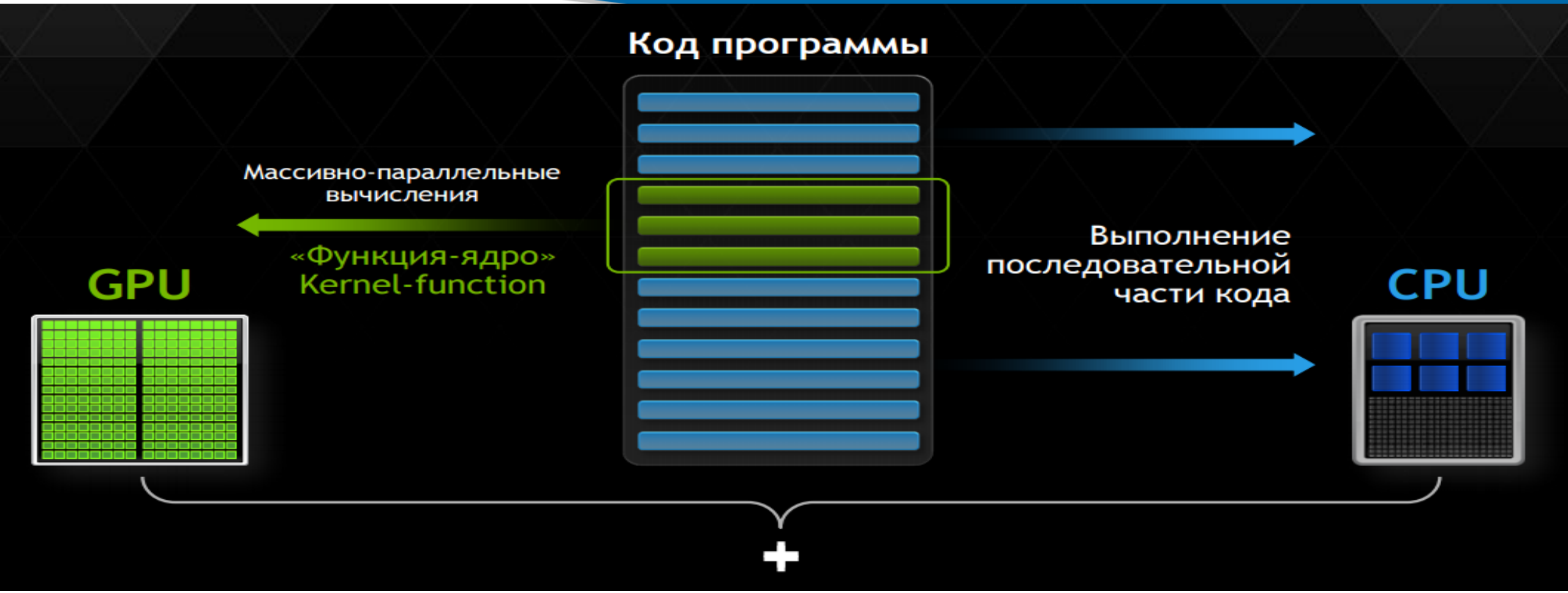
АПМ

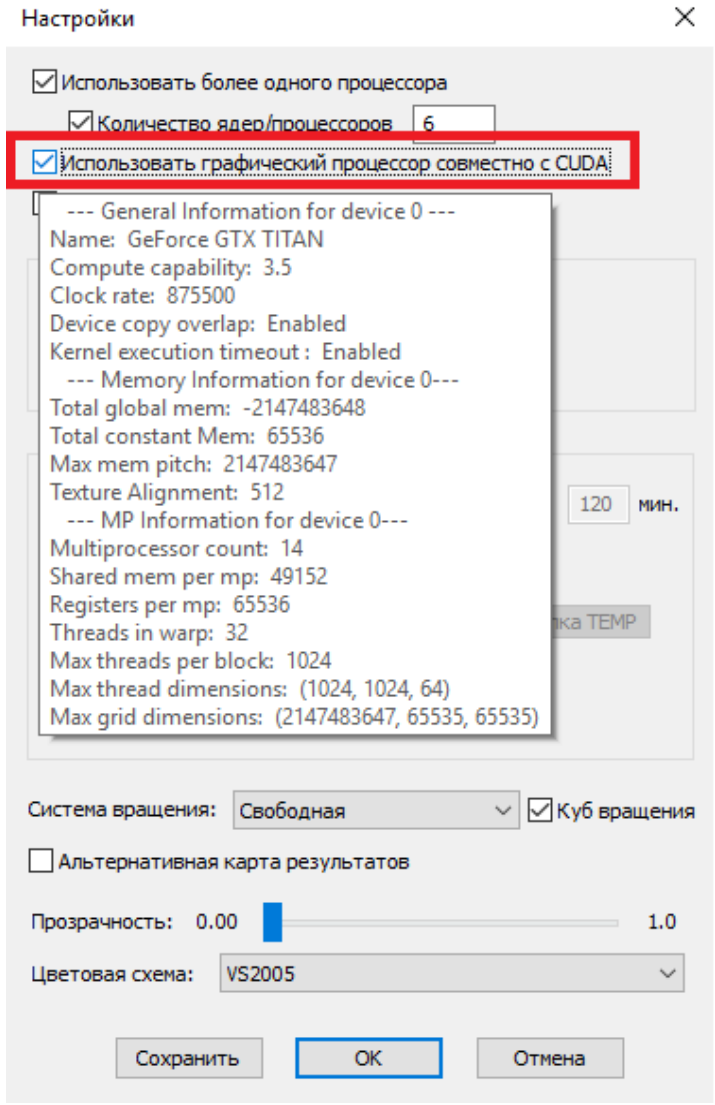
Применение технологии CUDA (NVIDIA) в программных модулях НТЦ «АПМ»

Программно-аппаратная технология CUDA

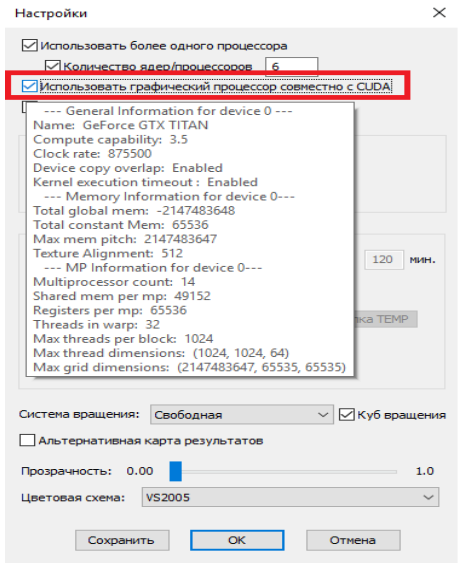


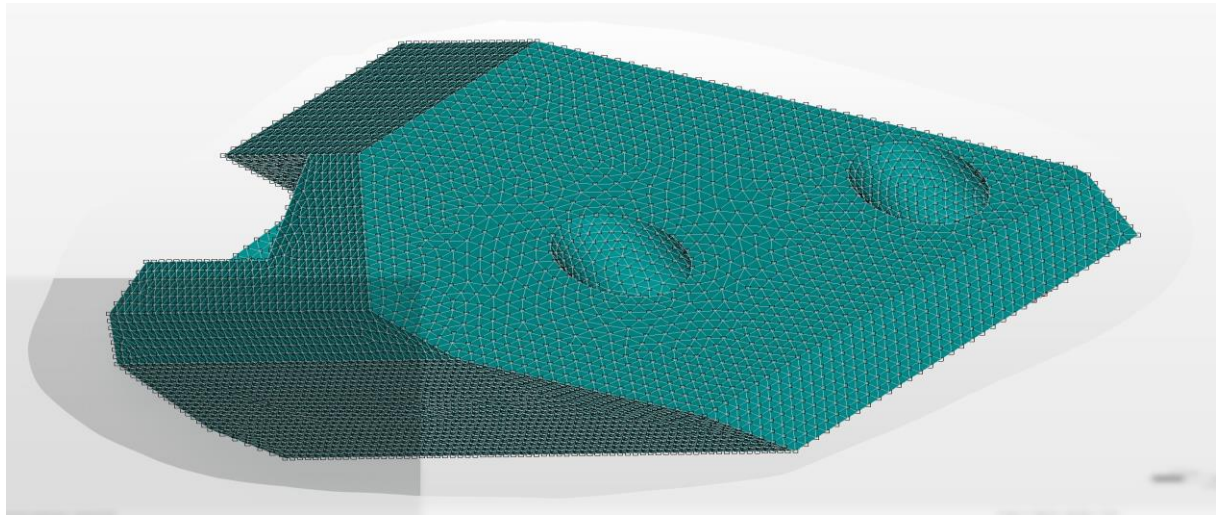
Работает только на видеокартах фирмы NVIDIA, начиная с архитектуры Kepler

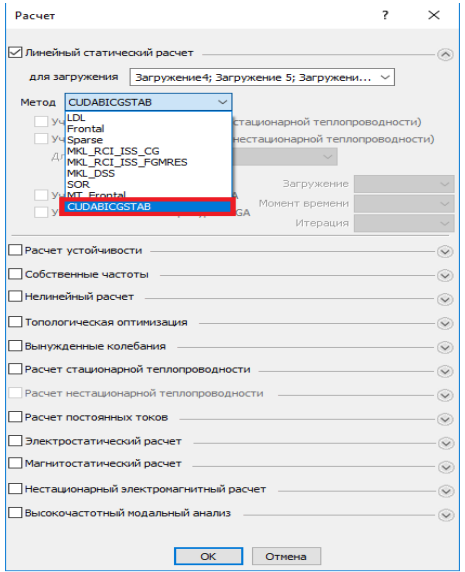
Определение доступности CUDA

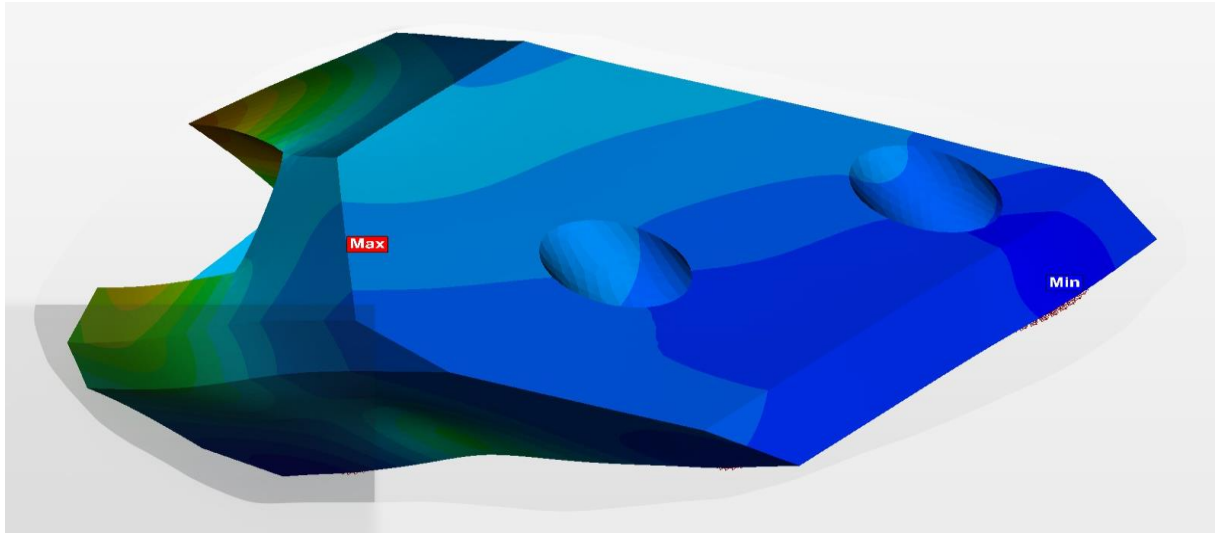


Статический расчет с несколькими загружениями









Статический расчет с несколькими загружениями

Решение задачи с несколькими загружениями:

$$[K]_{NxN}[X]_{NxM} = [F]_{NxM} \qquad (1)$$

Для ускорения сходимости применяется предобусловливание:

$$[M][K]\{X_j\} = [M]\{F_j\}, \qquad (2)$$

$$[M] = [L][U]$$

Стабилизированный метод бисопряженных градиентов с предобусловливанием:

1 Выбрать начальное приближение $\mathbf{x}_0$, вычислить $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\,\mathbf{x}_0$
2 Выбрать вектор $\tilde{\mathbf{r}}$, удовлетворяющий условию $(\mathbf{r}_0, \tilde{\mathbf{r}}) \neq 0$ (например, $\tilde{\mathbf{r}} = \mathbf{r}_0$)
3 Для $i = 1, 2, \ldots$ до сходимости или до N_{it}^{max}
4 $\rho_{i-1} = (\tilde{\mathbf{r}}, \mathbf{r}_{i-1})$
5 Если $\rho_{i-1} = 0$
6 то метод не может решить данную систему
7 Если $i = 1$
8 $\mathbf{p}_i = \mathbf{r}_{i-1}$
9 Иначе
10 $\beta_{i-1} = (\rho_{i-1} / \rho_{i-2})\,(\alpha_{i-1} / \omega_{i-1})$
11 $\mathbf{p}_i = \mathbf{r}_{i-1} + \beta_{i-1}(\mathbf{p}_{i-1} - \omega_{i-1}\,\mathbf{v}_{i-1})$
12 Найти $\tilde{\mathbf{p}}$ из системы $\mathbf{M}\,\tilde{\mathbf{p}} = \mathbf{p}_i$
13 $\mathbf{v}_i = \mathbf{A}\,\tilde{\mathbf{p}}$
14 $\alpha_i = \rho_{i-1} / (\tilde{\mathbf{r}}, \mathbf{v}_i)$
15 $\mathbf{s} = \mathbf{r}_{i-1} - \alpha_i\,\mathbf{v}_i$
16 Если $\|\mathbf{s}\|_2 / \|\mathbf{r}_0\|_2 \leq Tol$
17 то КОНЕЦ ($\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\,\tilde{\mathbf{p}}$ – полученное решение)
18 Найти $\tilde{\mathbf{s}}$ из системы $\mathbf{M}\,\tilde{\mathbf{s}} = \mathbf{s}$
19 $\mathbf{t} = \mathbf{A}\,\tilde{\mathbf{s}}$
20 $\omega_i = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$
21 $\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha_i\,\tilde{\mathbf{p}} + \omega_i\,\tilde{\mathbf{s}}$
22 $\mathbf{r}_i = \mathbf{s} - \omega_i\,\mathbf{t}$
23 Если $\|\mathbf{r}\|_2 / \|\mathbf{r}_0\|_2 \leq Tol$
24 то КОНЕЦ ($\mathbf{x}_i$ – полученное решение)
25 Увеличить i

Оценка эффективности параллельных вычислений

Ускорение (*speedup*), получаемое при использовании параллельного алгоритма для ***p*** процессоров, по сравнению с последовательным вариантом выполнения вычислений, определяется величиной:

$$S_p(n) = T_1(n) / T_p(n) \qquad (3)$$

где величина ***n*** используется для параметризации вычислительной сложности решаемой задачи и понимается как количество входных данных задачи.

Система, на которой проводились исследования:

CPU Intel i7, 4 cores, 3.4 GHZ, IvyBridge; ОП - DDR3, 16GB; NVIDIA GTX Titan Kepler, 6 GB, 2688 cores; HDD 2 Тб ; CUDA 8.0; ОС Windows 10;

Качесвто конечных элементов	Размерность матрицы	Количество загружений	Ускорение
Хорошее качество	102717x102717	10	17.58884
Хорошее качество	1075461x1075461	10	15.82659
Плохое качество	15246x15246	10	1.60072
Плохое качество	11616x11616	10	1.27866

Решение обобщённой задачи на собственные значения

Для решения обобщенной задачи поиска собственных значений и векторов:

$$[A]\{x\} = \lambda[B]\{x\} \tag{4}$$

предлагается использовать следующее преобразование:

$$[B] = [L][U]$$

$$[L]^{-1}[A][U]^{-1}\{y\} = \lambda\{y\} \tag{5}$$

$$\{y\} = [U]\{x\}$$

Обратные матрицы для L и U находятся при решении матричных уравнений:

$$[L][X] = [E]$$

$$[U][X] = [E] \tag{6}$$

$$[C] = [L]^{-1}[A][U]^{-1} \tag{7}$$

$$[C]\{y\} = \lambda\{y\}$$

Решение обобщённой задачи на собственные значения

Нахождение приближенных значений собственных чисел:

$$\left\|\{\upsilon^{(0)}\}\right\|_2 = 1;$$

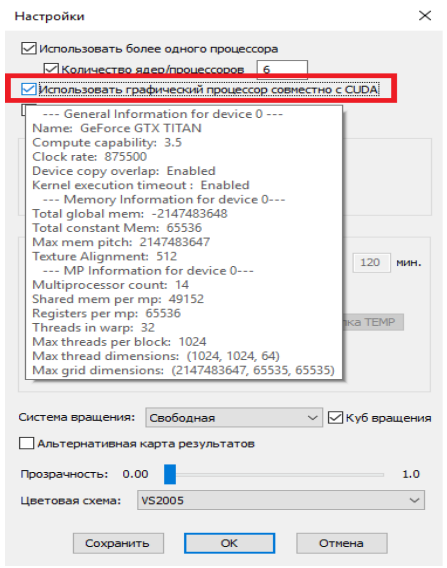
$$\begin{cases} [\mathrm{D}] = \{\upsilon^{(0)}\}\{\upsilon^{(0)}\}^{\mathrm{T}}; \\ [\mathrm{C}] = [\mathrm{C}] - \lambda_{\mathrm{k-1}}\{\mathrm{D}\}; \\ \begin{cases} \{\omega\} = [C]\{\upsilon^{(i-1)}\}, \\ \{\upsilon^{(i)}\} = \dfrac{\{\omega\}}{\left\|\omega\right\|_2}, \\ \lambda_k{}^{(i)} = \{\upsilon^{(i)}\}^T[C]\{\upsilon^{(i)}\}, \\ 0 \le i < 10; \end{cases} \\ \upsilon^{(0)} = \upsilon^{(i)}; \end{cases} \qquad (8)$$

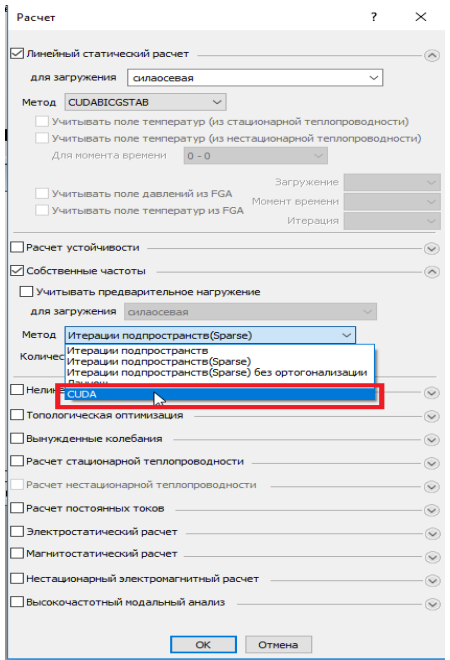
Метод обратных итераций со сдвигом:

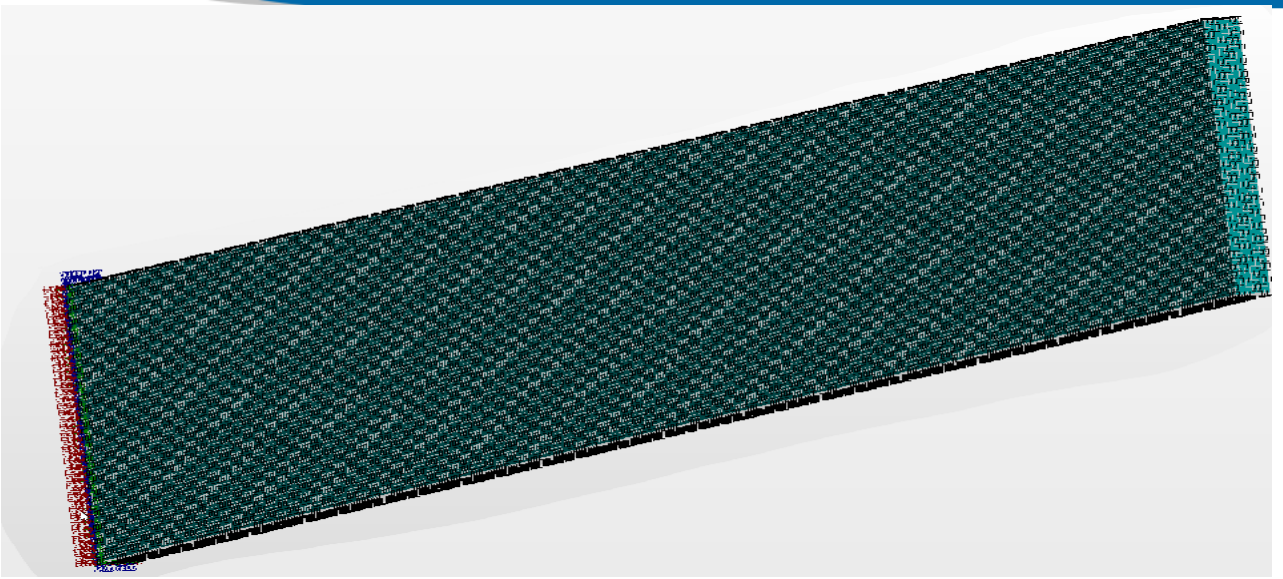
$$([A] - \lambda_i^{(s)}[E])\{y^{(s)}\} = \{x^{(s)}\},$$

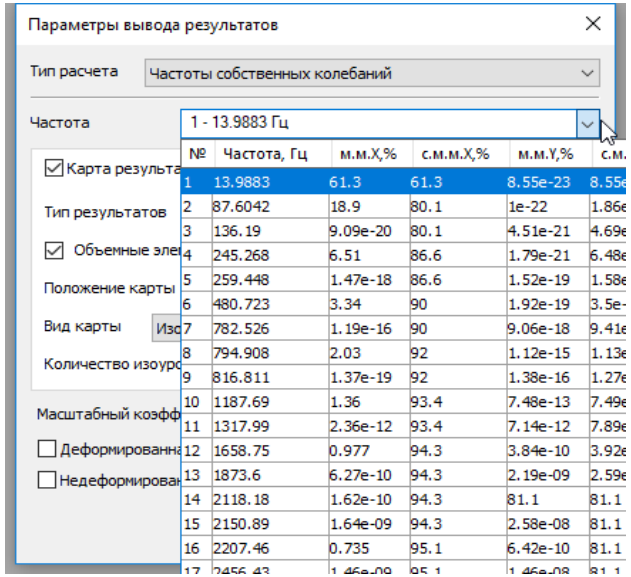
$$\lambda_i^{(s+1)} = \lambda_i^{(s)} + \frac{x_k^{(s)}}{y_k^{(s)}}, \quad \{x^{(s+1)}\} = \frac{\{y^{(s)}\}}{\left\|\{y^{(s)}\}\right\|}. \qquad (9)$$

Расчет собственных частот







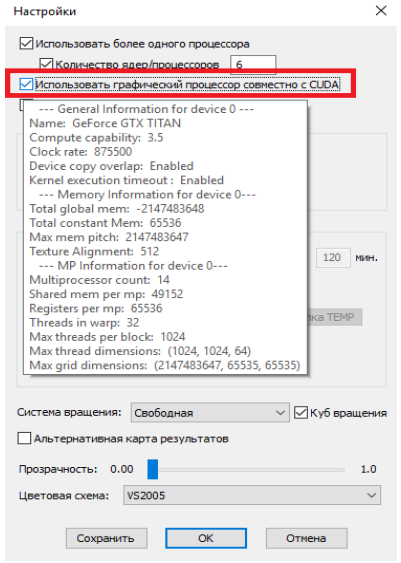


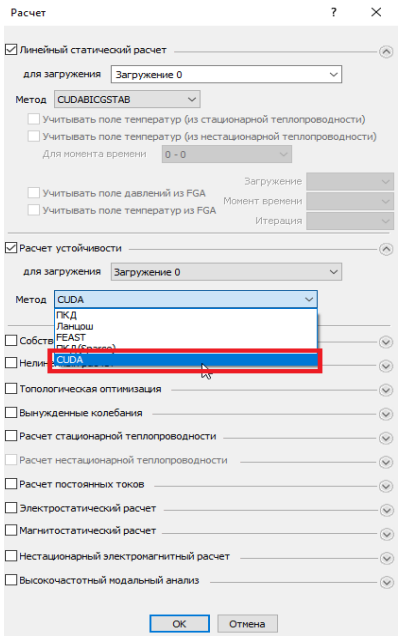
Ортогонализация векторов на матрицу

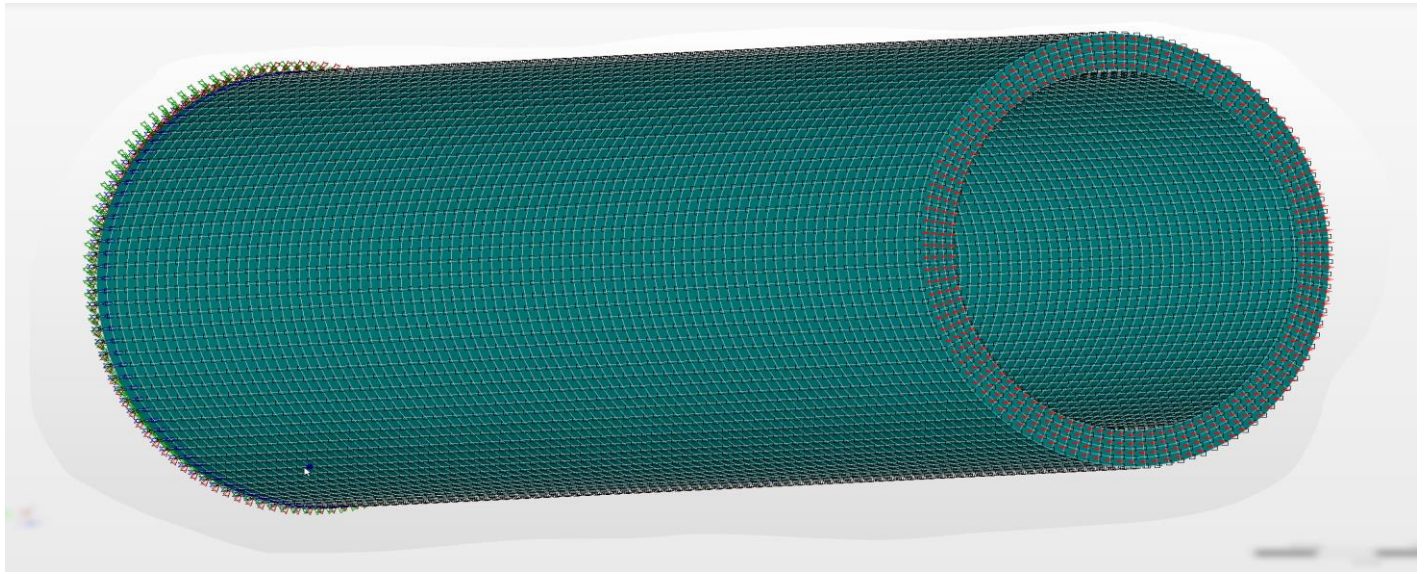
$$\{X_0\} = \frac{1}{\sqrt{\{X_0\}^T [A] \{X_0\}}} \{X_0\}, \tag{10}$$

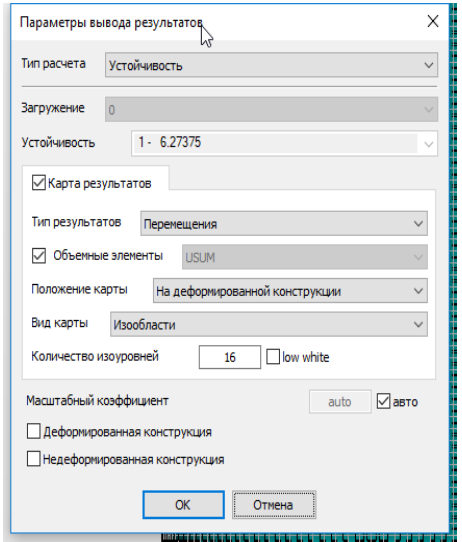
$$\begin{cases} \{X_i\} = \{X_i\} - \sum_{j=0}^{i-1} \left(\{X_j\}^T [A] \{X_i\} \{X_j\} \right), \\ \{X_i\} = \frac{1}{\sqrt{\{X_i\}^T [A] \{X_i\}}} \{X_i\}, \\ 1 \le i < m. \end{cases} \tag{11}$$

Расчет устойчивости









Определение положительной определенности матрицы

Разложение Холецкого:

$$[A]=[L][L^T] \tag{12}$$

$$\mathrm{L_{ii}}=\sqrt{A_{ii}-\sum_{k=1}^{i-1}L_{ik}^2}\,, \tag{13}$$

$$L_{ij}=\frac{1}{L_{jj}}\left(A_{ij}-\sum_{k=1}^{j-1}L_{ik}L_{jk}\right)$$

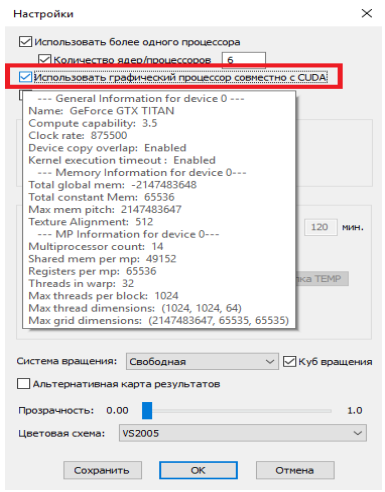
$$[A]+k*[B]=[L][L^T] \tag{14}$$

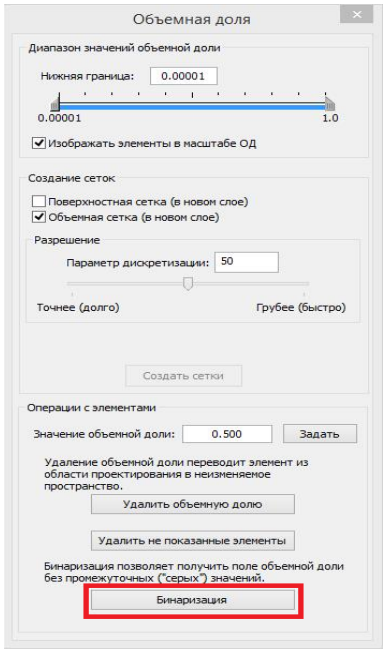
Вычислительные эксперименты

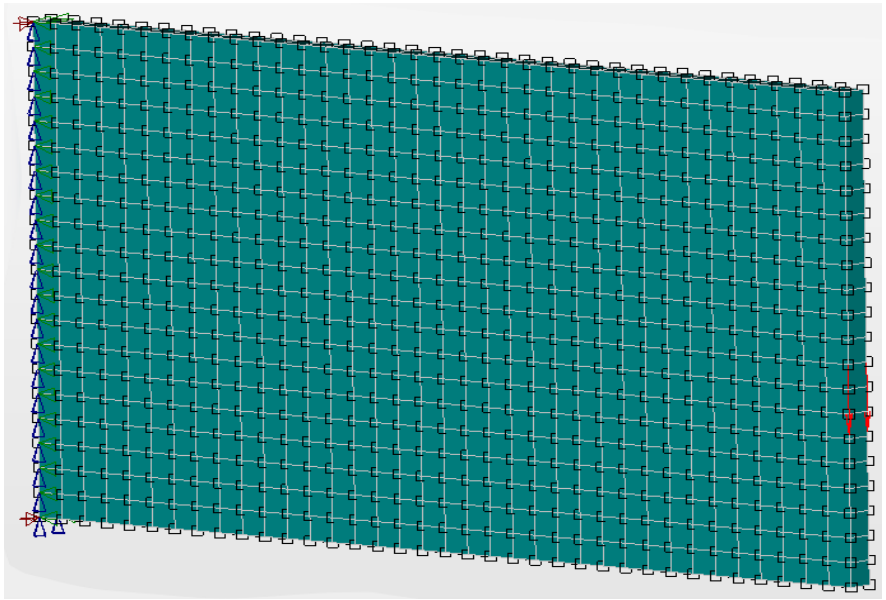
$$S_p(n) = T_1(n) / T_p(n)$$

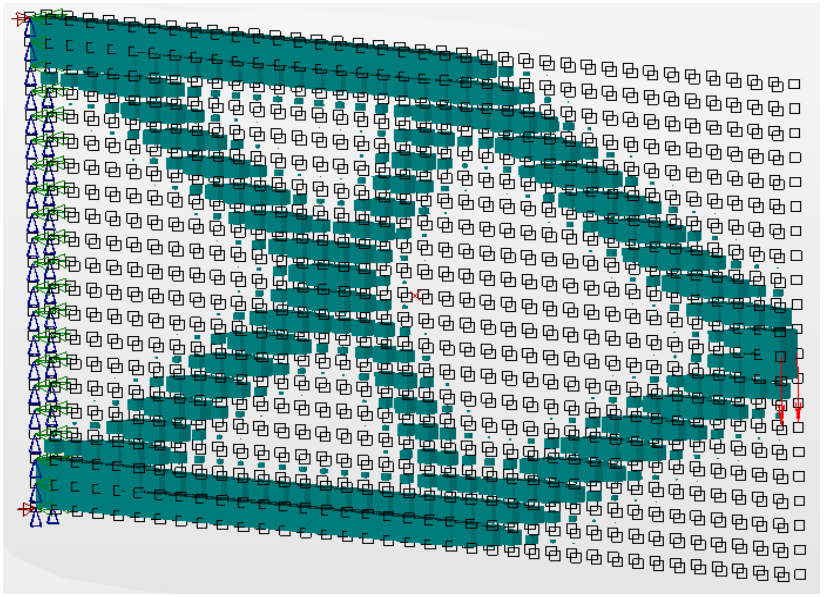
Размерность матрицы	**Количество ненулевых элементов матрицы**	**Ускорение,** $S_p(n)$
15246x15246	315768	7.29824
11616x11616	312058	6.56842

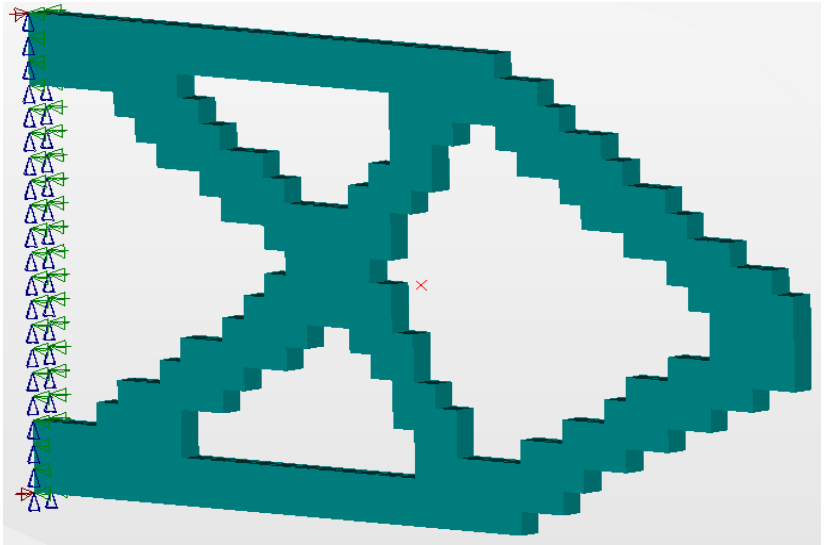
Бинаризация серых изображений при топологической оптимизации











Бинаризация серых изображений при топологической оптимизации

Метод Оцу

Пиксели классифицируются на 2 класса (“полезные” и “фоновые”). Метод Оцу ищет порог, уменьшающий дисперсию внутри класса, которая определяется как взвешенная сумма дисперсий двух классов:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{15}$$

Сначала вычисляется распределение гистограммы для изображения, а затем выполняется нормализация:

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)} \tag{16}$$

Бинаризация серых изображений при топологической оптимизации

Дисперсии для двух классов:

$$\sigma_1^2(t) = \sum_{i=1}^{t}[i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \sigma_2^2(t) = \sum_{i=t+1}^{I}[i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \tag{17}$$

Полная дисперсия может быть определена уравнением в виде суммы дисперсии внутри класса и дисперсии между классами:

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2(t) \tag{18}$$

Минимизация дисперсии внутри класса равносильна максимизации дисперсии между классами:

$$\sigma_b^2(t) = q_1(t)q_2(t)[\mu_1(t) - \mu_2(t)]^2 \tag{19}$$

Вычислительные эксперименты

$$S_p(n) = T_1(n) / T_p(n)$$

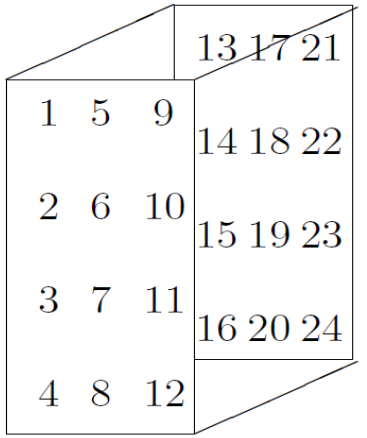
Количество элементов	**Ускорение,** $S_p(n)$
315768	8.75788
312058	7.88209
2066993	9.26103
23142191	9.4462506

Разложение тензорных поездов (Tensor-Train Decomposition)

$$A = \left(\begin{array}{ccc|ccc} 1 & 5 & 9 & 13 & 17 & 21 \\ 2 & 6 & 10 & 14 & 18 & 22 \\ 3 & 7 & 11 & 15 & 19 & 23 \\ 4 & 8 & 12 & 16 & 20 & 24 \end{array}\right)$$

$$A_{M\times N} \to A_{m\times n\times r\times \ldots \times z},$$
$$m\times n\times r\times \ldots \times z = M\times N$$

A =



TT-формат тензора [1-3]:

$$A(i_1, i_2, \ldots, i_d) \approx \sum_{\alpha_1,\ldots,\alpha_d} G_1(i_1, \alpha_1) G_2(\alpha_1, i_2, \alpha_2) \ldots G_{d-1}(\alpha_{d-2}, i_{d-1}, \alpha_{d-1}) G_d(\alpha_{d-1}, i_d) \quad (20)$$

$$\mathrm{A}(\mathrm{i}_1, \ldots, \mathrm{i}_\mathrm{d}) = \mathrm{G}_1(\mathrm{i}_1) \ldots \mathrm{G}_\mathrm{d}(\mathrm{i}_\mathrm{d}) \qquad G_k(i_k) = [G_k(\alpha_{k-1}, i_k, \alpha_k)] \quad (21)$$

1) L. V. Oseledets, “Tensor-Train Decomposition” , *SIAM J. Sci. Comput.* 2011, 33(5), 2295–2317

2) С.В. Долгов, “Алгоритмы и применения тензорных разложений для численного решения многомерных нестационарных задач”, диссертация на соискание ученой степени к.ф.-м.н, Москва 2014

3) Patrick Gelß, “The Tensor-Train Format and Its Applications”, dissertation zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.), Berlin 2017

Разложение тензорных поездов (Tensor-Train Decomposition)

Алгоритм TT-разложения :

1. Вычислить норму: $\|\mathbf{A}\|_F = \sqrt{\sum_{i_1,\ldots,i_d} A^2(i_1,\ldots,i_d)}$ $A \in \Re^{n_1 \times n_2 \times \ldots \times n_d}$

2. Вычислить размер первого ядра: $N_l = n_1, N_r = \Pi_{k=2}^{d} n_k$

3. M = A. Свернуть M в: $M = M \in \Re^{N_l \times N_r}$

4. Вычислить усеченное сингулярное разложение: $M \approx USV$ такое, что: $\sqrt{\sum_{k=r+1}^{\min(N_l,N_r)} \sigma_k^2} \le \frac{\varepsilon \cdot \|A\|_F}{\sqrt{d-1}}$

5. $G_1 = U, \quad M = SV^T, \quad r_1 = r$

6. Цикл от k = 2 до d - 1.

7. Вычислить размер k-ого ядра: $N_l = n_k, N_r = \frac{N_r}{n_k}$

8. $M = M \in \Re^{rN_l \times N_r}$

9. Вычислить усеченное сингулярное разложение: $M \approx USV$ такое, что: $\sqrt{\sum_{k=r+1}^{\min(N_l,N_r)} \sigma_k^2} \le \frac{\varepsilon \cdot \|A\|_F}{\sqrt{d-1}}$

10. $G_k = U \in \Re r_{k-1} \times n_k \times r_k$

11. $M = SV$

12. $G_d = M^T$

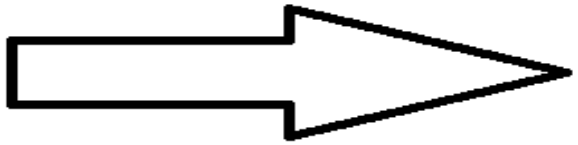
Разложение тензорных поездов (Tensor-Train Decomposition)

```
A = rand(10^3, 10^3);
error = 10^(-15);
A_Tensor = reshape(A, 100,1,100,1,100);
ttA = tt_decomposition(A_Tensor,error);
```

A ttA

1000x1000 double

	1	2	3	4
1	0.8464	0.1946	0.2679	0.1312
2	0.0514	0.9001	0.7362	0.9339
3	0.0227	0.5615	0.6041	0.8116
4	0.8694	0.6642	0.7733	0.7983
5	0.0871	0.2267	0.2976	0.9029
6	0.4870	0.4978	0.0228	0.5651
7	0.2535	0.0328	0.4343	0.7690
8	0.8939	0.7892	0.3128	0.6003
9	0.1354	0.0816	0.6948	0.4755



A ttA

5x1 cell

	1
1	*100x100 double*
2	*100x1 double*
3	*1x100x100 double*
4	*100x1 double*
5	*100x1 double*
6	

Матрица из миллиона элементов преобразуется в коллекцию из 20 тысяч элементов

Разложение тензорных поездов (Tensor-Train Decomposition)

Решение СЛАУ:

$$
\begin{aligned}
&Ax = b, \\
&x(i_1,\ldots,i_d) \approx \sum_{\alpha_1,\ldots,\alpha_{d-1}} x^{(1)}_{\alpha_1}(i_1)\,x^{(2)}_{\alpha_1\alpha_2}(i_2)\ldots x^{(d-1)}_{\alpha_{d-2}\alpha_{d-1}}(i_{d-1})\,x^{(d)}_{\alpha_{d-1}}(i_d), \\
&b(i_1,\ldots,i_d) \approx \sum_{\beta_1,\ldots,\beta_{d-1}} b^{(1)}_{\beta_1}(i_1)\,b^{(2)}_{\beta_1\beta_2}(i_2)\ldots b^{(d-1)}_{\beta_{d-2}\beta_{d-1}}(i_{d-1})\,b^{(d)}_{\beta_{d-1}}(i_d), \\
&A(i_1,\ldots,i_d,\,j_1,\ldots,j_d) \approx \sum_{\gamma_1,\ldots,\gamma_{d-1}} A^{(1)}_{\gamma_1}(i_1,j_1)\,A^{(2)}_{\gamma_1\gamma_2}(i_2,j_2)\ldots A^{(d)}_{\gamma_{d-1}}(i_d,j_d)
\end{aligned}
\tag{22}
$$

Решаем маленькие подзадачи:

$$A^{(k)}x^{(k)} = b^{(k)} \tag{23}$$

Спасибо за внимание!

Компания НТЦ «АПМ»
(научно-технический центр)

Московская область, г. Королев
Октябрьский бульвар, д. 14, офис 6

Тел.: (498) 600-25-10, (495) 514-84-19
Internet: www. apm.ru, www.cae.apm.ru

E-mail: com@apm.ru