

# **Применение метода сопряженных градиентов для определения частот и форм собственных колебаний при сейсмическом анализе зданий и сооружений на многоядерных компьютерах**

**Фиалко С. Ю.**

## Введение

- ✓ При сейсмическом анализе большеразмерных конечно-элементных расчетных моделей часто требуется определять большое количество частот и форм собственных колебаний для того, чтобы обеспечить достаточную сумму модальных масс по каждому из направлений сейсмического входа.
- ✓ Первостепенное значение имеет эффективное решение задачи об определении частот и форм собственных колебаний на доступных широкому кругу пользователей многоядерных настольных компьютерах.
- ✓ Этой задаче и посвящается данная работа.

## Постановка задачи

В результате применения МКЭ к задаче собственных колебаний возникает алгебраическая обобщенная частичная проблема на собственные значения:

**Дано:**

$$\mathbf{KV} - \mathbf{MV}\Lambda = 0, \quad \mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\}, \quad \Lambda = \left\{ \begin{array}{ccc} \lambda_1 & & \vdots \\ & \lambda_2 & \vdots \\ \dots & \dots & \dots \\ & & \vdots \\ & & \lambda_n \end{array} \right\} \quad (1)$$

$$\mathbf{K} = \mathbf{K}^T, \mathbf{K} > 0; \quad \mathbf{M} = \mathbf{M}^T, \mathbf{M} \geq 0$$

**Найти:**

$$(\lambda_1, \mathbf{v}_1), (\lambda_2, \mathbf{v}_2), \dots, (\lambda_n, \mathbf{v}_n),$$

$$n \ll N, N = \dim\{\mathbf{v}\}, 0 < \lambda_1 < \lambda_2 < \dots < \lambda_n .$$

- ✓ Наиболее широко в современных программных комплексах применяются блочный метод Ланцоша и блочный метод итерации в подпространстве, использующие на каждом шаге метод итерации обратной матрицы:

$$\mathbf{K}\mathbf{v}_i^{k+1} = \mathbf{M}\mathbf{v}_i^k \rightarrow \mathbf{v}_i^{k+1}, i \in [1, n] \quad (2)$$

- ✓ Перед началом итераций выполняется факторизация разреженной матрицы  $\mathbf{K} = \mathbf{L}\mathbf{S}\mathbf{L}^T$ , причем в том случае, если размер разреженной нижней треугольной матрицы  $\mathbf{L}$  превышает возможности оперативной памяти компьютера (типичная ситуация в случае задач размерностью 2,000,000 – 8,000,000 уравнений, решаемых на настольных компьютерах), матрица  $\mathbf{L}$  записывается на диск блок по блоку. Размер матрицы  $\mathbf{L}$  для таких задач составляет 6 – 40 Гб и более. Поэтому **при выполнении прямых и обратных подстановок на каждом шаге итерации необходимо дважды считать с диска такой объем данных – вычислительный комплекс работает со скоростью медленного диска, а не быстрого процессора.**

- ✓ Альтернативный подход: метод сопряженных градиентов с предобуславливанием (Preconditioned Conjugate Gradient – PCG).
- ✓ Преимущество: используется только оперативную память. 😊
- ✓ Недостаток: для большинства задач строительной механики вследствие плохой обусловленности происходит закиривание сходимости итерационного процесса. ☹️
- ✓ Возникает необходимость конструировать сложные схемы оператора предобуславливания и использовать технику сдвигов. Один из таких подходов опубликован в [S. Yu. Fialko, “Natural vibrations of complex bodies,” *Int. Applied Mechanics*, vol. 40, no. 1, pp. 83 – 90, 2004, <http://DOI:10.1023/B:INAM.0000023814.13805.34>].

- ✓ На сегодняшний день в качестве предобуславливания мы используем неполную факторизацию Холецкого, отбрасывая малые по абсолютной величине элементы с последующей коррекцией по А. Дженнингсу диагональных элементов для сохранения положительной определенности неполного фактора Холецкого.
- ✓ Следующим шагом к обеспечению высокой вычислительной устойчивости PCG метода и его эффективного распараллеливания на многоядерных компьютерах является разработка блочной версии – BPCG.
- ✓ Одним из таких подходов выступает local block PCG method (LOBPCG) [A. V. Knyazev and K. Neymayr, “Efficient solution of symmetric eigenvalue problem using multigrid preconditioners in the locally optimal block conjugate gradient method,” *Electronic Transactions on Numerical Analysis*, vol. 15, pp. 38 – 55, 2003].

- ✓ Однако оказалось, что для задач строительной механики сходимость LOBPCG прекращается, как только сходится первая собственная пара, поскольку базисные векторы в итерируемом подпространстве становятся практически линейно зависимыми. Кроме того, в случае определения большого количества собственных пар размерность итерируемого подпространства увеличивается, что приводит к значительному замедлению вычислений и увеличению требуемого количества оперативной памяти.
- ✓ Цель работы - разработка блочного метода сопряженных градиентов, обладающего высокой вычислительной устойчивостью и способного составить конкуренцию блочному методу Ланцоша и блочному методу итераций в подпространстве.

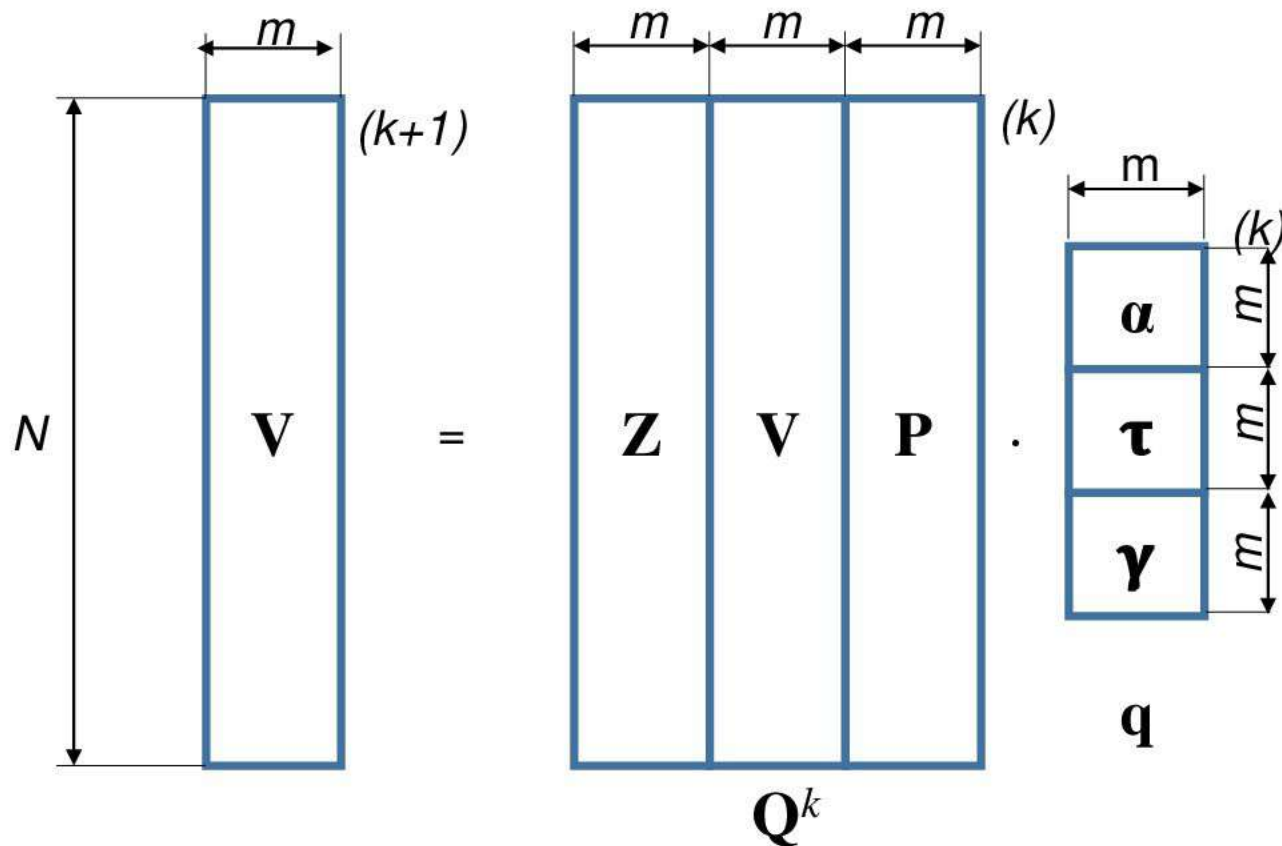
## Идея метода

- ✓ Используется аппроксимация собственных векторов и векторов сопряженных направлений, предложенная в [А. V. Knyazev and K. Neumaier]:

$$\begin{cases} \mathbf{v}_i^{k+1} = \sum_{j=1}^m \alpha_j^k \mathbf{z}_j^k + \sum_{j=1}^m \tau_j^k \mathbf{v}_j^k + \sum_{j=1}^m \gamma_j^k \mathbf{p}_j^k \\ \mathbf{p}_i^{k+1} = \sum_{j=1}^m \alpha_j^k \mathbf{z}_j^k + \sum_{j=1}^m \gamma_j^k \mathbf{p}_j^k \end{cases}, \quad i \in [1, m] \quad (3)$$

- ✓ Матрица проекции  $\mathbf{Q}^k = \{\mathbf{Z}^k \mathbf{V}^k \mathbf{P}^k\}$  размерностью  $N \times 3 \cdot m$ , где  $m$  - размерность блока, составлена из векторов  $\mathbf{Z}^k = \{\mathbf{z}_1^k, \mathbf{z}_2^k, \dots, \mathbf{z}_m^k\}$ ,  $\mathbf{V}^k = \{\mathbf{v}_1^k, \mathbf{v}_2^k, \dots, \mathbf{v}_m^k\}$  и  $\mathbf{P}^k = \{\mathbf{p}_1^k, \mathbf{p}_2^k, \dots, \mathbf{p}_m^k\}$ . Здесь  $\mathbf{v}_j^k$  - аппроксимации собственных векторов,  $\mathbf{p}_j^k$  - векторы сопряженных направлений,  $\mathbf{z}_j^k$  - векторы невязок, вычисляемых как  $\mathbf{z}_j^k = \mathbf{B}^{-1} \mathbf{r}_j^k$ ,  $\mathbf{B}$  - preconditioning operator,  $\mathbf{r}_j^k = \lambda_j^k \mathbf{M} \mathbf{v}_j^k - \mathbf{K} \mathbf{v}_j^k$ .





$$\begin{aligned}
 \mathbf{V}^{k+1} &= \mathbf{Q}^k \cdot \mathbf{q}^k, \quad \mathbf{q}^k = \{\mathbf{q}_1^k, \dots, \mathbf{q}_m^k\}, \quad \mathbf{a}_j^k = \{\alpha_{1,j}^k, \dots, \alpha_{m,j}^k\}, \\
 \mathbf{q}_j^k &= \{\mathbf{a}_j^k \ \boldsymbol{\tau}_j^k \ \boldsymbol{\gamma}_j^k\}^T, \quad j \in [1, m], \quad \boldsymbol{\tau}_j^k = \{\tau_{1,j}^k, \dots, \tau_{m,j}^k\}, \\
 & \quad \boldsymbol{\gamma}_j^k = \{\gamma_{1,j}^k, \dots, \gamma_{m,j}^k\}.
 \end{aligned} \tag{4}$$

- ✓ Подставляя (4) в (1) и умножая слева на  $\mathbf{Q}^T$ , получаем редуцированную проблему на собственные значения:

$$\mathbf{k}^k \mathbf{q}^k - \mathbf{m}^k \mathbf{q}^k \Lambda^k = 0, \quad \mathbf{k}^k = (\mathbf{Q}^k)^T \mathbf{K} \mathbf{Q}^k, \quad \mathbf{m}^k = (\mathbf{Q}^k)^T \mathbf{M} \mathbf{Q}^k. \quad (5)$$

- ✓ Подставляя  $\mathbf{q}^k$  в (4), получаем  $\mathbf{V}^{k+1}$ ,  $\mathbf{P}^{k+1} = \{\mathbf{Z}^k \cup \mathbf{P}^k\} \mathbf{q}^k$  для следующей итерации.
- ✓ Для получения  $\mathbf{Z}^{k+1}$  выполняем следующую процедуру.

Нормализация:

$$(\mathbf{V}^{k+1})^T \mathbf{M} \cdot \mathbf{V}^{k+1} = \mathbf{I}$$

Вычисляем частное Рэля:

$$\lambda_j^{k+1} = (\mathbf{v}_j^{k+1})^T \mathbf{K} \mathbf{v}_j^{k+1}$$

Определяем вектор невязки:

$$\mathbf{r}_j^{k+1} = \lambda_j^{k+1} \mathbf{M} \mathbf{v}_j^{k+1} - \mathbf{K} \mathbf{v}_j^{k+1}$$

Решаем относительно предобуславливания:  $\mathbf{B} \mathbf{z}_j^{k+1} = \mathbf{r}_j^{k+1} \rightarrow \mathbf{z}_j^{k+1}, j \in [1, m]$ .

- ✓ В отличие от [А. V. Knyazev и К. Neumaier] мы удерживаем размер блока  $m$  постоянным и независимым от количества требуемых собственных пар  $n$ . Как только очередная пара сходится, мы удаляем из блока  $Q$  все векторы  $z$ ,  $v$ ,  $p$ , соответствующие сошедшейся собственной паре, помещаем сошедшую пару в хранилище результатов и на месте удаленных векторов в блоке генерируем новое начальное приближение. И так до тех пор, пока не будут определены желаемые  $n$  собственных пар.
- ✓ Если на каком-то этапе вычислений окажется, что столбцы матрицы  $Q$  становятся почти линейно зависимыми, мы производим полную реортогонализацию столбцов, используя модифицированный алгоритм Грама-Шмидта.

## Алгоритм

- A. Инициализация. Для обеспечения равномерной вычислительной нагрузки на потоки, принимаем размер блока  $m$  кратным количеству потоков  $nr$ . Генерируем блок линейно независимых стартовых векторов  $\mathbf{V}^0$ . Принимаем  $\mathbf{P}^0 = 0$ ,  $k = 0$ .
- B. Выполняем главный итерационный цикл  $k = 0, 1, \dots$ , пока не определим требуемое количество собственных пар  $n$ .
- C. Вычисляем векторы невязки в подблоке  $\mathbf{Z}_k$

**parallel loop for  $j = 1, \dots, m$**

$$(\mathbf{v}_j^k)^T \cdot \mathbf{M} \cdot \mathbf{v}_j^k = \mathbf{I} \text{ (normalization procedure)}$$

$$\lambda_j^k = (\mathbf{v}_j^k)^T \mathbf{K} \mathbf{v}_j^k$$

$$\mathbf{r}_j^k = \lambda_j^k \mathbf{M} \mathbf{v}_j^k - \mathbf{K} \mathbf{v}_j^k$$

$$\mathbf{B} \mathbf{z}_j^k = \mathbf{r}_j^k \rightarrow \mathbf{z}_j^k$$

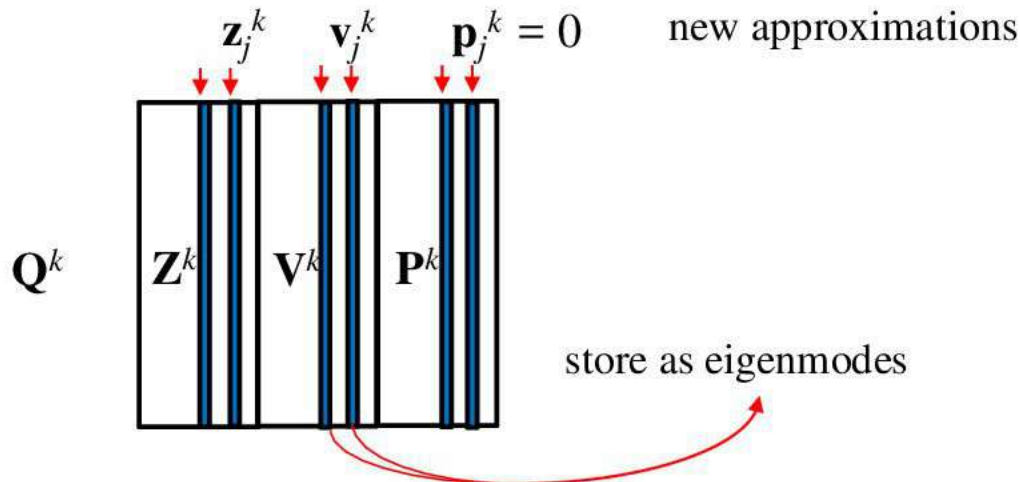
**end of parallel loop for**

## D. Проверка сходимости:

*if* (  $\|\mathbf{r}_j^k\|_2 / \lambda_j^k < tol, \forall j \in [1, m]$  )

$\{\lambda_j^k, \mathbf{v}_j^k\} \rightarrow$  помещаем в хранилище результатов и удаляем  $\mathbf{v}_j^k$  из подблока  $\mathbf{V}^k$ , генерируем новый стартовый вектор  $\mathbf{v}_j^k$  и помещаем его на место сошедшегося вектора; устанавливаем  $\mathbf{p}_j^k = \mathbf{0}$  в подблоке  $\mathbf{P}^k$  и производим ортогонализацию относительно всех сошедшихся собственных векторов. Повторяем этап С только для новых стартовых векторов и помещаем  $\mathbf{z}_j^k$  в подблок  $\mathbf{Z}^k$ .

*end if*



E. Вычисление проекций матрицы жесткости и матрицы масс на  $\text{span} \{ \mathbf{Q}^k \}$ :

*parallel region:*

$$\mathbf{m} = (\mathbf{Q}^k)^T \mathbf{M} \mathbf{Q}^k \quad (6)$$

*end of parallel region*

$$\text{Chol}(\mathbf{m}): \mathbf{m} = \mathbf{l} \cdot \mathbf{l}^T \quad (7)$$

if(! Chol( $\mathbf{m}$ ))

Реортогонализировать столбцы матрицы  $\mathbf{Q}^k$  с помощью параллельного модифицированного алгоритма Грама-Шмидта. Go to (6).

*parallel region:*

$$\mathbf{k} = (\mathbf{Q}^k)^T \mathbf{K} \mathbf{Q}^k \quad (8)$$

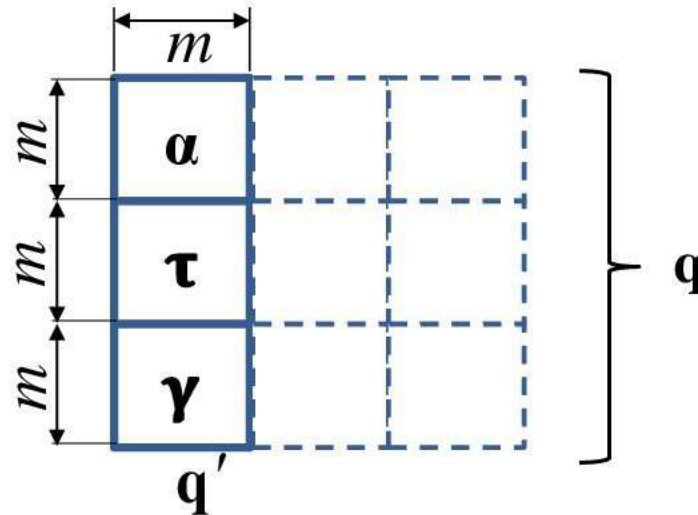
*end of parallel region*

- F. Решение обобщенной проблемы собственных значений на подпространстве:

$$\mathbf{kq}' - \mathbf{mq}'\boldsymbol{\mu} = 0. \quad (9)$$

- G. Вычисление базисных векторов для следующей итерации. [Используется параллельная версия процедуры \*dgemm\* из библиотеки Intel MKL.](#)

$$\mathbf{V}^{k+1} = \mathbf{Q}^k \mathbf{q}', \quad \mathbf{P}^{k+1} = \mathbf{Z}^k \boldsymbol{\alpha} + \mathbf{P}^k \boldsymbol{\gamma} \quad (10)$$



Н. Ортогонализация  $\mathbf{V}^{k+1}$  и  $\mathbf{P}^{k+1}$  ко всем сошедшимся собственным векторам (параллельный модифицированный метод Грама-Шмидта).



## Неполная факторизация Холецкого

- ✓ Неполная факторизация Холецкого «по величине» лежит в основе построения предобуславливания, используемого в данной работе. В процессе факторизации из нижней треугольной матрицы  $\mathbf{H}$  удаляются все «малые» по величине элементы:  $\mathbf{H}_{ij}^2 < \psi \mathbf{H}_{ii} \mathbf{H}_{jj}$ , где  $0 < \psi < 1$ , где  $\psi$  - параметр отсечения. При каждом отбрасывании элемента  $\mathbf{H}_{ij}$  производится коррекция диагональных элементов:  $\mathbf{H}_{ii} += \sqrt{(\mathbf{H}_{ii}/\mathbf{H}_{jj}) \cdot |\mathbf{H}_{ij}|}$  и  $\mathbf{H}_{jj} += \sqrt{(\mathbf{H}_{jj}/\mathbf{H}_{ii}) \cdot |\mathbf{H}_{ij}|}$ , обеспечивающая положительную определенность предобуславливания  $\mathbf{B} = \mathbf{H}\mathbf{H}^T$ .
- ✓ После завершения неполной факторизации производится пост-факторное отбрасывание: из нижней треугольной матрицы  $\mathbf{H}$  удаляются все элементы, удовлетворяющие неравенству:  $\mathbf{H}_{ij}^2 < \psi_1 \mathbf{H}_{ii} \mathbf{H}_{jj}$ , где  $\psi_1$  - пост-факторный параметр отсечения,  $0 < \psi \leq \psi_1 < 1$ . Такой подход позволяет значительно уменьшить количество ненулевых элементов в разреженной нижней треугольной матрице  $\mathbf{H}$  без существенного ущерба для способности предобуславливания ускорять сходимость.

- ✓ Чем ближе  $\psi$ , и  $\psi_1$  к нулю, тем выше способность предобуславливания ускорять сходимость, однако тем больший объем памяти требуется для хранения нижней треугольной матрицы  $\mathbf{H}$  и тем большее время требуется для решения системы уравнений  $\mathbf{Bz}^k = \mathbf{r}^k$ .
  - ✓ Задача определения частот и форм собственных колебаний требует удерживать гораздо меньшие значения  $\psi$ , и  $\psi_1$ , чем задача линейной статики. Поэтому мы предлагаем два подхода.
1. Параллельный блочный метод неполной факторизации Холецкого, представленный в [Fialko S. Yu., Karpilovskyi V. S. Block subspace projection preconditioned conjugate gradient method for structural modal analysis. *Proceedings of the Federated Conference on Computer Science and Information Systems, ISSN 2300-5963 ACSIS, Vol. 11pp. 497–506, DOI: 10.15439/2017F64* ].

2. Полная факторизация параллельным решателем PARFES ( $\psi = 0$ ) и пост-факторное отбрасывание ( $\psi_1 \neq 0$ ).

## Численные результаты

1. Рассмотрим тестовую задачу 1.

$$a = 2 \text{ м,}$$

$$E = 200\,000 \text{ МПа,}$$

$$\rho = 7\,600 \text{ кг/м}^3,$$

$$A = 0.001 \text{ м}^2,$$

$$I = 0.0001 \text{ м}^4.$$

3 собственные пары

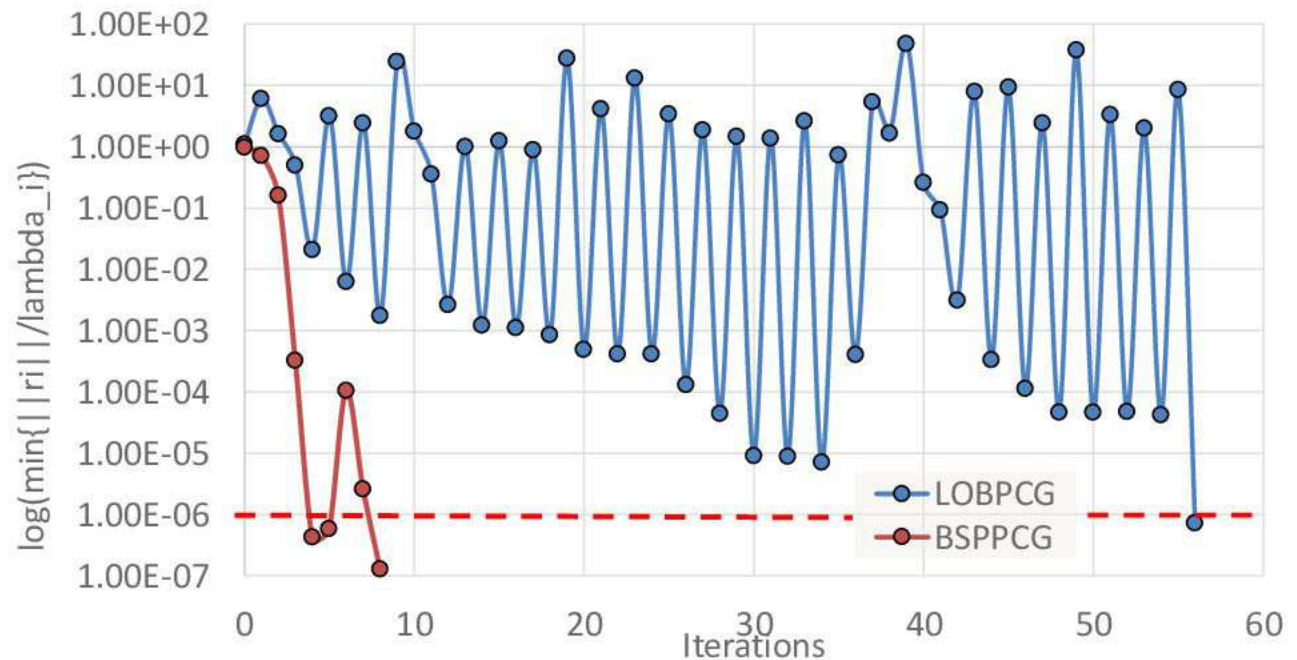
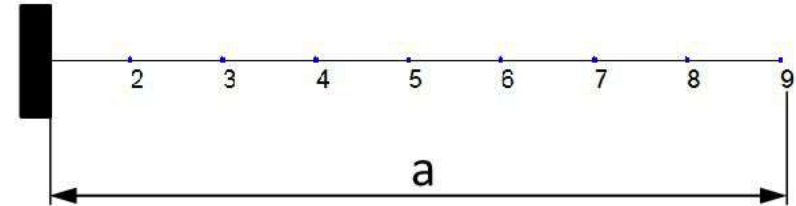
$$N = 24, n = m = 3.$$

Упорядочение MMD

$$\psi = 10^{-16}$$

$$\psi_1 = 10^{-13}$$

$$tol = 10^{-6}.$$



Сходимость LOBPCG и BSPPCG методов.

2. Задача 2. Многоэтажное здание, опирающееся на грунт (2 989 476 уравнений).

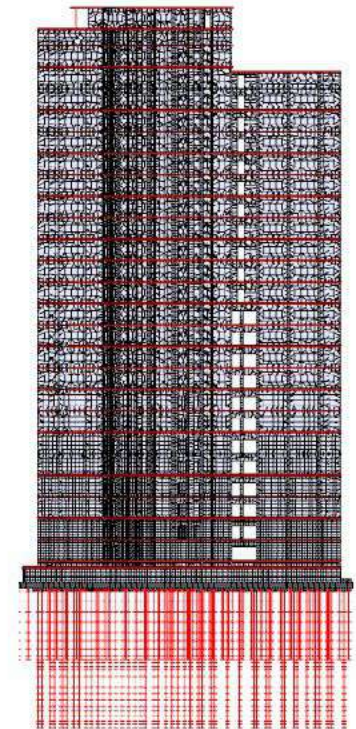
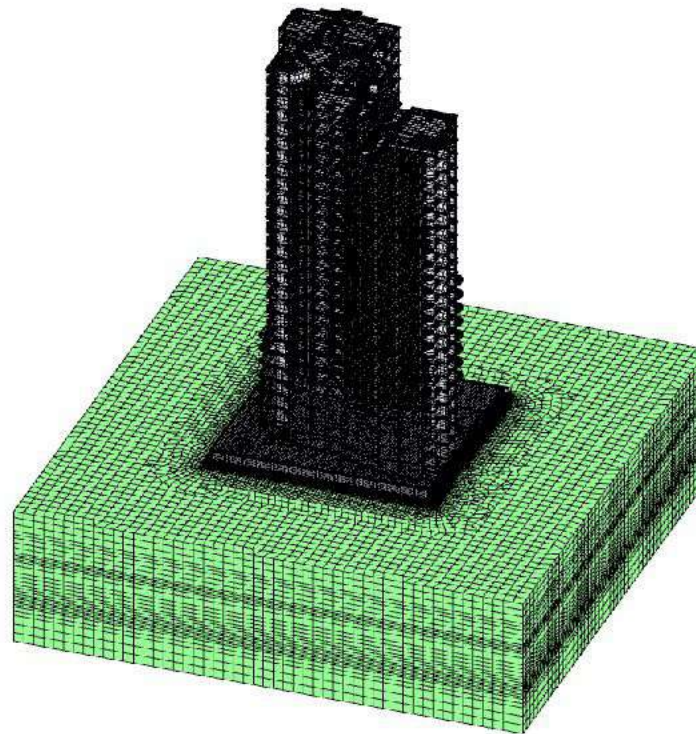
Количество частот и форм:  $n = 100$ .

Размер блока:  $m = 32$ .

Параметры  
предобуславливания:

Упорядочение METIS,  
 $\psi = 0$ ,  $\psi_1 = 10^{-13}$ .

Граница погрешности:  
 $tol = 10^{-3}$ .

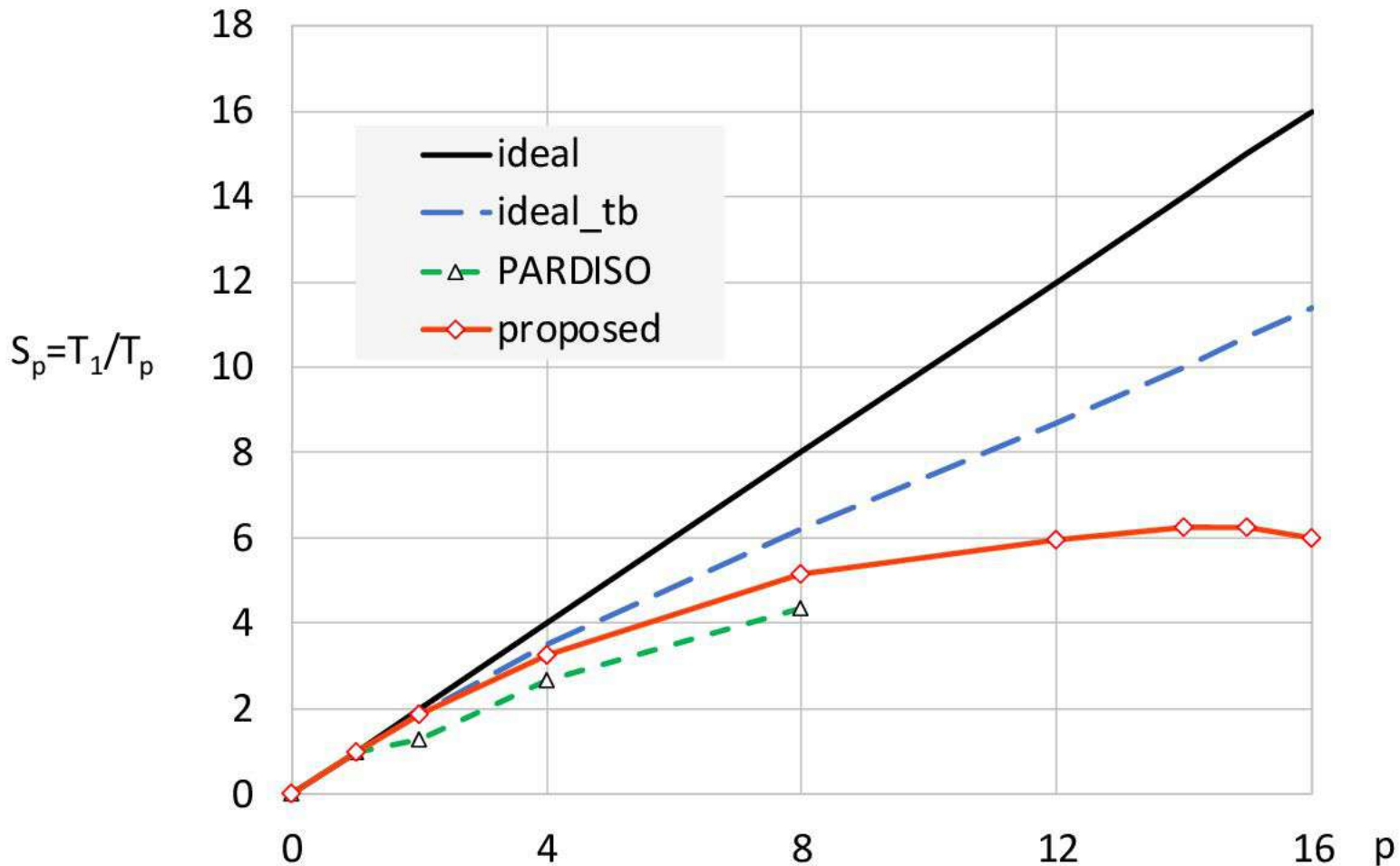


- ✓ Компьютер: 16-ядерный процессор AMD Opteron 6276, 2.3/3.2 GHz, 64 GB DDR3 RAM, OS Windows Server 2008 R2 Enterprise SP1, 64 bit.

Сравнение продолжительности факторизации Холецкого и speed up.

К-во потоков	Продолж. блочной факториз. Холецкого, с	Продолж. PARDISO (Intel MKL), с	Блочн. Факт. Холецкого, $S_p = T_1/T_p$	PARDISO, $S_p = T_1/T_p$
1	10 211	8 282	1	1
2	5 448	6 539	1.87	1.27
4	3 151	3 114	3.24	2.66
8	1 984	1 898	5.15	4.36
12	1 713	–	5.96	–
14	1 635	–	6.25	–
15	1 629	–	6.27	–
16	1 704	–	6.00	–

Сравнение speed up на этапе факторизации для блочного метода Холецкого и PARDISO.



Продолжительность вычисления различных этапов в зависимости от количества потоков.

$p$	Реортог онализ ация в блоке, $c$	Ортогон. относит. сошедших- ся пар, $c$	Вычисление векторов невязок $z$ , $c$	Вычисле ние матриц $m$ и $k$ $c$	Вычис. $V, P$ на шаге $k+1$ , $c$	Общее время решения $c$	$S_p$
1	5241	6420	40039	8515	319	62711	1
2	2384	3197	21861	5303	183	34323	1.82
4	1628	2298	12039	3621	122	20814	3.01
8	1068	2315	9031	2809	104	16390	3.82
16	620	2183	8229	2415	103	14552	4.31



Сравнение продолжительности вычисления для различных методов.  
BSPPCG – предлагаемый подход, SBLANC – блочный метод Ланцоша со  
спектральными трансформациями.

Метод	Общее время решения, s
BSPPCG (core mode, $\psi = 0$ , $\psi_1 = 10^{-13}$ )	14 552
BSPPCG (core mode, $\psi = 10^{-8}$ , $\psi_1 = 10^{-8}$ )	16 334
SBLANC (100% RAM – PARFES runs in CM)	14 096
SBLANC (50% RAM – PARFES runs in OOC)	34 660

Размер матрицы **L** составляет 36 Гб.

## Заключение

1. Вычислительная устойчивость BSPPCG метода.
2. Высокая производительность на всех этапах решения, достигнутая в результате распараллеливания всех ведущих вычислительных процедур.
3. Конкурентоспособность с блочным методом Ланцоша на многоядерных настольных компьютерах.

**СПАСИБО ЗА ВНИМАНИЕ!**