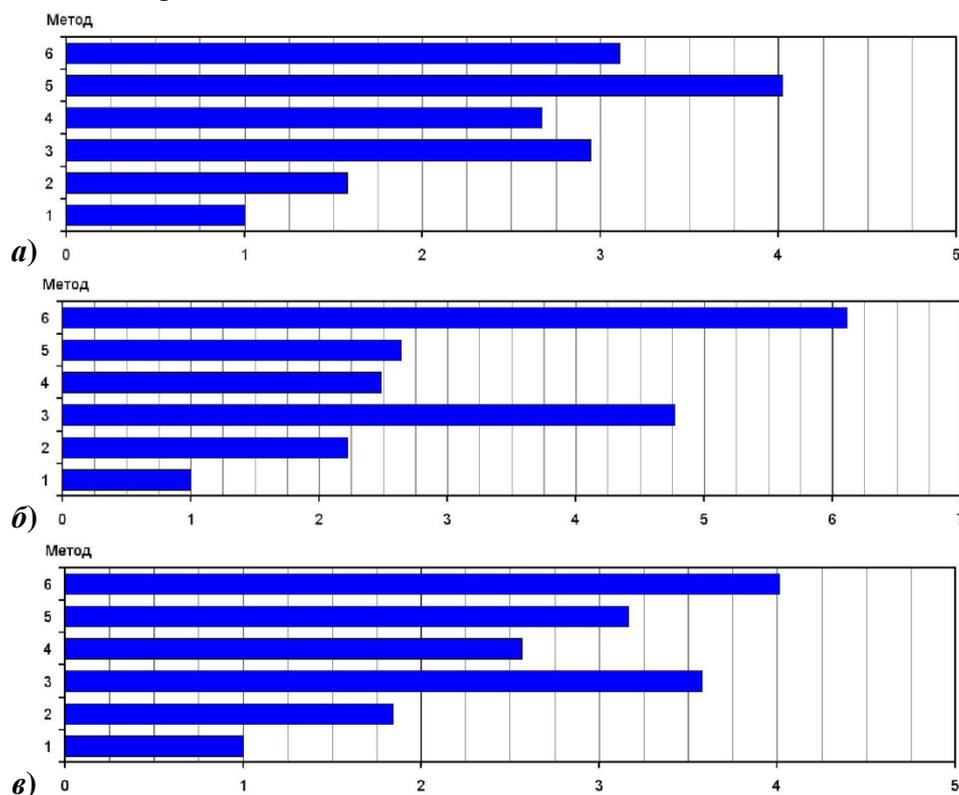
**Рисунок 3. Время счета одной итерации разными методами для различных задач (за 100% принято время одной итерации при решении ленточным методом):**  
**a – балочная; б – квадратная; в – кольцевая; г – тавровая; д – замкнутая тавровая.**  
**Методы решения: 1 – ленточный; 2 – профильный; 3 – минимальной степени; 4 – дерево факторов; 5 – параллельных сечений; 6 – вложенных сечений**

Реализованные методы показали свою высокую эффективность. В конструкциях, где заведомо хорошо работает ленточный метод, время одной итерации удалось уменьшить до 67%, а объем занимаемой оперативной памяти – до 42%. В остальных же случаях время удалось уменьшить до 9%, а использование памяти — до 19%.

На рисунке 4 представлены рейтинги для всех методов с учетом всех типов структур. В среднем наибольшую экономию памяти дает метод параллельных сечений, наибольшую

экономии времени счета одной итерации – метод вложенных сечений. На рисунке 4,в представлен обобщенный рейтинг с учетом использования памяти и времени счета одной итерации. Видно, что наибольшей эффективностью обладает метод вложенных сечений.

Таким образом, реализация приведенных методов позволяет повысить эффективность расчета более чем в 4 раза. При этом можно либо достичь уменьшения времени счета при решении задач от 10 раз, либо увеличить объем решаемых задач до 5 раз при сохранении тех же затрат машинного времени.



**Рисунок 4. Оценка эффективности реализованных методов с учетом всех типов задач: а – по использованию памяти; б – по времени счета одной итерации; в – обобщенный рейтинг. Методы решения: 1 – ленточный; 2 – профильный; 3 – минимальной степени; 4 – дерево факторов; 5 – параллельных сечений; 6 – вложенных сечений**

### Выводы

В модуле решателя МКЭ «Cycle2D» реализованы алгоритмы решения разреженных систем линейных алгебраических уравнений, что, как показано на тестовых задачах, сократило время их решения от 1,5 до 10 раз, уменьшило объем требуемой памяти от 1,5 до 5 раз, повысило среднюю эффективность решения от 2 до 4 раз по сравнению с базовым ленточным методом.

Анализ показывает, что наиболее эффективным алгоритмом для любой структуры является алгоритм вложенных сечений.

### Литература

1. Зенкевич О. Метод конечных элементов в технике. М.: Мир, 1975. 544 с.
2. Бате К.-Ю. Методы конечных элементов / Пер. с англ. В.П. Шидловского под ред. Л.И. Турчака. – М.: ФИЗМАТЛИТ, 2010. – 1024 с.
3. Темис Ю.М., Азметов Х.Х. Математическое моделирование циклического деформирования // Известия МГТУ «МАМИ» 2011, с. 195-202.
4. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. Пер. с англ. - М.: Мир, 1984. – 333 с.

### Метод Холецкого

При анализе конструкции методом конечных элементов задача сводится к решению системы линейных алгебраических уравнений (СЛАУ), занимающей более 90% времени счета. Основными свойствами матрицы данной системы является большая размерность и разреженность. Предложен ряд алгоритмов решения такой СЛАУ, описанных в работе [4].

Поскольку матрицы получаемой СЛАУ содержат большое количество нулевых элементов, первым шагом каждого алгоритма является переупорядочивание матрицы для того, чтобы собрать все ненулевые элементы вместе, исключив таким образом нулевые элементы из процесса счета. Используя разреженность, можно достигнуть больших сокращений в запросах к ресурсам памяти и вычислительной работе. Имеются различные схемы хранения разреженных матриц, отличающиеся способом использования нулей. В некоторых случаях допускается хранение части нулей в обмен на упрощение схемы хранения, в других – исключаются все нули системы.

Полученная пересортированная матрица решается методом Холецкого.

Общий алгоритм решения можно записать следующим образом:

1. *Упорядочение*: найти «хорошее» упорядочение (перестановку  $P$ ) для данной матрицы  $A$  с учетом выбранного метода хранения матрицы.
2. *Распределение памяти*: определить необходимую информацию о множителе Холецкого  $L$  матрицы  $PAP^T$  с тем, чтобы сформировать подходящую схему хранения.
3. *Разложение*: разложить переупорядоченную матрицу  $PAP^T$  в произведение  $LU$ .
4. Решение треугольных систем: решить системы  $Ly = b$  и  $L^T z = y$ .
5. Получение результата: положить  $x = P^T z$ .

Все используемые методы основаны на численном алгоритме, известном как метод Холецкого, – симметричном варианте гауссова исключения для симметричных положительно определенных матриц. Каждый метод использует его при конечном расчете результата, а также некоторые из них пользуются им в процессе сортировки.

Предположим, что система уравнений, которую нужно решить, есть:

$$Ax = b, \quad (1)$$

где:  $A$  – симметричная положительно определенная матрица коэффициентов (матрица жесткости) размерности  $N \times N$ ;  $b$  – вектор длины  $N$ , называемый правой частью;  $x$  – вектор-решение длины  $N$ , компоненты которого нужно вычислить.

Применение к матрице  $A$  метода Холецкого приводит к треугольному разложению:

$$A = LL^T, \quad (2)$$

где:  $L$  – нижняя треугольная матрица с положительными диагональными элементами.

Подставляя данное представление в систему уравнений, имеем:

$$LL^T x = b. \quad (3)$$

Замена  $L^T x = y$  показывает, что вектор  $x$  можно получить, решая треугольные системы:

$$Ly = b \text{ и } L^T x = y. \quad (4)$$

Вычислительную схему алгоритма метода Холецкого, в так называемой форме внешних произведений, можно описать в матричном виде:

$$A = H_0 = \begin{pmatrix} d_1 & v_1^T \\ v_1 & H_1 \end{pmatrix} = \begin{pmatrix} \sqrt{d_1} & 0 \\ \frac{v_1}{\sqrt{d_1}} & I_{N-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & H_1 - \frac{v_1 v_1^T}{d_1} \end{pmatrix} \begin{pmatrix} \sqrt{d_1} & \frac{v_1}{\sqrt{d_1}} \\ 0 & I_{N-1} \end{pmatrix} = L_1 \begin{pmatrix} 1 & 0 \\ 0 & H_1 \end{pmatrix} L_1^T = L_1 A_1 L_1^T, \quad (5)$$

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & H_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & d_2 & v_2^T \\ 0 & v_2 & H_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sqrt{d_2} & 0 \\ 0 & \frac{v_2}{\sqrt{d_2}} & I_{N-2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & H_2 - \frac{v_2 v_2^T}{d_2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sqrt{d_2} & \frac{v_2^T}{\sqrt{d_2}} \\ 0 & 0 & I_{N-2} \end{pmatrix} = L_2 A_2 L_2^T$$

$$, \dots, A_{N-1} = L_N I_N L_N^T,$$

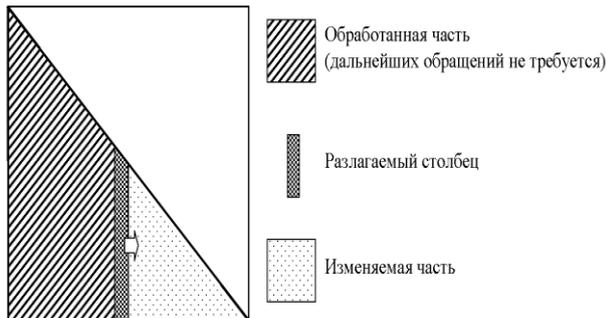
где:  $d_i$  – положительный скаляр;  $v_i$  – вектор длины  $N-i$ ; а  $H_i$  – положительно определенная симметричная матрица порядка  $N-i$  для значений  $i$  от 1 до  $N$ .

После  $N$  шагов алгоритма имеем:

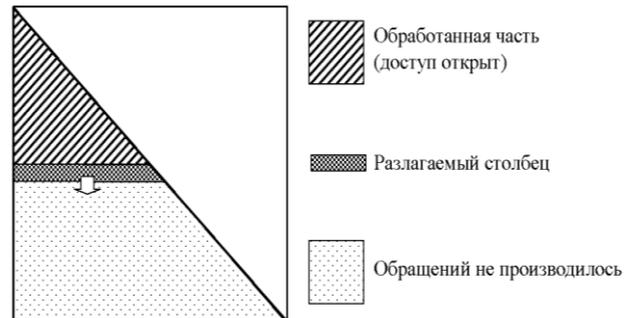
$$A = L_1 L_2 \dots L_N L_N^T \dots L_2^T L_1^T = LL^T. \quad (6)$$

Можно показать, что  $L = L_1 + L_2 + \dots + L_N - (N-1)I_N$ . Таким образом,  $i$ -й столбец  $L$  есть в точности 1-й столбец  $L_i$ .

В этой схеме вычисляются один за другим столбцы  $L$ . В то же время каждый шаг требует модификации подматрицы  $H$ , посредством внешнего произведения  $v_i v_i^T / d_i$ . Результатом является матрица  $H_i$ , т.е. как раз та матрица, которую остается разложить. Порядок обращения к элементам матрицы  $A$  в процессе разложения показан на рисунке П1.



**Рисунок П1. Порядок вычислений в форме внешних произведений**



**Рисунок П2. Порядок вычислений в методе окаймления**

Другая формулировка процесса разложения – это метод окаймления. Пусть матрица  $A$  представлена в виде:

$$A = \begin{pmatrix} M & u \\ u^T & s \end{pmatrix}, \quad (7)$$

причем уже получено симметричное разложение  $L_M L_M^T$  ведущей главной подматрицы  $M$  порядка  $N-1$ . Тогда разложением  $A$  будет:

$$A = \begin{pmatrix} L_M & 0 \\ w^T & t \end{pmatrix} \begin{pmatrix} L_M^T & w \\ 0 & t \end{pmatrix}, \quad (8)$$

где:

$$w = L_M^{-1} u \text{ и } t = s - w^T w. \quad (9)$$

Заметим, что разложение  $LL^T$  подматрицы  $M$  также можно получить приемом окаймления. Поэтому схема может быть описана следующим образом:

1. Для  $i = 1, 2, \dots, N$  решить систему:

$$\begin{pmatrix} l_{1,1} & \dots & 0 \\ & \dots & \\ l_{i-1,1} & \dots & l_{i-1,i-1} \end{pmatrix} \begin{pmatrix} l_{i,1} \\ \dots \\ l_{i,i-1} \end{pmatrix} = \begin{pmatrix} a_{i,1} \\ \dots \\ a_{i,i-1} \end{pmatrix}. \quad (10)$$

2. Вычислить  $l_{i,i} = \left( a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2 \right)^{1/2}$ .

В этой схеме поочередно вычисляются строки  $L$ . К еще не разложенной части матрицы обращение не производится до тех пор, пока не будет вычисляться соответствующая часть  $L$ . Последовательность вычислений изображена на рисунке П2.

При решении задач МКЭ со сложными граничными условиями используются множители Лагранжа [3]. Получаемые матрицы заведомо не обладают положительной определенностью из-за нулевых элементов на диагонали, и использование описанного выше алгоритма

невозможно. Но имеется возможность использовать усовершенствованный метод окаймления для решения таких матриц.

Если  $A$  – симметрическая матрица, у которой все главные миноры невырождены. Можно доказать, что для нее возможно разложение:

$$A = L \cdot \text{diag} \left( \begin{matrix} \leftarrow 1 \\ \rightarrow \end{matrix} \right) L^T, \quad (10)$$

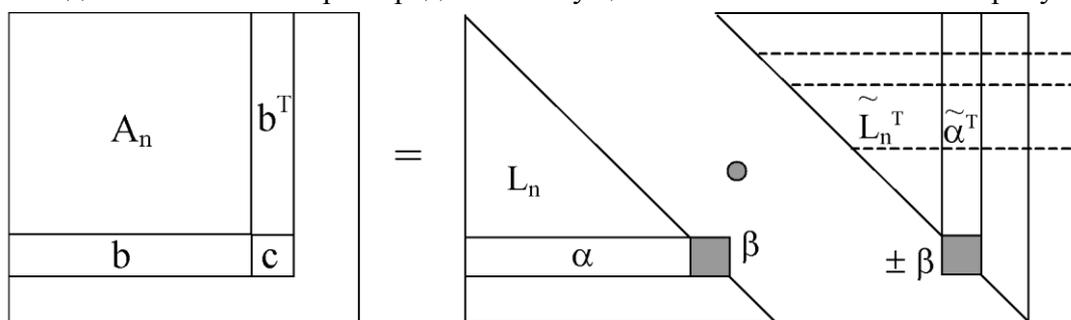
где:  $L$  – нижнетреугольная невырожденная матрица.

Это делается по следующему алгоритму. Обозначим:

$$\tilde{L}^T = \text{diag} \left( \begin{matrix} \leftarrow 1 \\ \rightarrow \end{matrix} \right) L^T, \quad (11)$$

где:  $\tilde{L}^T$  – верхнетреугольная матрица, отличающаяся от  $L^T$  только знаками в некоторых строках. Их выбор осуществляется в зависимости от знака элемента на диагонали матрицы при разложении.

Если разложение выполнено для главного минора порядка  $n$  ( $L_n \cdot \tilde{L}_n^T = A_n$ ), то подобное разложение для главного минора порядка  $n+1$  осуществляется как показано на рисунке ПЗ.



**Рисунок ПЗ. Усовершенствованный метод окаймления**

Чтобы выполнялось равенство, изображенное на рисунке ПЗ, необходимо выполнение условий:

$$\alpha \cdot \tilde{L}_n^T = b, \quad \alpha \cdot \tilde{\alpha}^T \pm \beta^2 = c. \quad (11)$$

Из данных уравнений находится сначала вектор  $\alpha$ . После его определения из  $\alpha$  и  $c$  следует найти  $\beta$ , выбрав нужный знак. От этого выбора будет зависеть, станет ли  $n+1$  строка матрицы  $\tilde{L}_n^T$  «инвертированной». Если минор  $A_{n+1} = L_{n+1} \cdot \tilde{L}_{n+1}^T$  невырожден, то  $\beta$  не может оказаться нулем.