

# Parallel Factorization Methods for Domain Decomposition

V. P. Il'in

Institute of Computational Mathematics  
and Mathematical Geophysics SBAS

17-th International Conference PACT - 2023

August, 21-25, 2023

Astana

# ЛИТЕРАТУРА

1. Y..L. Gurieva , V.P. Il'in , and D.I. Kozlov. Parallel Domain Decomposition Methods with Graph Preconditioning. Параллельные вычислительные технологии (ПаВТ'2023) Parallel computational technologies (PCT'2023) 215-228
2. Dolean, V., Jolivet, P., Nataf, F.: An introduction to domain decomposition methods: algorithms, theory and parallel implementation. SIAM, Philadelphia (2015).
3. Il'in, V.P.: Multi-Preconditioned Domain Decomposition Methods in the Krylov Subspaces. In: LNCS, 10187, Springer, pp. 95–106 (2017).
4. Il'in, V.P.: Iterative Preconditioned Methods in Krylov Spaces: Trends of the XXI Century. Comput. Math. Math. Phys., vol. 61, N. 11, pp. 1750–1775. Pleiades Publishing, Ltd (2021).
5. Ili'n, V. P., Kuznetsov, Yu.I: Tridiagonal Matrices and its Applications. Moscow, Nauka Publ. (1985). [in Russian].

# МЕТОДЫ ДЕКОМПОЗИЦИИ ОБЛАСТЕЙ

$$\Omega = \bigcup_{q=1}^P \Omega_q, \quad \bar{\Omega}_q = \Omega_q \cup \Gamma_q, \quad \Gamma_q = \bigcup_{q' \in \omega_q} \Gamma_{q,q'}, \quad \Omega_0 = R^d / \Omega,$$

$$\Gamma_{q,q'} = \Gamma_q \cap \bar{\Omega}_{q'}, \quad q' \neq q, \quad \Omega_0 = R^d / \Omega,$$

$$\omega_q = \{q_1, \dots, q_{M_q}\},$$

$$\bar{\Omega}_0 = \Omega_0 \cup \Gamma, \quad \Gamma_{q,0} = \Gamma_q \cap \bar{\Omega}_0 = \Gamma_q \cap \Gamma, \quad \Gamma_q = \Gamma_q^i \cup \Gamma_{q,0},$$

$$\Gamma_q^i = \bigcup_{q' \neq 0} \Gamma_{q,q'}, \quad \Gamma_{q,0} = \Gamma_q^e$$

# МЕТОД ШВАРЦА – ЯКОБИ

$$Lu_q(\vec{r}) = f_q, \quad \vec{r} \in \Omega_q, \quad l_{q,q'}(u_q) \Big|_{\Gamma_{q',q}} = g_{q,q'} \equiv l_{q',q}(u_{q'}) \Big|_{\Gamma_{q',q}},$$

$$q' \in \omega_q, \quad l_{q,0}u_q \Big|_{\Gamma_{q,0}} = g_{q,0}, \quad q = 1, \dots, P, \quad \alpha_q \cdot \beta_q \geq 0,$$

$$\alpha_q u_q + \beta_q \frac{\partial u_q}{\partial \vec{n}_q} \Big|_{\Gamma_{q,q'}} = \alpha_{q'} u_{q'} + \beta_{q'} \frac{\partial u_{q'}}{\partial \vec{n}_{q'}} \Big|_{\Gamma_{q',q}}, \quad |\alpha_q| + |\beta_q| > 0.$$

$$A_{q,q}u_q + \sum_{r \in \hat{\omega}_q} A_{q,r}u_r = f_q, \quad q = 1, \dots, P,$$

Стрелообразная блочная матрица

$$A = \{A_{q,r}\} = \left| \begin{array}{ccc|c} A_{1,1} & \dots & 0 & A_{1,P+1} \\ \dots & \dots & \dots & \dots \\ 0 & \dots & A_{P,P} & A_{P,P+1} \\ A_{P+1,1} & \dots & A_{P+1,P} & A_{P+1,P+1} \end{array} \right|$$

# КЛАССИЧЕСКИЙ ПРИМЕР ДЕКОМПОЗИЦИИ ОБЛАСТИ НА ДВЕ ПОДОБЛАСТИ

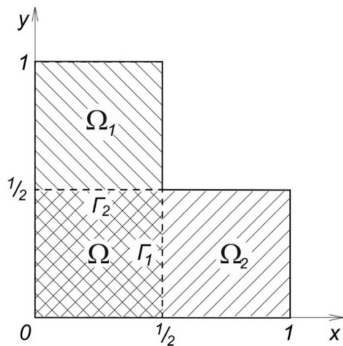


Figure: 1

# ПРИМЕР ДЕКОМПОЗИЦИИ НЕОГРАНИЧЕННОЙ ОБЛАСТИ ДЛЯ ВНЕШНЕЙ КРАЕВОЙ ЗАДАЧИ

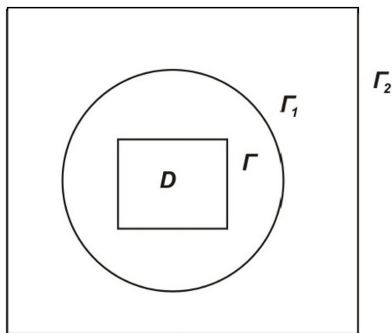


Figure: 2

# Примеры 1D-, 2D- и 3D-декомпозиции

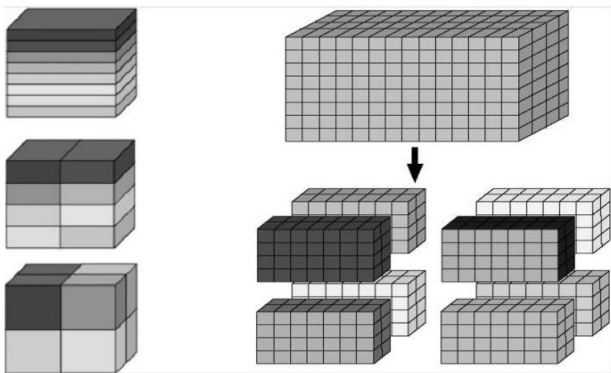


Figure: 3

## ПОСТАНОВКА ЗАДАЧИ: с.п.о. СЛАУ

$$Au = f, \quad A = \{a_{l,m}\} \in \mathbb{R}^{N,N}, \quad u = \{u_l\}, \quad f = \{f_l\} \in \mathbb{R}^N$$

Сеточные уравнения для 2D

$$\Omega = [x_0, x_{N_x+1}] \times [y_0, y_{N_y+1}], \quad \bar{\Omega} \in \Omega \cup \Gamma$$

$$(Au)_{i,j} = a_{i,j}^{(0)} u_{i,j} - a_{i,j}^{(1)} u_{i-1,j} - a_{i,j}^{(2)} u_{i,j-1} - a_{i,j}^{(3)} u_{i+1,j} - a_{i,j}^{(4)} u_{i,j+1} = f_{i,j},$$
$$i, j \in \Omega^h, \quad a_{1,j}^{(1)} = a_{i,1}^{(2)} = a_{N_x,j}^{(3)} = a_{i,N_y}^{(4)} = 0$$

# ДЕКОМПОЗИЦИЯ СЕТОЧНОЙ ОБЛАСТИ 2D

Макросеть  $\Omega^h$  :

$$\Omega^h : \quad x_{i_1}, \dots, x_{i_{M_x}}; \quad y_{j_1}, \dots, y_{j_{M_y}}; \quad 1 < i_1 < i_{M_x} \leq N_x; \quad 1 < j_1 < j_{M_y}$$

Подобласти:

$$\Omega_1^h, \dots, \Omega_M^h,$$

$$M = (M_x + 1)(M_y + 1),$$

$$M_v = M_x M_y$$

$$K = M_x + M_y + 2M_x M_y,$$

$$\hat{u} = \{\hat{u}_k, k = 1, \dots, K\},$$

$$\check{u} = \{\check{u}_m, m = 1, \dots, M_v\}$$

# ДЕКОМПОЗИЦИЯ СЕТОЧНОЙ ОБЛАСТИ 2D

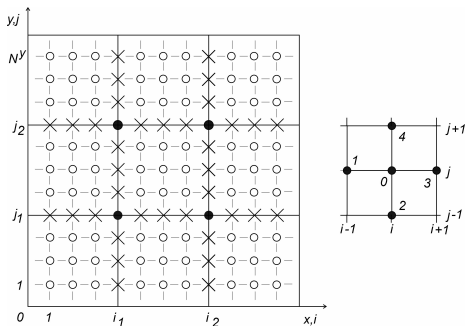


Figure: 4 Пример декомпозиции сеточной двумерной области

- – макроузлы,
- × – макрорёберные узлы,
- – макрограничные узлы.

# БЛОЧНАЯ СТРУКТУРА СЛАУ ДЛЯ 2D

$$\bar{u}_1 = \{\hat{u}_k, k = 1, \dots, K\}, \quad \text{и } \bar{u}_2 = \{\check{u}_m, m = 1, \dots, M_v\}$$

$$\bar{A}\bar{u} = \begin{bmatrix} \bar{A}_{1,1} & \bar{A}_{1,2} \\ \bar{A}_{2,1} & \bar{A}_{2,2} \end{bmatrix} \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} = \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix}.$$

$$\bar{A}_{2,2} = \text{block-diag}\{D_s, s = 1, \dots, M\}$$

$$\bar{N}_1 = M_x M_y + KN_e, \quad \bar{N}_2 = (M_x + 1)(M_y + 1)N_e^2 = MN_e^2.$$

# СТРУКТУРА МАТРИЦЫ ДЛЯ 2D-МАКРОГРАФА

$$\bar{A}_{1,1} = \begin{array}{|cccc|cccc|c}
\hline
T_1^x & \cdot & \cdot & \cdot & 0 & 0 & \cdot & \cdot & \cdot & 0 & C_1^x \\
\cdot & \cdot & & & \cdot & \cdot & & & & \cdot & \cdot \\
\cdot & & \cdot & & \cdot & \cdot & & & & \cdot & \cdot \\
\cdot & & & \cdot & \cdot & \cdot & & & & \cdot & \cdot \\
0 & \cdot & \cdot & \cdot & T_{M_x}^x & 0 & \cdot & \cdot & \cdot & 0 & C_{M_x}^x \\
\hline
0 & \cdot & \cdot & \cdot & 0 & T_1^y & \cdot & \cdot & \cdot & 0 & C_1^y \\
\cdot & & & & \cdot & \cdot & & & & \cdot & \cdot \\
\cdot & & & & \cdot & & \cdot & & & \cdot & \cdot \\
\cdot & & & & \cdot & & & \cdot & & \cdot & \cdot \\
0 & \cdot & \cdot & \cdot & 0 & 0 & \cdot & \cdot & \cdot & T_{M_y}^y & C_{M_y}^y \\
\hline
(C_1^x)^\top & \cdot & \cdot & \cdot & (C_{M_x}^x)^\top & (C_1^y)^\top & \cdot & \cdot & \cdot & (C_{M_y}^y)^\top & S \\
\hline
\end{array}$$

# МАТРИЧНЫЕ СТРУКТУРЫ ДЛЯ ДЕКОМПОЗИЦИИ ТРЕХМЕРНОЙ КРАЕВОЙ ЗАДАЧИ

Сеточная расчетная область

$$\Omega = [x_0, x_{N_x+1}] \times [y_0, y_{N_y+1}] \times [z_0, z_{N_z+1}]$$

$$\Omega^h : x = x_i, i = 1, \dots, N_x; y = y_j, j = 1, \dots, N_y; z = z_k, k = 1, \dots, N_z$$

Макросеть :

$$x = x_{i_1}, \dots, x_{i_{M_x}}; y = y_{j_1}, \dots, y_{j_{M_y}}; z = z_{k_1}, \dots, z_{k_{M_z}}$$

$$M = (M_x + 1)(M_y + 1)(M_z + 1), \quad M_v = M_x M_y M_z$$

# ТРЕХМЕРНЫЙ СЕТОЧНЫЙ 7-ТОЧЕЧНЫЙ ШАБЛОН

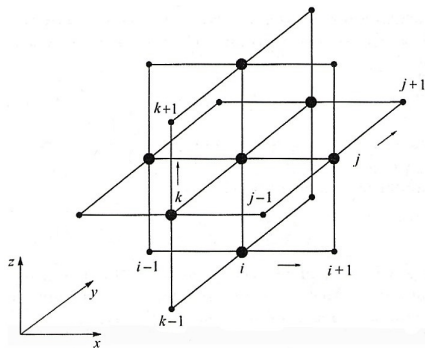


Figure: 5

# ФРАГМЕНТ ДЕКОМПОЗИЦИИ СЕТОЧНОЙ ТРЕХМЕРНОЙ ОБЛАСТИ

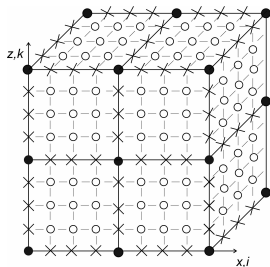


Figure: 6

- – макроузлы,
- × – макрорёберные узлы,
- – макрограневые узлы.
- ⊗ – внутренние узлы в подобластях

# ПРЕДОБУСЛАВЛИВАЮЩИЕ МАТРИЦЫ

$$B = \begin{bmatrix} G_1 & 0 \\ \bar{A}_{2,1} & G_2 \end{bmatrix} G^{-1} \begin{bmatrix} G_1 & \bar{A}_{1,2} \\ 0 & G_2 \end{bmatrix}$$

- методы неполной факторизации с компенсацией (CIF)

$$G_1 = \bar{A}_{1,1} - \theta_1 S_1, \quad G_2 = \bar{A}_{2,2} - \overline{\bar{A}_{2,1} G_1^{-1} \bar{A}_{1,2}} - \theta_2 S_2,$$

$$S_1 e = \bar{A}_{1,2} e, \quad S_2 e = \left( \bar{A}_{2,1} G_1^{-1} \bar{A}_{1,2} - \overline{\bar{A}_{2,1} G_1^{-1} \bar{A}_{1,2}} \right) e$$

- методы симметричной последовательной релаксации (SSOR)

$$G_k = \frac{1}{\omega} \text{diag} \bar{A}_k, \quad k = 1, 2, \quad \omega \in (0, 2)$$

## РЕШЕНИЕ СЛАУ $Bq^{n+1} = r^{n+1}$

$$Bq^{n+1} = \begin{bmatrix} \bar{G}_1 & O \\ \bar{A}_{2,1} & G_2 \end{bmatrix} \begin{bmatrix} \bar{v}_1^{n+1} \\ \bar{v}_2^{n+1} \end{bmatrix} = \begin{bmatrix} r_1^{n+1} \\ r_2^{n+1} \end{bmatrix},$$

$$\begin{bmatrix} I & \bar{G}_1^{-1} \bar{A}_{1,2} \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{q}_1^{n+1} \\ \bar{q}_2^{n+1} \end{bmatrix} = \begin{bmatrix} v_1^{n+1} \\ v_2^{n+1} \end{bmatrix}$$

$$\begin{aligned} G_1 v_1^n &= r_1^{n+1}, & G_2 v_2^n &= r_2^{n+1} - \bar{A}_{2,1} v_1^n, \\ q_2^{n+1} &= v_2^n, & G_1 q_1^{n+1} &= v_1^n - \bar{A}_{1,2} v_2^{n+1} \end{aligned}$$

# ПРЕДОБУСЛОВЛЕННЫЕ ИТЕРАЦИОННЫЕ МЕТОДЫ В ПОДПРОСТРАНСТВАХ КРЫЛОВА

$$\begin{aligned}\hat{p}^0 &= \hat{r}^0 = B^{-1}r^0 = B^{-1}(f - Au^0), \quad n = 1, 2, \dots; \\ u^n &= u^{n-1} + \alpha_{n-1}\hat{p}^{n-1}, \quad \hat{r}^n = \hat{r}^{n-1} - \alpha_{n-1}A\hat{p}^{n-1}; \\ \hat{p}^n &= \hat{r}^n + \beta_{n-1}\hat{p}^{n-1}, \quad \alpha_{n-1} = \sigma_{n-1}/\rho_{n-1}, \quad \beta_{n-1} = \sigma_n/\sigma_{n-1}; \\ \sigma_{n-1} &= (A^{\gamma-1}\hat{r}^{n-1}, \hat{r}^{n-1}), \quad \rho_{n-1} = (B^{-1}A\hat{p}^{n-1}, A^{\gamma-1}\hat{p}^{n-1}),\end{aligned}$$

Алгоритмы сопряженных направлений

$\gamma = 1$  – метод сопряженных градиентов (CG)

$\gamma = 2$  – метод сопряженных невязок (CR)

$\gamma = 0$  – максимальных ошибок (ME)

# ПАРАЛЛЕЛЬНЫЙ ПРЯМОЙ МЕТОД РЕШЕНИЯ СЛАУ НА МАКРОГРАФЕ, 1

Мы рассмотрим в единообразной форме алгоритм решения СЛАУ вида

$$A_{1,1}u = \begin{bmatrix} T & C \\ C^T & S \end{bmatrix} \begin{bmatrix} u_e \\ u_v \end{bmatrix} = \begin{bmatrix} f_e \\ f_v \end{bmatrix}, \quad u_e, f_e \in \mathbb{R}^{N_e}, \quad u_v, f_v \in \mathbb{R}^{N_v}, \quad (1)$$

где  $T = \text{block-diag}\{T_l, l = 1, \dots, N_e/n\}$  — блочно-диагональная матрица, в которой каждый блок  $T_e$  есть трёхдиагональная матрица порядка  $N_e/n$ .

Каждое из  $N_v$  уравнений для макровершин ( $N_v \ll N_e$ ) содержит по 4 связи с макрорёберными переменными в двумерном случае и по 6 — в трёхмерном.

Идея алгоритма заключается в предварительном исключении из системы (1) подвектора  $u_e$  с помощью экономичного метода прогонки.

## ПАРАЛЛЕЛЬНЫЙ ПРЯМОЙ МЕТОД РЕШЕНИЯ СЛАУ НА МАКРОГРАФЕ, 2

Трёхточечные уравнения на одном макрорёбре

$$(Tu)_t = -a_t u_{t-1} + b_t u_t - c_t u_{t+1} = f_t, \quad t = 1, 2, \dots, N_e, \quad (2)$$

где  $u_0, u_{N_e+1}$  — значения в примыкающих макровершинах, через которые будем выражать искомые  $u_t$  из (3). По принципу суперпозиции мы можем записать

$$u = \{u_t = u_0 \hat{u}_t + u_e \check{u}_t + \bar{u}_t\}, \quad u = \hat{u} + \check{u} + \bar{u}.$$

Общее решение трёхдиагональной системы может быть представлено с помощью одной из двух рекурсий

$$u_t = \begin{cases} \hat{\beta}_t u_{t+1} + \hat{z}_t, & t = N_{e-1}, \dots, 1; \quad u_{N_e} = \hat{z}_{N_e}, \\ \check{\beta}_t u_{t-1} + \check{z}_t, & t = 2, \dots, N_e; \quad u_1 = \check{z}_1. \end{cases} \quad (3)$$

## ДОПОЛНИТЕЛЬНОЕ УСКОРЕНИЕ В 2 РАЗА

$$\begin{aligned}\hat{\beta}_1 &= c_1 \hat{d}_1, \hat{d}_1 = b_1^{-1}, \hat{\beta}_t = c_t \hat{d}_t, \hat{d}_t = (b_t - a_t \hat{\beta}_{t-1})^{-1}, \quad t = 2, \dots, N_e, \\ \check{\beta}_{N_e} &= a_{N_e} \hat{d}_{N_e}, \hat{d}_{N_e} = b_{N_e}^{-1}, \check{\beta}_t = a_t \check{d}_t, \check{d}_t = (b_t - c_t \check{\beta}_{t+1})^{-1}, \quad t = N_e, \dots, 1, \\ \hat{z}_1 &= f_1 \hat{d}_1, \quad \hat{z}_t = (f_t - a_t \hat{z}_{t-1}) \hat{d}_t, \quad t = 2, \dots, N_e, \\ \check{z}_{N_e} &= f_{N_e} \check{d}_{N_e}, \quad \check{z}_t = (f_t - c_t \check{z}_{t+1}) \check{d}_t, \quad t = N_e - 1, \dots, 1.\end{aligned}\quad (4)$$

Пусть  $t_0 = N_e/2$ . Вычислим величины  $\hat{\beta}_t, \hat{d}_t, \hat{z}_t$ , для  $t = 1, \dots, i_0 - 1$ , и значения  $\check{\beta}_t, \check{d}_t, \check{z}_t$  — при  $t = N_e, \dots, i_0 + 1$ . После этого подставим решение в форме (3) в уравнение (2), в результате чего получаем формулу

$$\begin{aligned}u_{t_0} &= (f_{t_0} + a_{t_0-1} \hat{z}_{t_0-1} + c_{t_0+1} \check{z}_{t_0+1}) / \gamma_{t_0}, \\ \gamma_{t_0} &= t_0 b_{t_0} - a_{t_0} \hat{\beta}_{t_0-1} - c_{t_0} \check{\beta}_{t_0+1}.\end{aligned}\quad (5)$$

После нахождения  $u_{t_0}$  остальные компоненты решения рассчитываются для  $t < t_0$  и  $t > t_0$  синхронно с помощью рекурсий (3).

# ДАЛЬНЕЙШЕЕ ПОВЫШЕНИЕ МАСШТАБИРУЕМОСТИ РАСПАРАЛЛЕЛИВАНИЯ

Проведём экономичное вычисление столбцов матрицы  $G = \{g_{t,t'}\}$ , обратной к трёхдиагональной. Такой  $t'$ -й столбец можно определить через частное решение  $\bar{u}_t$ , при правой части вида  $f_t = \delta_{t',t''}$ , где  $\delta_{t',t''}$  – символ Кронекера. Диагональный элемент обратной матрицы:

$$g_{t,t'} = \gamma_{t'}^{-1}, \quad g_{t,t'} = \begin{cases} \hat{\beta}_{t''} \gamma_{t'}, & t'' = t' - 1, \dots, 1, \\ \check{\beta}_{t''} \gamma_{t'}, & t'' = t' + 1, \dots, N_e. \end{cases}$$

Отсюда для решения  $\bar{u}$  имеем достаточно ресурсоёмкую формулу

$$\bar{u} = Gf = \left\{ \bar{u}_t = \sum_{t'=1}^{N_e} g_{t,t'} f_t \right\},$$

которая, однако, при наличии  $N_e^2$  процессов реализуется на общей памяти всего за время  $\tau_a \ln N_e$ , где  $\tau_a$  — средняя длительность выполнения одной арифметической операции.

# DIRECT METHODS IN 2D, 1

LU-decomposition

$$A = \begin{bmatrix} A_{1,1} & 0 \\ A_{2,1} & S \end{bmatrix} \begin{bmatrix} A_{1,1}^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & S \end{bmatrix},$$

$$S = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}.$$

Exact factorization:

$$Au = \begin{bmatrix} A_{1,1} & 0 \\ A_{2,1} & S \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad \begin{bmatrix} I & A_{1,1}^{-1}A_{1,2} \\ 0 & I \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

$$A_{1,1}v_1 = f_1, \quad Sv_2 = f_2 - A_{2,1}v_1 \equiv g_2$$

$$u_2 = v_2, \quad A_{1,1}w_1 = A_{1,2}u_2, \quad u_1 = v_1 - w_1.$$

## DIRECT METHODS IN 2D, 2

$$A_{1,1} = \begin{array}{c|ccc} C & C_1 & \dots & C_{M_e} \\ \hline C_1^\top & T_1 & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ C_{M_e}^\top & 0 & 0 & T_{M_e} \end{array},$$

$$A_{1,1}u_1 = \begin{bmatrix} C & C_{1,2} \\ C_{2,1} & T \end{bmatrix} \begin{bmatrix} u_c \\ u_T \end{bmatrix} = \begin{bmatrix} f_c \\ f_T \end{bmatrix}$$

$$A_0 u_c \equiv (C - C_{1,2} T^{-1} C_{2,1}) u_c = f_c - C_{1,2} T^{-1} f_T, \quad u_1 = T^{-1} (f_T - C_{2,1} u_c)$$

$$A_{1,1} G_{l'l'} = \delta_{l'l'},$$

$$S^{(m)} v^{(m)} = \begin{bmatrix} S_{1,1}^{(m)} & S_{1,2}^{(m)} \\ S_{2,1}^{(m)} & S_{2,2}^{(m)} \end{bmatrix} \begin{bmatrix} v_1^{(m)} \\ v_2^{(m)} \end{bmatrix} = \begin{bmatrix} g_1^{(m)} \\ g_2^{(m)} \end{bmatrix}, \quad m = 1, \dots, M_\alpha.$$

## DIRECT METHODS FOR 3D

$$A = \begin{bmatrix} A_{1,1} & 0 & 0 \\ A_{2,1} & G_2 & 0 \\ 0 & A_{3,2} & G_3 \end{bmatrix} \begin{bmatrix} A_{1,1}^{-1} & 0 & 0 \\ 0 & G_2^{-1} & 0 \\ 0 & 0 & G_3^{-1} \end{bmatrix} \begin{bmatrix} A_{1,1} & A_{1,2} & 0 \\ 0 & G_2 & A_{2,3} \\ 0 & 0 & G_3 \end{bmatrix},$$

$$Au = \begin{bmatrix} A_{1,1} & 0 & 0 \\ A_{2,1} & G_2 & 0 \\ 0 & A_{3,2} & G_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix},$$

$$\begin{bmatrix} I & A_{1,1}^{-1}A_{2,1} & 0 \\ 0 & I & G_2^{-1}A_{2,3} \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

$$G_2 = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}, \quad G_3 = A_{3,3} - A_{3,1}G_2^{-1}A_{2,3}.$$

Block forward and back recursions

$$A_{1,1}v_1 = f, \quad G_2v_2 = f_2 - A_{2,1}v_1, \quad G_3v_3 = f_3 - A_{3,2}v_2,$$

$$u_3 = v_3, \quad u_2 = v_2 - G_2^{-1}u_3, \quad u_1 = v_1 - A_{1,1}^{-1}A_{1,2}v_2.$$

СПАСИБО ЗА ВНИМАНИЕ!

