

Теория решения уравнения теплопроводности на декартовых сетках.

Аппроксимация уравнения теплопроводности на декартовых прямоугольных сетках.

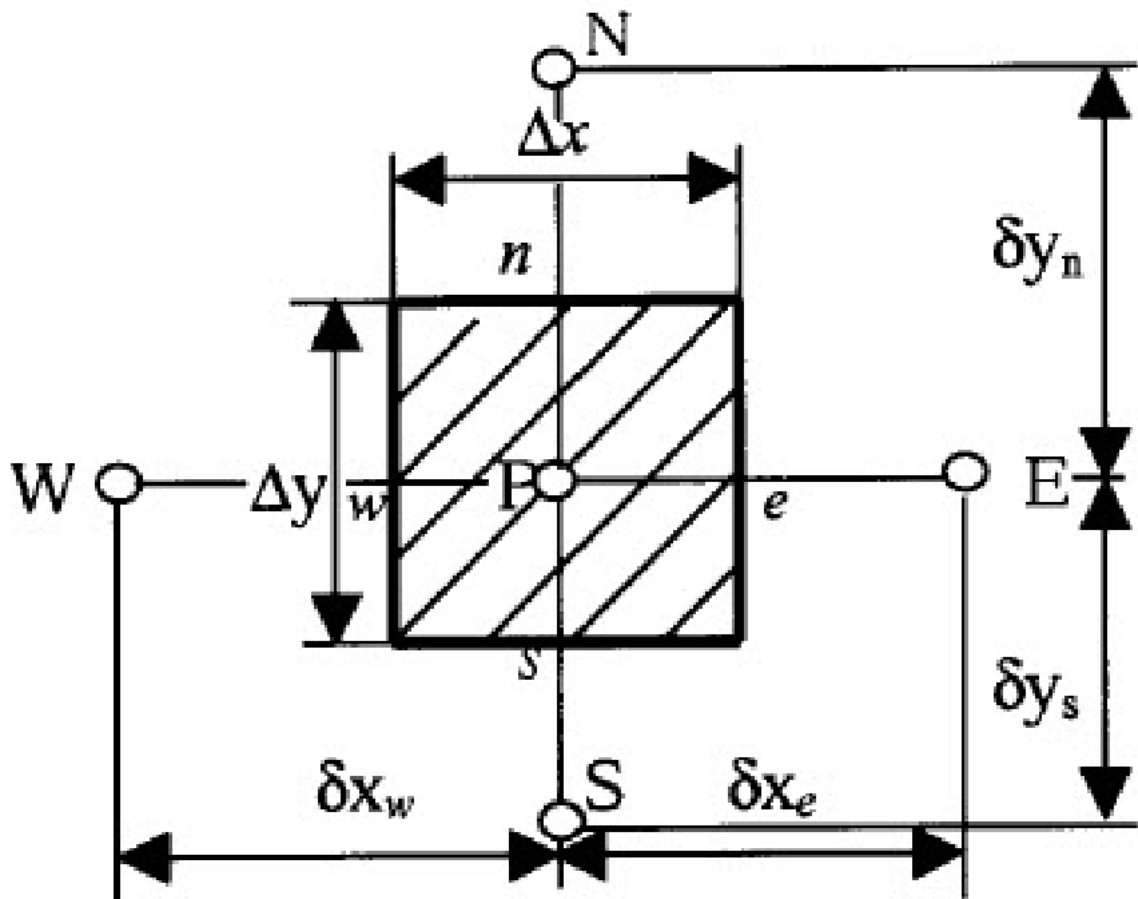


Рисунок 1 — Ячейка дискретизации.

Уравнение теплопроводности в стационарной постановке в 2D

$$\frac{\partial}{\partial x} \left(\lambda(x, y, T) \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda(x, y, T) \frac{\partial T}{\partial y} \right) = S_c + S_p \cdot T$$

В 3D

$$\frac{\partial}{\partial x} \left(\lambda(x, y, z, T) \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda(x, y, z, T) \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda(x, y, z, T) \frac{\partial T}{\partial z} \right) = S_c + S_p \cdot T$$

Где T – температура $^{\circ}\text{C}$, λ – коэффициент теплопроводности $\text{Вт}/(\text{м}\cdot\text{К})$, x, y, z – декартовы прямоугольные координаты, S – источники тепла в объеме моделируемого образца – S_c , $S_p < 0$ – линеаризация источников члена, в предположении его зависимости от температуры.

Для аппроксимации уравнения теплопроводности на структурированной сетке используется шаблон типа «крест»

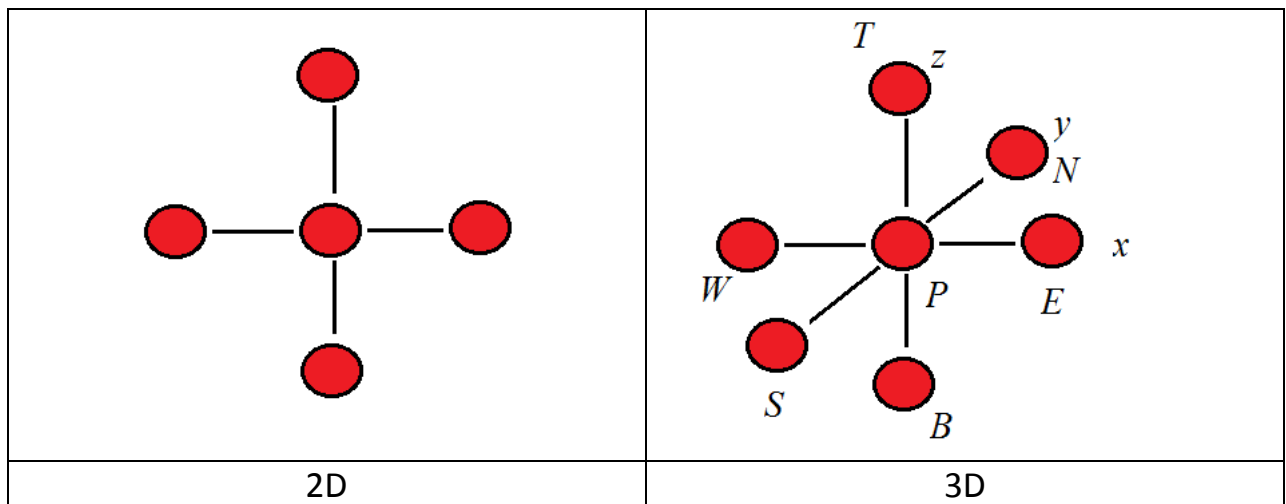


Рисунок 2 — Шаблон дискретизации уравнения теплопроводности на структурированной декартовой сетке.

На структурированной сетке в двумерном случае матрица имеет 5 диагоналей, а в трехмерном 7 диагоналей.

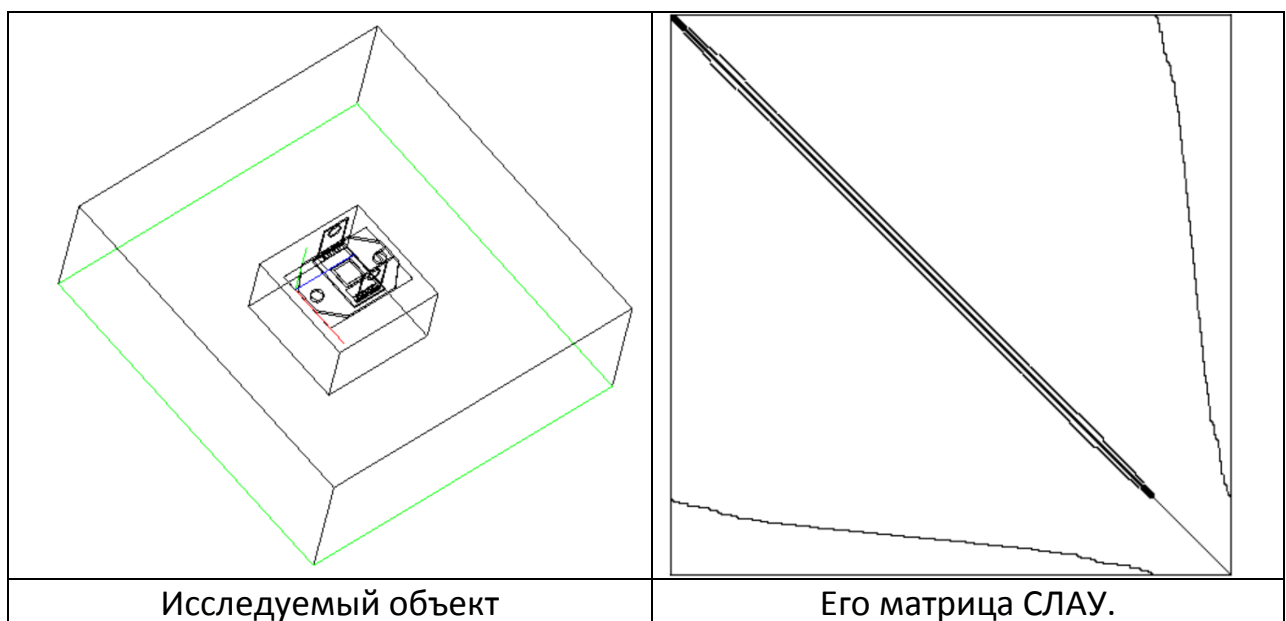


Рисунок 3 — Портрет матрицы СЛАУ для уравнения теплопередачи в трехмерном пространстве на структурированной сетке.

Сборка матрицы на Адаптивной Локально Измельчённой Сетке (АЛИС) в 2D случае.

Составление дискретного аналога для уравнения теплопередачи [7],[8]

$$\frac{\partial(\rho\Phi)}{\partial t} = \nabla(\Gamma\nabla\Phi) + S$$

на АЛИС сетке (для 2D).

Обозначения:

$\tilde{\Omega}_p$ - рассматриваемый контрольный объём,

ρ - плотность (в обобщённом смысле коэффициент перед нестационарным членом),

t - время,

Γ - коэффициент диффузии (в обобщённом смысле коэффициент диффузионного члена),

$S = S_C + S_p \cdot \Phi_p$ - линеаризованный источниковый член, $S_p < 0$.

Φ - обобщённая переменная, относительно которой записано уравнение.

$\nabla = \vec{i} \frac{\partial}{\partial x} + \vec{j} \frac{\partial}{\partial y}$ векторный символический оператор Гамильтона.

$\Omega_{ij} = (x_i, x_{i+1}) \times (y_j, y_{j+1})$ - ячейка дискретизации,

$h_x \times h_y$ - объём ячейки дискретизации,

\hat{A} - локальная матрица СЛАУ, \hat{b} - локальный вектор правой части.

1. 2D(для сетки АЛИС – адаптивной локально измельчённой сетки).

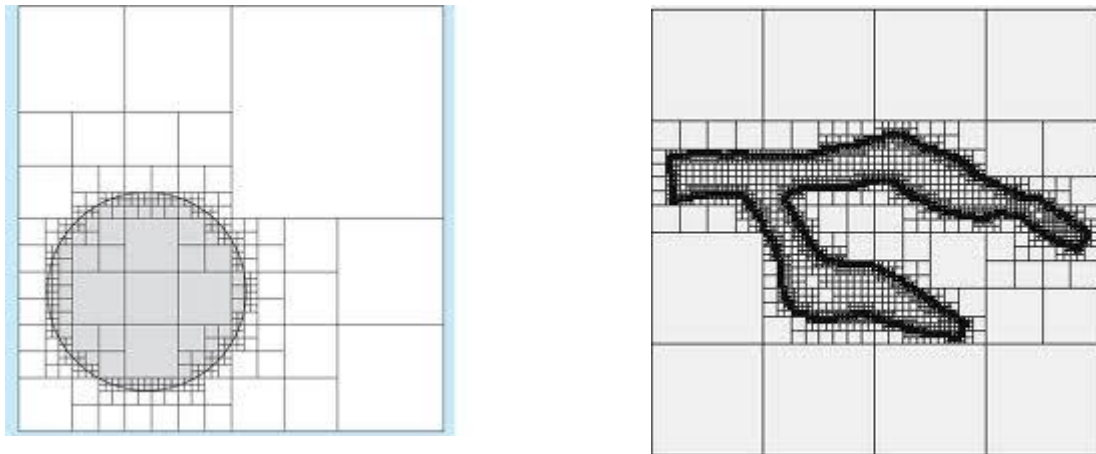


Рисунок 4 — Адаптивно Локально Измельчённая Сетка (АЛИС сетка).

Замечание 1: для того чтобы использовать АЛИС нужно применить поячеечную сборку матрицы СЛАУ. Поэтому сначала будет рассмотрена поузловая сборка на примере простого уравнения, а затем на основе поузловой сборки будет расписана поячеечная сборка матрицы.

1.1. Поузловая сборка матрицы – обрабатывается один узел сетки или контрольный объём $\tilde{\Omega}_p$.

Для внутреннего узла P (контрольного объёма окружающего узел P) формируется одна строка матрицы СЛАУ A .

Замечание 2: дискретизация по времени полностью неявная.

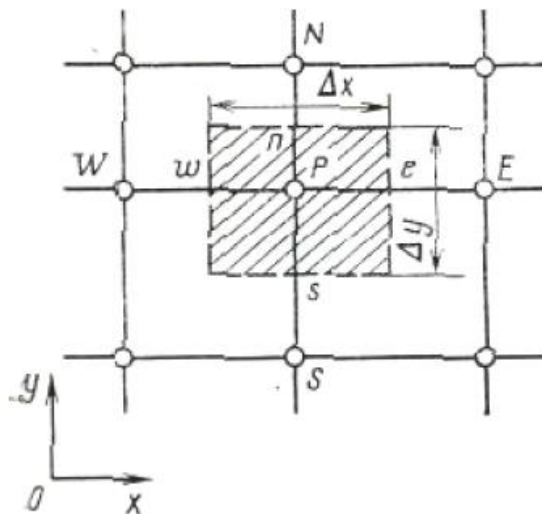


Рисунок 5 — Контрольный объём вокруг ячейки P .

$$(\delta x)_e = x_E - x_P, (\delta x)_w = x_P - x_W.$$

$$(\delta y)_n = y_N - y_P, (\delta y)_s = y_P - y_S.$$

$$\rho \frac{\partial \Phi}{\partial t} = \nabla(\Gamma \nabla \Phi) + S \text{ рассматриваемое уравнение [7]}$$

$$a_P \Phi_P = a_E \Phi_E + a_W \Phi_W + a_N \Phi_N + a_S \Phi_S + b \text{ - пяти узловой шаблон.}$$

$$a_E = \frac{\Gamma_e \Delta y}{(\delta x)_e}, a_W = \frac{\Gamma_w \Delta y}{(\delta x)_w}, a_N = \frac{\Gamma_n \Delta x}{(\delta y)_n}, a_S = \frac{\Gamma_s \Delta x}{(\delta y)_s}.$$

$$a_P^0 = \frac{\rho \Delta x \Delta y}{\Delta t}; b = S_C \Delta x \Delta y + a_P^0 \Phi_P^0.$$

$$a_P = a_E + a_W + a_N + a_S + a_P^0 - S_P \Delta x \Delta y \text{ - диагональный коэффициент матрицы СЛАУ.}$$

После обработки всех узлов, включая граничные (о которых будет рассказано отдельно), получается СЛАУ:

$$A\Phi = b.$$

которая, в данном случае, заполнялась построчно.

1.2. Поярочная сборка матрицы – обрабатывается одна ячейка сетки. Рассматривается вклад в четыре четвертинки четырёх различных контрольных объёмов (в 2D).

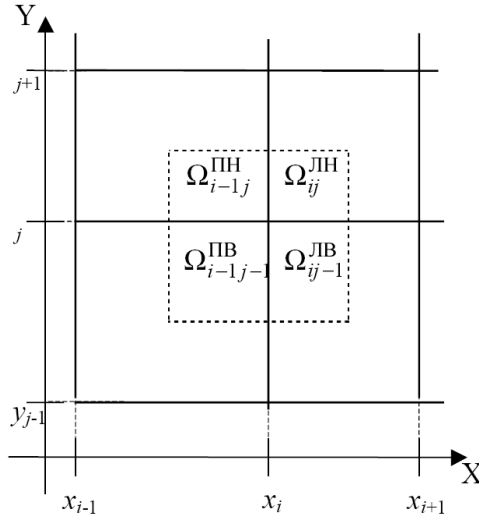


Рисунок 6 — Вид построенного вокруг внутреннего узла дискретизации (x_i, y_j) контрольного объёма $\tilde{\Omega}_p = \tilde{\Omega}_{ij}$, состоящего из четвертинок четырёх ячеек дискретизации. $\Omega_{i-1,j-1}, \Omega_{ij-1}, \Omega_{i-1,j}, \Omega_{ij}$. Здесь $p = i + (j-1) \cdot n_x$, где n_x количество узлов сетки по оси x .

Рассмотрим ячейку дискретизации $\Omega_{ij} = (x_i, x_{i+1}) \times (y_j, y_{j+1})$ [8]. Пусть $\delta x = h_x = x_{i+1} - x_i$, $\delta y = h_y = y_{j+1} - y_j$.

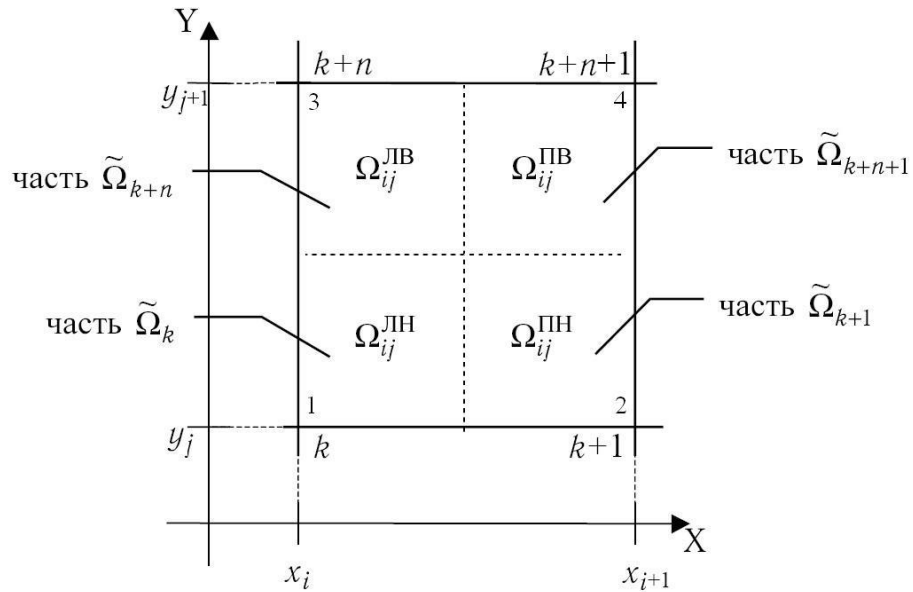


Рисунок 7 — Вид ячейки дискретизации Ω_{ij} с глобальной и локальной нумерацией её вершин.

Замечание 3 : теперь свойства материалов ρ, Γ полностью определяются ячейкой дискретизации и постоянны в её пределах.

Замечание 4:источниковый член определяется, напротив, в узлах расчётной сетки $\hat{S}_{Ci}, \hat{S}_{Pi}$, где $i=1..4$.

Локальная матрица и локальный вектор правой части:

$$\hat{A} = \begin{pmatrix} \hat{a}_{11} & \hat{a}_{12} & \hat{a}_{13} & \hat{a}_{14} \\ \hat{a}_{21} & \hat{a}_{22} & \hat{a}_{23} & \hat{a}_{24} \\ \hat{a}_{31} & \hat{a}_{32} & \hat{a}_{33} & \hat{a}_{34} \\ \hat{a}_{41} & \hat{a}_{42} & \hat{a}_{43} & \hat{a}_{44} \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{b}_4 \end{pmatrix}.$$

$$\hat{a}_{11} = 0.5\Gamma \cdot \left(\frac{h_y}{h_x} + \frac{h_x}{h_y} \right) - \hat{S}_{P1} \frac{h_x h_y}{4} + \frac{\rho h_x h_y}{4\Delta t},$$

$$\hat{a}_{12} = -0.5\Gamma \frac{h_y}{h_x}, \quad \hat{a}_{13} = -0.5\Gamma \frac{h_x}{h_y}, \quad \hat{a}_{14} = 0;$$

$$\hat{a}_{21} = -0.5\Gamma \frac{h_y}{h_x}, \quad \hat{a}_{22} = 0.5\Gamma \cdot \left(\frac{h_y}{h_x} + \frac{h_x}{h_y} \right) - \hat{S}_{P2} \frac{h_x h_y}{4} + \frac{\rho h_x h_y}{4\Delta t},$$

$$\hat{a}_{23} = 0, \quad \hat{a}_{24} = -0.5\Gamma \frac{h_x}{h_y};$$

$$\hat{a}_{31} = -0.5\Gamma \frac{h_x}{h_y}, \quad \hat{a}_{32} = 0,$$

$$\hat{a}_{33} = 0.5\Gamma \cdot \left(\frac{h_y}{h_x} + \frac{h_x}{h_y} \right) - \hat{S}_{P3} \frac{h_x h_y}{4} + \frac{\rho h_x h_y}{4\Delta t}, \quad \hat{a}_{34} = -0.5\Gamma \frac{h_y}{h_x};$$

$$\hat{a}_{41} = 0, \quad \hat{a}_{42} = -0.5\Gamma \frac{h_x}{h_y}, \quad \hat{a}_{43} = -0.5\Gamma \frac{h_y}{h_x},$$

$$\hat{a}_{44} = 0.5\Gamma \cdot \left(\frac{h_y}{h_x} + \frac{h_x}{h_y} \right) - \hat{S}_{P4} \frac{h_x h_y}{4} + \frac{\rho h_x h_y}{4\Delta t}.$$

$$\hat{b}_i = \hat{S}_{C_i} \cdot \frac{h_x h_y}{4} + \frac{\rho h_x h_y}{4\Delta t} \hat{\Phi}_i^0, \text{ где } i=1..4.$$

Локальные матрицы добавляются в глобальную, и затем производится учёт краевых условий. В результате решения глобальной СЛАУ искомая величина Φ будет найдена в каждом узле сетки.

Построение АЛИС сетки для геометрии пользователя [5],[9].

Для разрешения малых деталей геометрии расчетной области и высоких градиентов рассчитываемых величин используется прямоугольная адаптивная локально измельченная сетка (АЛИС).

Сущность технологии АЛИС заключается в следующем. Во всей расчетной области вводится прямоугольная сетка. Выделяются подобласти с особенностями геометрии или течения, в которых необходимо провести расчет на более мелкой сетке по сравнению с сеткой на нынешнем уровне разбиения. При этом расчетная ячейка, в которую попала выделяемая особенность, делится на 8 равных ячеек. Далее при необходимости, ячейки

делятся еще раз и так до достижения необходимой точности. Ячейки начальной сетки называются ячейками уровня 0, ячейки, получаемые измельчением уровня 0, называются ячейками уровня 1 и т.д. При генерации АЛИС накладывается условие, что гранями и ребрами могут граничить друг с другом только ячейки с номерами уровней, отличающимися не более чем на единицу.

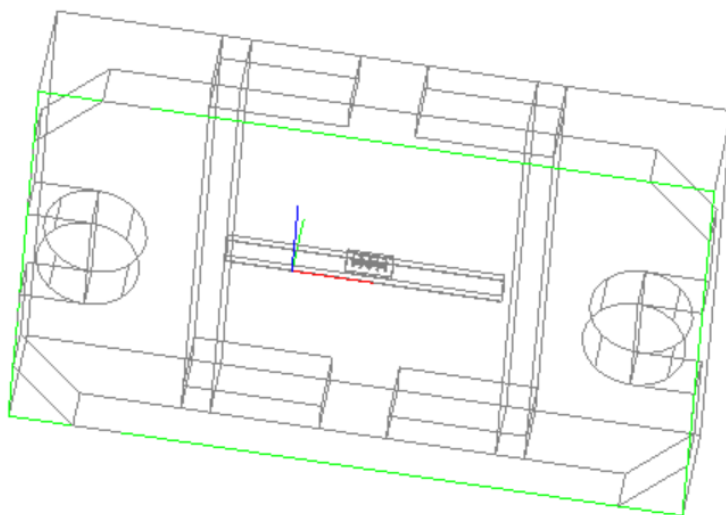


Рисунок 8 — ПТБШ с Шз=5мм в корпусе.

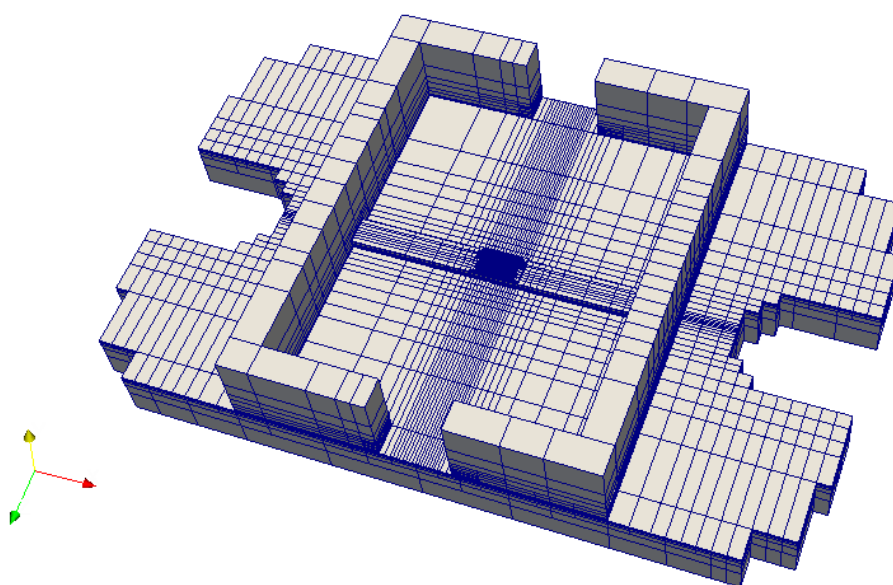


Рисунок 9 — АЛИС сетка для ПТБШ с Шз=5мм в корпусе. 40К ячеек сетки.

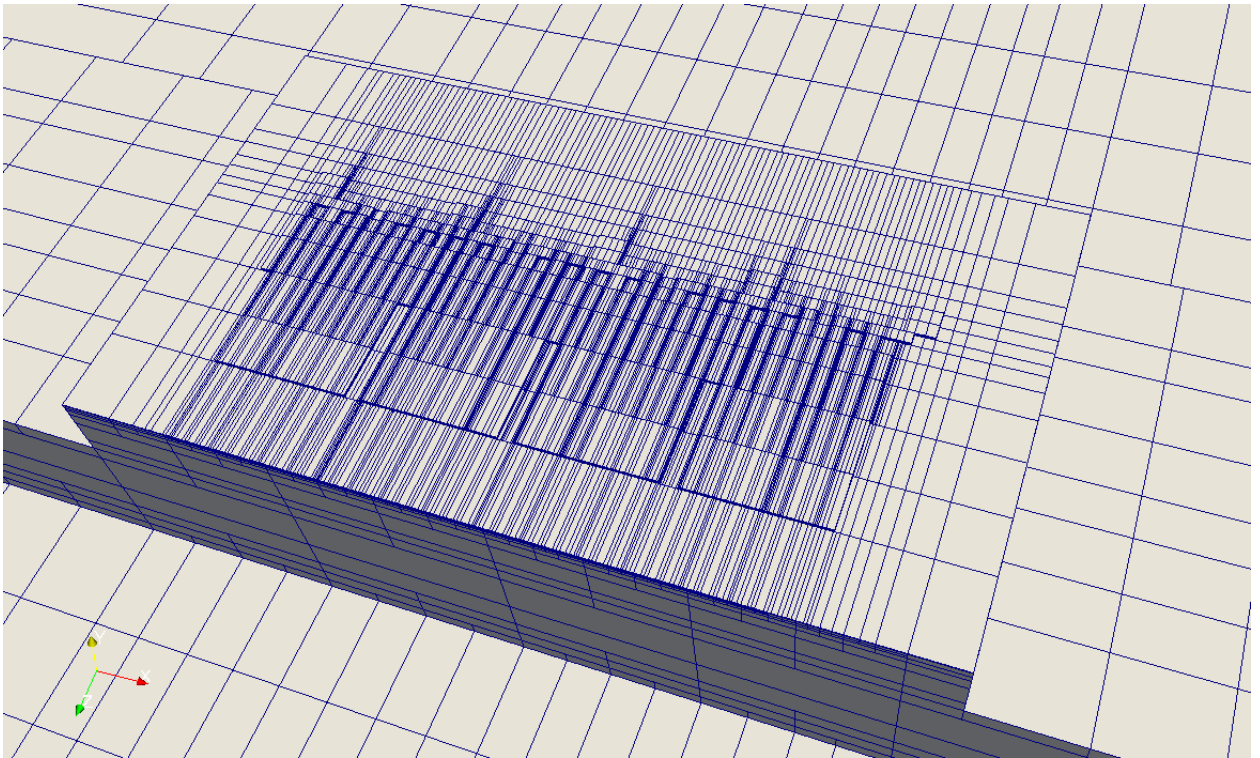


Рисунок 10 —АЛИС сетка для кристалла ПТБШ с $Шз=5\text{мм}$ в корпусе. 40К ячеек сетки.

АЛИС предназначена для описания разномасштабной геометрии от размеров затвора 0.5мкм до размеров корпуса 13мм и даже до размера целых модулей $\approx 100\text{мм}$.

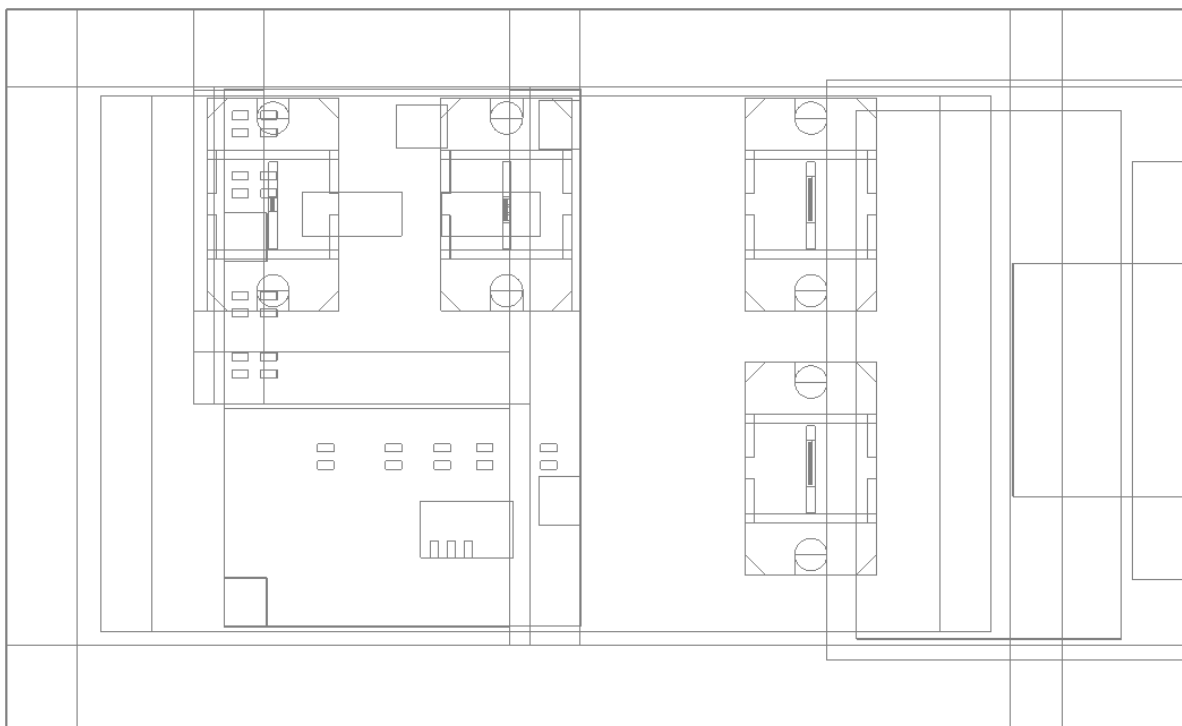


Рисунок 11 — Внешний вид модели ВУМ с четырьмя ПТБШ совместно.

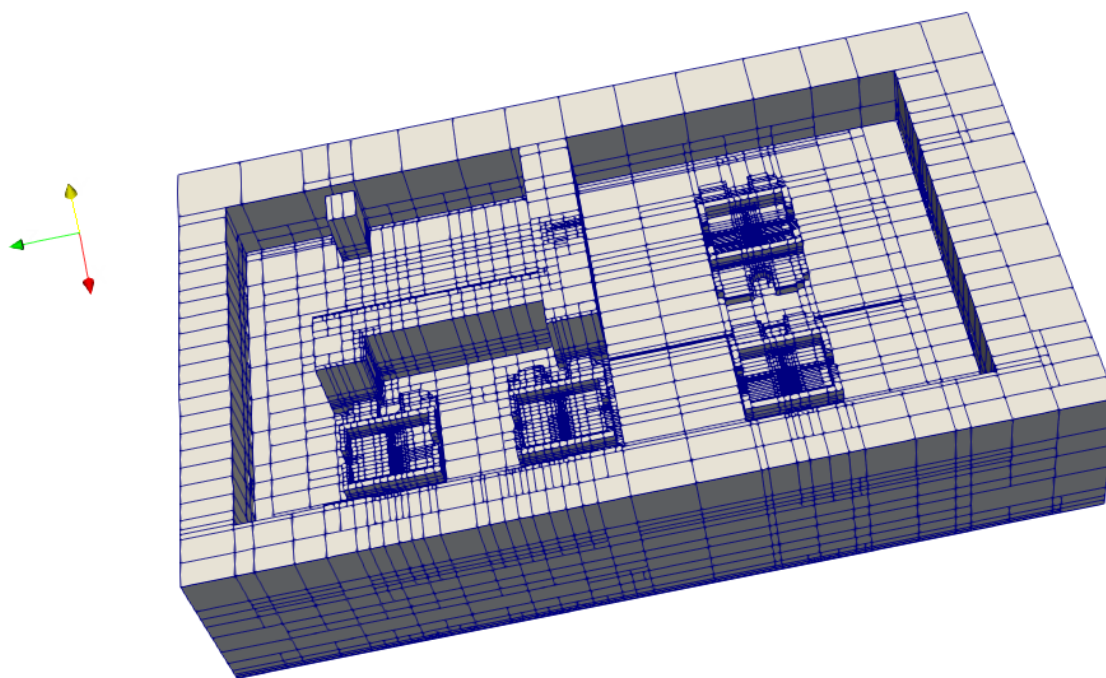


Рисунок 12 — Внешний вид расчётной сетки АЛИС для модели ВУМ с четырьмя ПТБШ совместно.

Алгоритм построения Адаптивной Локально Измельчённой Сетки (АЛИС).

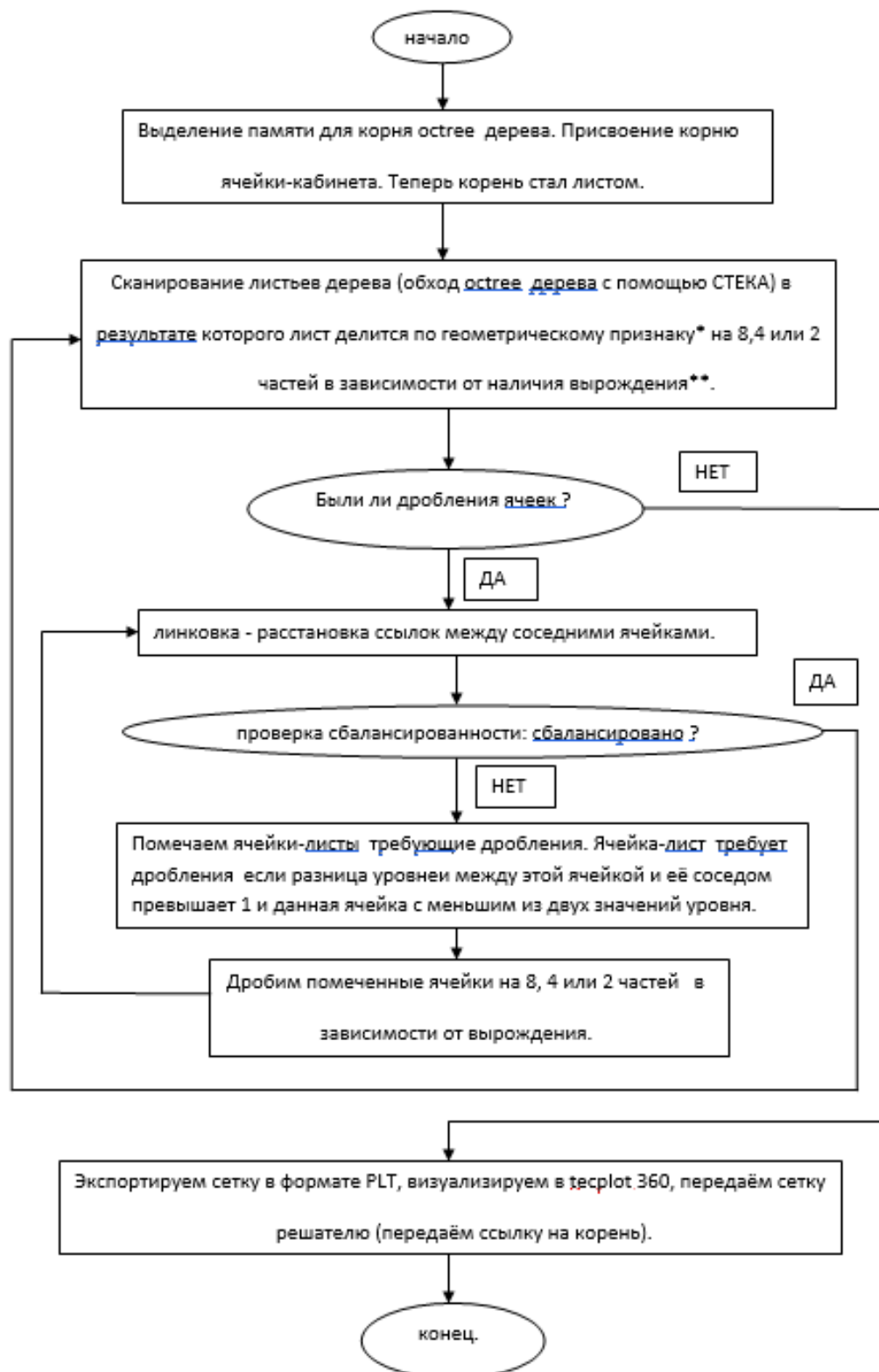


Рисунок 13 – Блок схема алгоритма построения АЛИС сеток.

*Дробление по геометрическому признаку:	**Ситуация вырождения:
<ol style="list-style-type: none"> 1. Построение первоначальной подробной структурированной HEX сетки (x,y, z). 2. Внутри текущей ячейки просмотр каждого координатного направления с помощью метода сканирующих отрезков. 3. При обнаружении пересечения сканирующего отрезка с пользовательской геометрией производится деление ячейки. 	<p>При достижении уровня первоначальной сетки возможны случаи:</p> <ol style="list-style-type: none"> 1. Нет вырождения. 2. Запрет дробления по X. 3. Запрет дробления по Y. 4. Запрет дробления по Z. 5. Запрет дробления по Y и Z. 6. Запрет дробления по X и Z. 7. Запрет дробления по X и Y. 8. Полный запрет дробления.

Метод решения СЛАУ[6].

На решение СЛАУ тратится более 95% времени работы алгоритма решения уравнения теплопередачи (за исключением времени затрачиваемого на запись файла с результатом на жесткий диск).

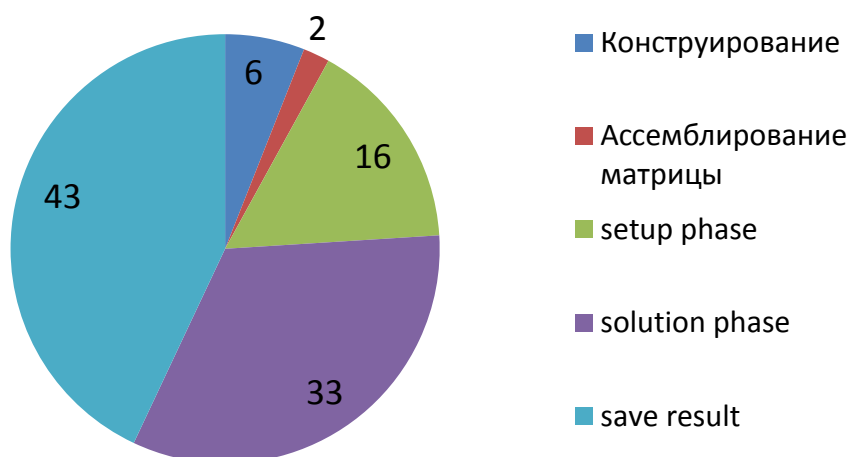


Рисунок 14 — Диаграмма распределения продолжительности операций в процентах % при решении типовой задачи теплопередачи с помощью алгебраического многосеточного метода.

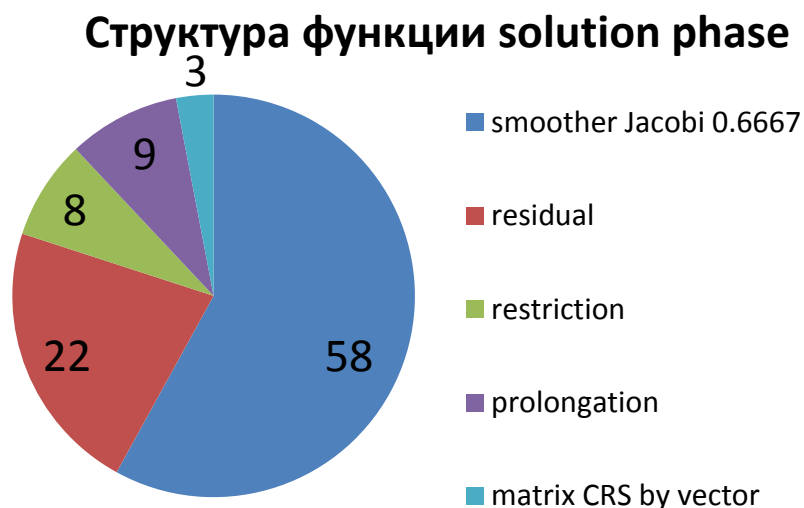


Рисунок 15 — Диаграмма распределения продолжительности операций функции *solution phase* в процентах % при решении типовой задачи теплопередачи с помощью алгебраического многосеточного метода. Полное время выполнения функции *solution phase* принято за 100 %.

Скорость сходимости многих методов обращения матриц может быть значительно повышена за счет использования методики под названием «многосеточный». Многосеточный процесс включает в себя проведение ранних итераций на мелкой сетке и более поздних итерации на вложенных более грубых «виртуальных» сетках. Результаты затем передаются обратно из грубых сеток на первоначальные более подробные мелкие сетки. С числовой точки зрения, многосеточный подход предлагает значительное преимущество. Для заданной сетки конкретного конечного размера, итерационные методы являются эффективными только при снижении ошибок, которые имеют длину волны порядка шага сетки (ребро тетраэдра при тетра мешировании, ребро гексаэдра при гекса доминант мешировании). Таким образом, в то время как более короткие длины волны ошибки исчезают довольно быстро, ошибки с большей длиной волны, порядка размера расчётной области, исчезают очень медленно (катастрофически медленно). Метод Многосеточный обходит эту проблему, используя ряд грубых сеток (они являются вложенными в первоначальную подробную сетку как матрёшки) таким образом, что имеющиеся компоненты вектора ошибки с большой длиной волны являются коротковолновыми (легко подавляемыми обычным итерационным методом - сглаживателем) на грубых сетках из иерархии вложенности. Для того, чтобы избежать необходимости в грубосеточного меширования геометрии с использованием ряда различных шагов сетки, в данной работе используется Алгебраический многосеточный метод.

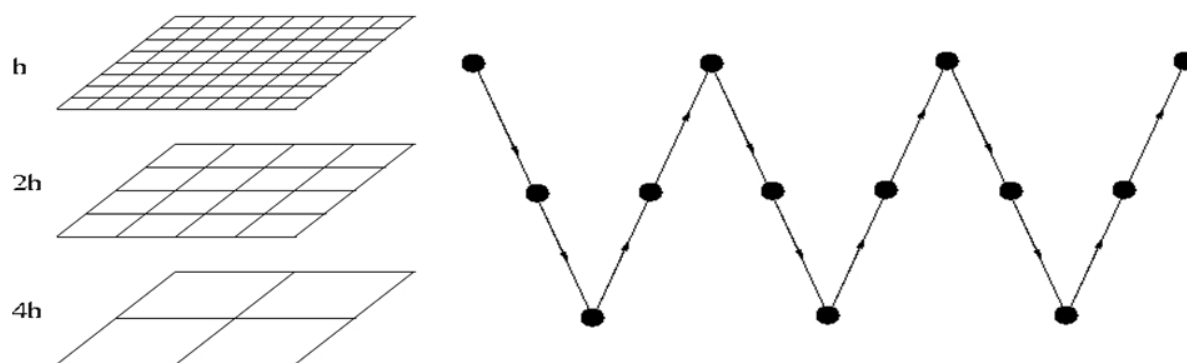
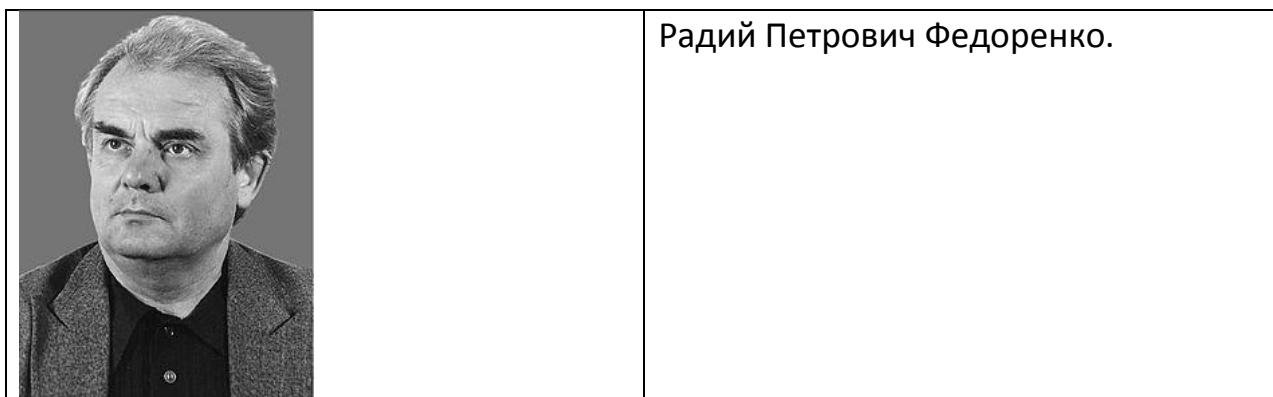


Рисунок 16 – Многосеточный метод решения СЛАУ.

Родоначальником многосеточного метода решения СЛАУ является наш соотечественник Радий Петрович Федоренко.



«Публикация (1961 г.) связана с попыткой применения численных методов для прогноза погоды, предпринятая тоже по инициативе Гельфанда (решение двумерных уравнений газовой динамики на поверхности сферы с учётом силы Кориолиса). При численном интегрировании этих уравнений на каждом шаге надо решать уравнение Пуассона, и именно на это уходит большая часть машинного времени. К счастью, я в то время не был знаком с наиболее эффективными итерационными методами и начал со "школьного" метода простой итерации, который сходится, но очень медленно, и скорость сходимости резко снижается при уменьшении шага сетки. Но мы уже имели опыт анализа возникающих трудностей - надо посмотреть изменение чисел в процессе работы алгоритма, и попытаться понять, что мешает его работе (вопрос только в том, какие числа смотреть и как смотреть, в этом должны помочь теория и интуиция). Увидев, что невязка быстро становится гладкой функцией и после этого очень медленно убывает, нетрудно было догадаться, что нужно решать уравнение для корректирующей функции на сетке с крупным шагом».

Метод решения СЛАУ состоит из внешней итерационной процедуры Крыловского типа (обычно BiCGStab[1] или FGMRes[1]), алгебраического многосеточного метода который используется как предобуславливатель к алгоритму Крыловского типа.

$$\begin{aligned}
 & r_0 \leftarrow f - Au_0, \beta = \|r_0\|, q_0 \leftarrow r_0/\|r_0\|, \xi \leftarrow (1, 0, \dots, 0)^T \\
 & \text{For } n \in \{0, 1, \dots\} \text{ while } \|r_n\| > \varepsilon\beta \\
 & \quad z_n \leftarrow B_n^{-1}q_n \\
 & \quad \tilde{q}_{n+1} \leftarrow Az_n \\
 & \quad \text{For } k \in [0, \dots, n] \\
 & \quad \quad H_{k,n} \leftarrow (\tilde{q}_{n+1}, q_k) \\
 & \quad \quad \tilde{q}_{n+1} \leftarrow \tilde{q}_{n+1} - H_{k,n}q_k \\
 & \quad \text{EndFor} \\
 & \quad H_{n+1,n} \leftarrow \|\tilde{q}_{n+1}\|, q_{n+1} \leftarrow \tilde{q}_{n+1}/H_{n+1,n} \\
 & \quad \text{For } k \in \{0, \dots, n-1\} \\
 & \quad \quad \begin{bmatrix} H_{k,n} \\ H_{k+1,n} \end{bmatrix} \leftarrow \begin{bmatrix} c_k & s_k \\ -\bar{s}_k & c_k \end{bmatrix} \begin{bmatrix} H_{k,n} \\ H_{k+1,n} \end{bmatrix} \\
 & \quad \text{EndFor} \\
 & \quad c_n \leftarrow |H_{n,n}|/\sqrt{|H_{n,n}|^2 + |H_{n+1,n}|^2} \\
 & \quad \bar{s}_n \leftarrow c_n H_{n+1,n}/H_{n,n} \\
 & \quad \begin{bmatrix} \xi_n \\ \xi_{n+1} \end{bmatrix} \leftarrow \begin{bmatrix} c_n & s_n \\ -\bar{s}_n & c_n \end{bmatrix} \begin{bmatrix} \xi_n \\ \xi_{n+1} \end{bmatrix} \\
 & \quad H_{n,n} \leftarrow c_n H_{n,n} + s_n H_{n+1,n}, H_{n+1,n} \leftarrow 0 \\
 & \quad \|r_{n+1}\| \leftarrow \beta|\xi_{n+1}| \\
 & \text{EndFor} \\
 & y_n \leftarrow \beta H^{-1}\xi \\
 & x_n \leftarrow x_0 + [z_0 \dots z_{n-1}]y_n.
 \end{aligned}$$

Метод FGMRES

Рисунок 17 – Гибкий вариант обобщенного метода минимальных невязок Ю.Саада и Мартина Г.Шульца.

Здесь B_n^{-1} – предобуславливание: один V цикл алгебраического многосеточного метода.

Блок схема алгоритма BiCGStab представлена на рисунке 16.

1. $x_0 =$ начальное приближение; $r_0 = b - Ax_0$
 2. $\rho_0 = (r_0, r_0)$
 3. $p_0 = r_0$
 - для $j = 0, 1, \dots$
 4. $v = Ap_j$
 5. $\alpha_j = \frac{\rho_j}{(v, r_0)}$
 6. $s_j = r_j - \alpha_j v$
 7. $t = As_j$
 8. $\omega_j = \frac{(t, s_j)}{(t, t)}$
 9. $x_{j+1} = x_j + \alpha_j p_j + \omega_j s_j$; проверка сходимости
 10. $r_{j+1} = s_j - \omega_j t$
 11. $\rho_{j+1} = (r_{j+1}, r_0)$
 12. $\beta_j = \frac{\rho_{j+1} \alpha_j}{\rho_j \omega_j}$
 13. $p_{j+1} = r_{j+1} + \beta_j (p_j - \omega_j v)$
- конец

Рисунок 18 – Классический BiCGStab алгоритм без предобуславливания.

Метод BiCGStab[1],[4] относится к классу итерационных методов Крыловских подпространств для решения СЛАУ вида $Ax = b$. BiCGStab является «стабилизированным» вариантом метода бисопряженных градиентов BiCG, он обеспечивает устойчивость вычислений и более гладкую по сравнению с BiCG сходимость к решению за счет устранения матрично-векторных операций с транспонированной матрицей системы. Применение метода BiCGStab требует реализации следующих операций:

- вычисление скалярного произведения векторов: $a = (x, y)$;
- вычисление матрично-векторного умножения: $y = Ax$;
- применение операции предобуславливания: $y = M^{-1} x \approx A^{-1} x$, где M^{-1} – подходящий линейный оператор, матрица которого довольно часто не задается явно.

Во многом качество работы итерационного метода зависит от качества используемого предобуславливателя, поскольку процедура предобуславливания является наиболее ресурсоемким элементом алгоритма. Кроме того, т.к. процедура распараллеливания скалярного произведения и матрично-векторного умножения может быть проведена стандартным образом без особого труда, эффективность параллельной реализации общего алгоритма зависит от степени параллелизма, которую допускает используемый метод предобуславливания.

Алгоритм 1 Алгоритм предобусловленного метода BiCGStab для решения системы линейных алгебраических уравнений $A\mathbf{x} = \mathbf{b}$.

1. Инициализация: выбрать начальное приближение \mathbf{x}_0 .
 2. Вычислить $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$.
 3. Выбрать произвольный \mathbf{r}_0^* , такой, что $(\mathbf{r}_0^*, \mathbf{r}_0) \neq 0$, например $\mathbf{r}_0^* = \mathbf{r}_0$.
 4. Задать $\rho_0 = \alpha = \omega_0 = 1$, $\mathbf{v}_0 = \mathbf{p}_0 = 0$.
 5. Для $k = 1, 2, 3 \dots$
 - (a) Вычислить $\rho_k = (\mathbf{r}_0^*, \mathbf{r}_{k-1})$.
 - (b) Вычислить $\beta = (\rho_k / \rho_{k-1})(\alpha / \omega_{k-1})$.
 - (c) Вычислить $\mathbf{p}_k = \mathbf{r}_{k-1} + \beta(\mathbf{p}_{k-1} - \omega_{k-1}\mathbf{v}_{k-1})$.
 - (d) Вычислить $\mathbf{y} = M^{-1}\mathbf{p}_k$.
 - (e) Вычислить $\mathbf{v}_k = A\mathbf{y}$.
 - (f) Вычислить $\alpha = \rho_k / (\mathbf{r}_0^*, \mathbf{v}_k)$.
 - (g) Вычислить $\mathbf{s} = \mathbf{r}_{k-1} - \alpha\mathbf{v}_k$.
 - (h) Проверка сходимости: если $\|\mathbf{s}\|_2 < \text{atol}$, вычислить $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha\mathbf{y}$ и выйти из итераций.
 - (i) Вычислить $\mathbf{z} = M^{-1}\mathbf{s}$.
 - (j) Вычислить $\mathbf{t} = A\mathbf{z}$.
 - (k) Вычислить $\omega_k = (M^{-1}\mathbf{t}, M^{-1}\mathbf{s}) / (M^{-1}\mathbf{t}, M^{-1}\mathbf{t})$.
 - (l) Вычислить $\mathbf{r}_k = \mathbf{s} - \omega_k\mathbf{t}$.
 - (m) Вычислить $\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha\mathbf{y} + \omega_k\mathbf{z}$.
 - (n) Проверка сходимости: если $\|\mathbf{r}_k\|_2 < \text{rtol} \cdot \|\mathbf{r}_0\|_2 + \text{atol}$, выйти из итераций.
-

Рисунок 19 – Предобусловленный алгоритм BiCGStab.

Предобусловленный алгоритм Хенка Ван дер Ворста BiCGStab[4] использует на одной своей итерации два V – цикла алгебраического многосеточного метода.

Алгебраический многосеточный метод состоит из SETUP Phase и SOLUTION Phase[3],[1],[6].

Setup Phase:

- Select coarse “grids”
- Define interpolation, $P^{(m)}$, $m = 1, 2, \dots$
- Define restriction and coarse-grid operators

$$R^{(m)} = P^{(m)T} \quad A^{(m+1)} = P^{(m)T} A^{(m)} P^{(m)}$$

Solve Phase

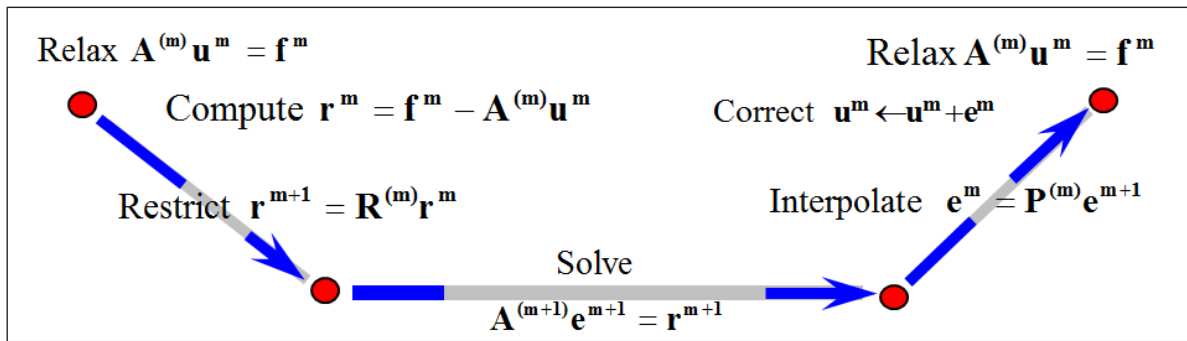


Рисунок 20 – Вычислительная схема многосеточного алгоритма.

SETUP Phase основана на алгоритмах селектора и разреженного матричного умножения. Селекторы: Руге-Стубен, метод агрегации, метод сглаженной агрегации, метод сглаженной агрегации с минимальной энергией, CLJP, Falgout(hybrid RS-CLJP), PMIS и др.

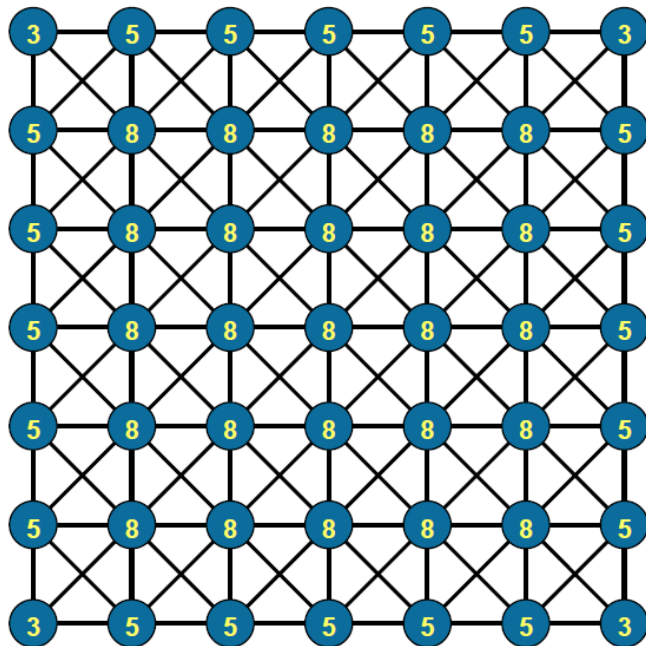
Цель селектора – пометить узлы, которые войдут в следующий более грубый уровень сетки, и при этом будут удовлетворять определенным критериям качества разбиения. Селектор работает в полностью автоматическом режиме, без какого либо участия пользователя на всем многообразии пользовательских входных данных.

Классический Руге-Стубен селектор [2]. Исторически первый. 1985 год.

Достоинства Руге-Стубен(RS) алгоритма огрубления: создает более сильное предобуславливание, чем остальные селекторы. Хорошо документирован.

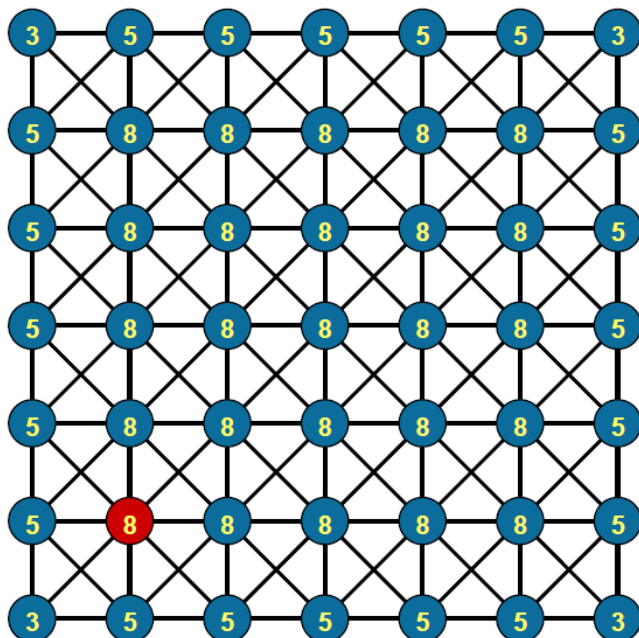
Недостатки RS: однопоточный, большая операторная сложность, что нежелательно для ряда задач.

C-AMG coarsening



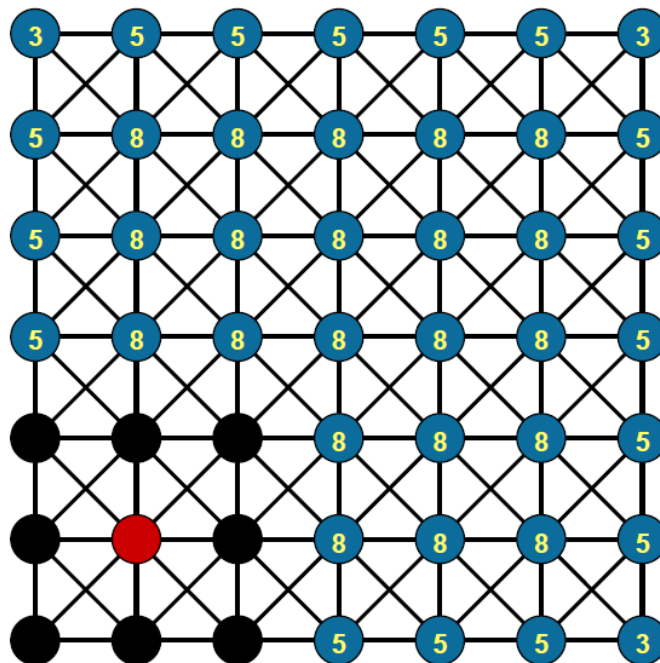
- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening



- select C-pt with maximal measure
- select neighbors as F-pts
- update measures of F-pt neighbors

C-AMG coarsening

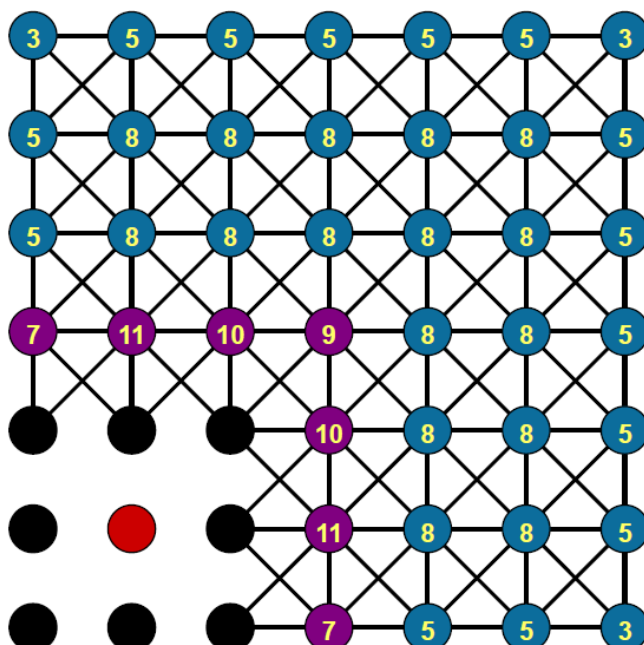


→ select C-pt with maximal measure

→ select neighbors as F-pts

→ update measures of F-pt neighbors

C-AMG coarsening

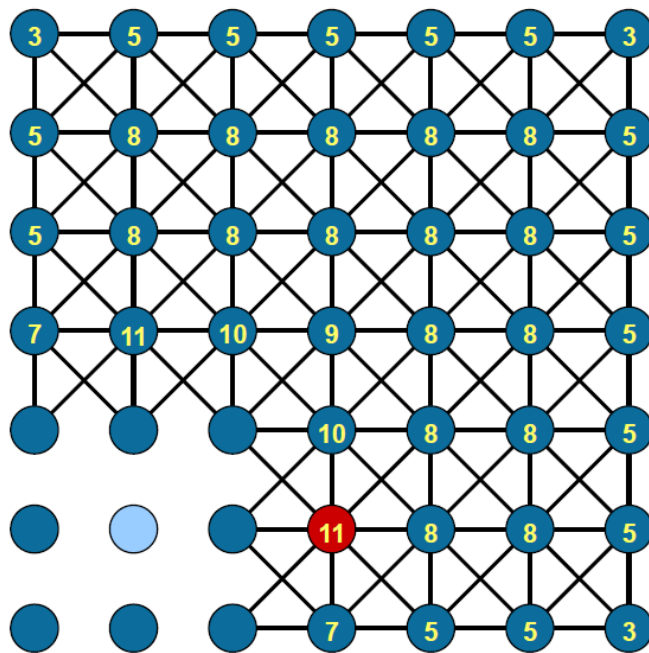


→ select C-pt with maximal measure

→ select neighbors as F-pts

→ update measures of F-pt neighbors

C-AMG coarsening

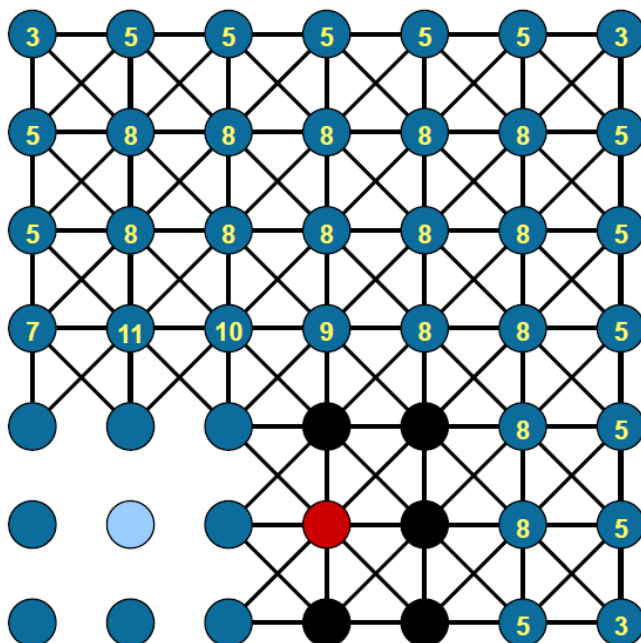


→ select C-pt with maximal measure

→ select neighbors as F-pts

→ update measures of F-pt neighbors

C-AMG coarsening

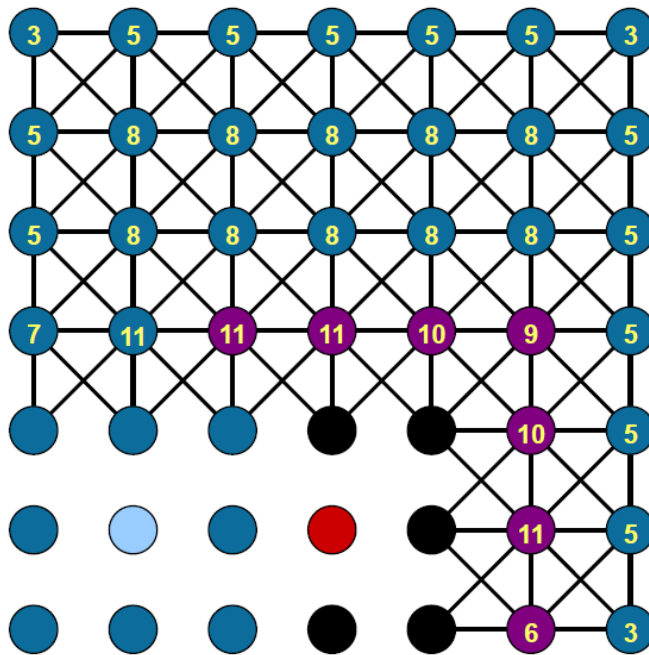


→ select C-pt with maximal measure

→ select neighbors as F-pts

→ update measures of F-pt neighbors

C-AMG coarsening

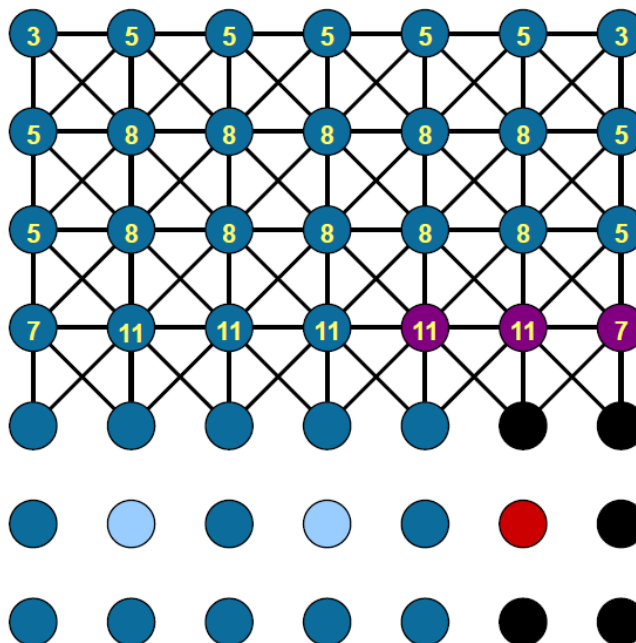


→ select C-pt with maximal measure

→ select neighbors as F-pts

→ update measures of F-pt neighbors

C-AMG coarsening

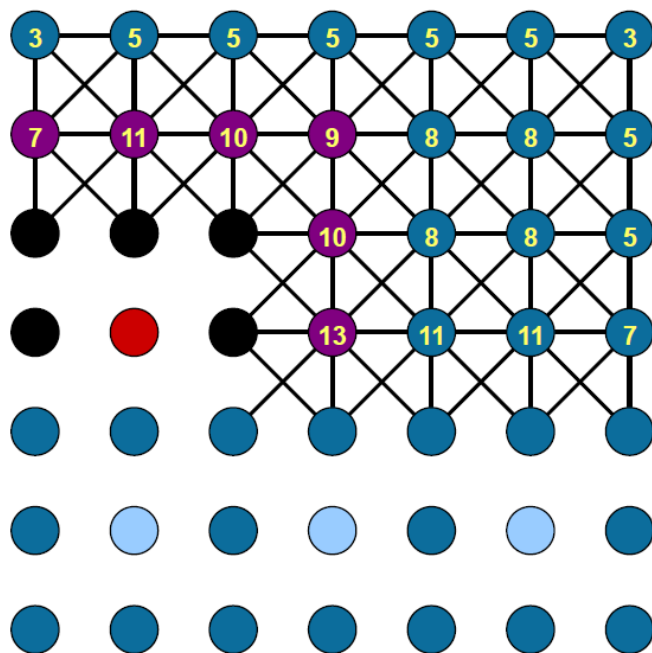


→ select C-pt with maximal measure

→ select neighbors as F-pts

→ update measures of F-pt neighbors

C-AMG coarsening

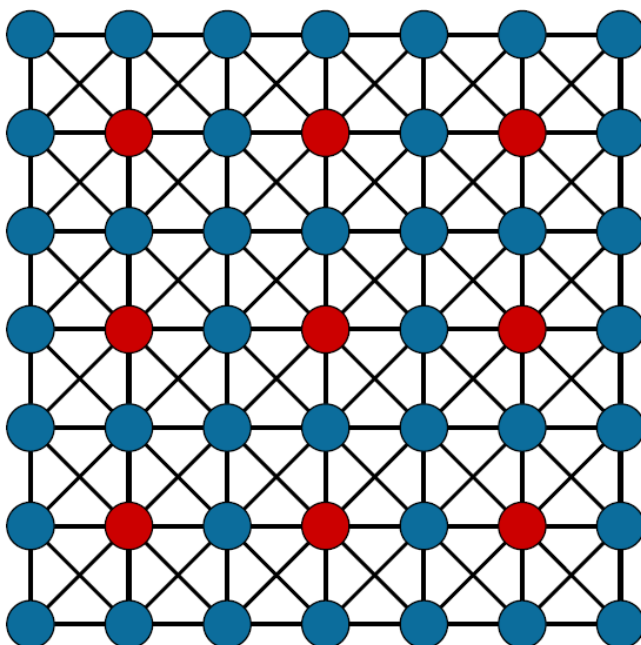


→ select C-pt with maximal measure

→ select neighbors as F-pts

→ update measures of F-pt neighbors

C-AMG coarsening is inherently sequential



→ select C-pt with maximal measure

→ select neighbors as F-pts

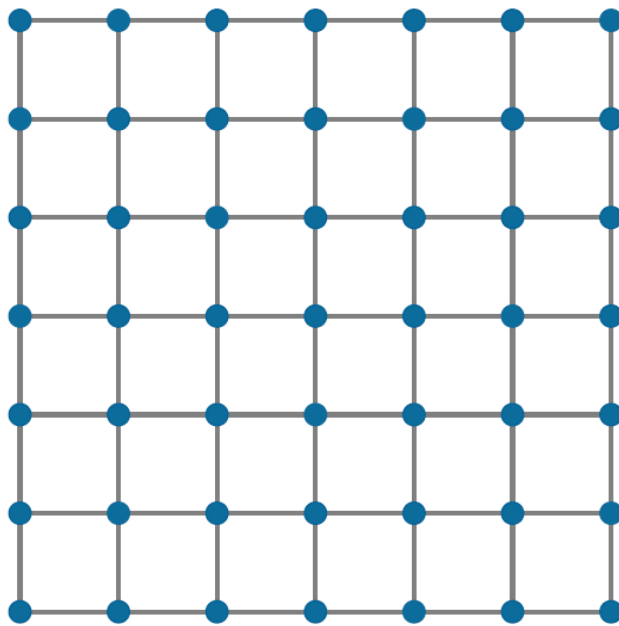
→ update measures of F-pt neighbors

Рисунок 21 — Работа RS (Pyre-Стубен) алгоритма огрубления.

В результате однопоточной работы Рунге-Стубен алгоритма огрубления красные узлы составляют сетку следующего уровня вложенности многосеточного алгоритма.

Селектор метода агрегации [2].

SA coarsening (5-pt Laplacian)



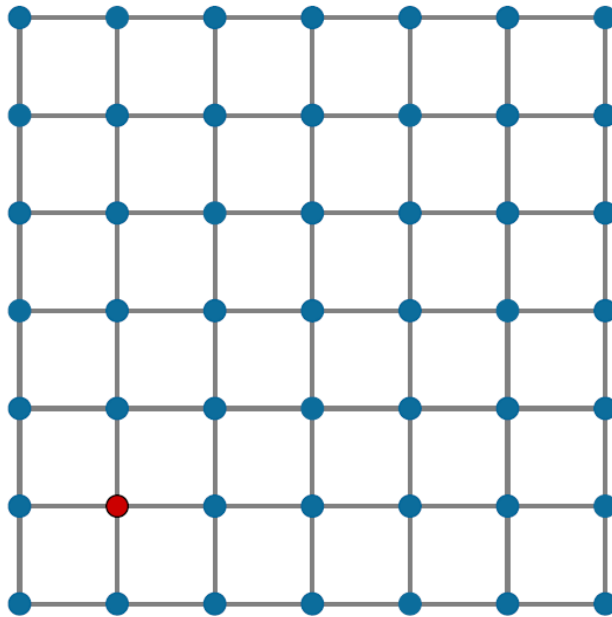
Phase 1:

- a) *Pick root pt not adjacent to agg*
- b) *Aggregate root and neighbors*

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



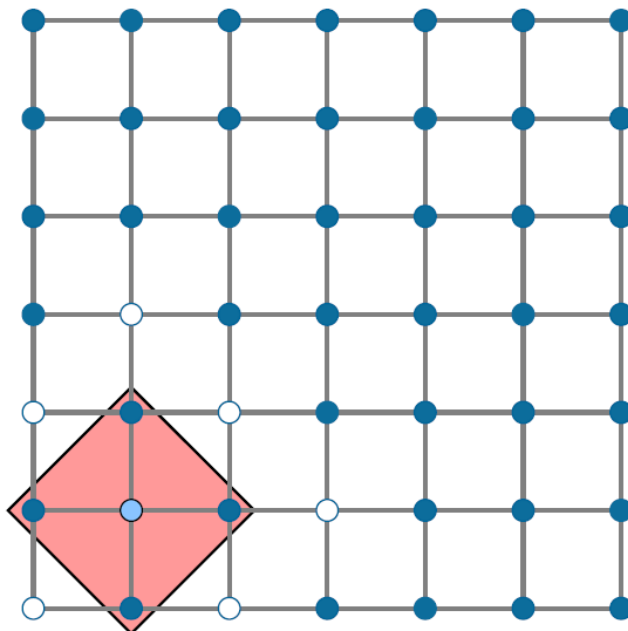
Phase 1:

- a) *Pick root pt not adjacent to agg*
- b) *Aggregate root and neighbors*

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



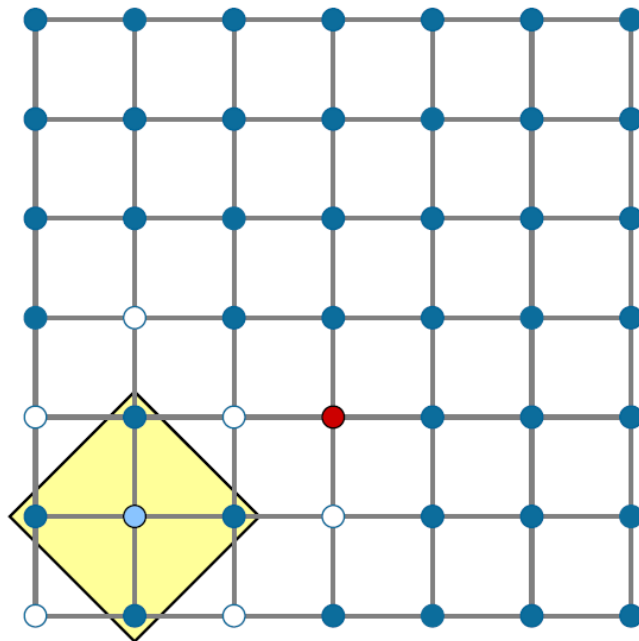
Phase 1:

- a) *Pick root pt not adjacent to agg*
- b) *Aggregate root and neighbors*

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



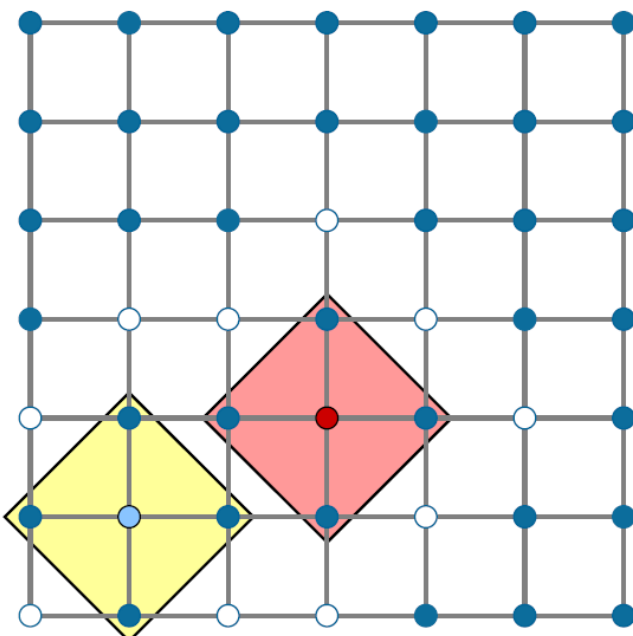
Phase 1:

- a) Pick root pt not adjacent to agg*
- b) Aggregate root and neighbors*

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



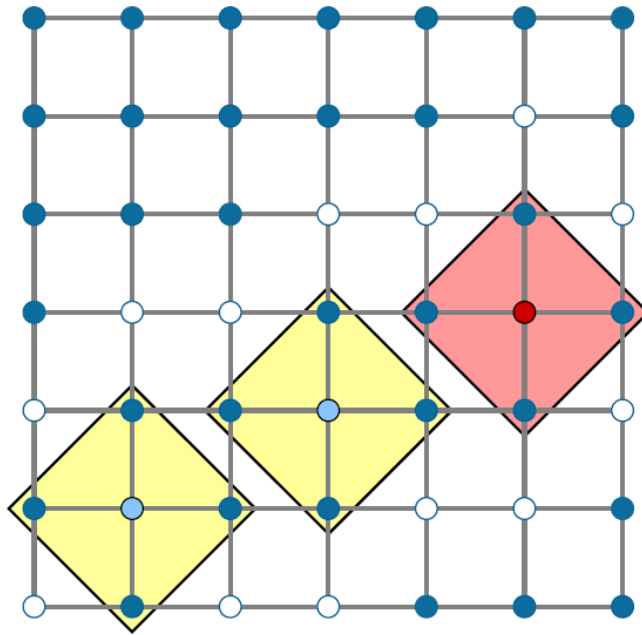
Phase 1:

- a) Pick root pt not adjacent to agg*
- b) Aggregate root and neighbors*

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



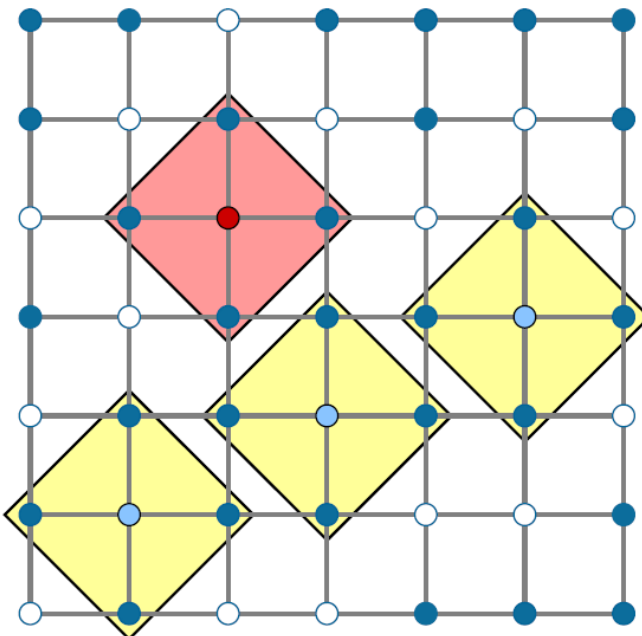
Phase 1:

- a) Pick root pt not adjacent to agg
- b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



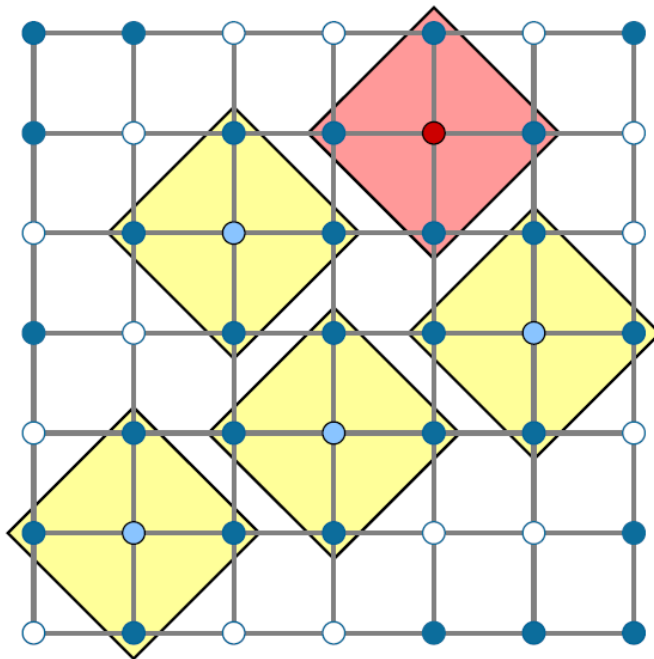
Phase 1:

- a) Pick root pt not adjacent to agg
- b) Aggregate root and neighbors

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



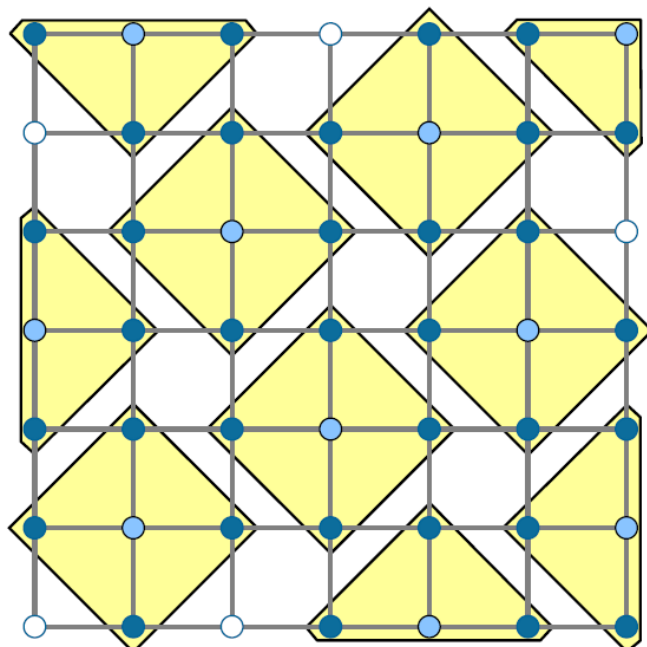
Phase 1:

- a) Pick root pt not adjacent to agg*
- b) Aggregate root and neighbors*

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



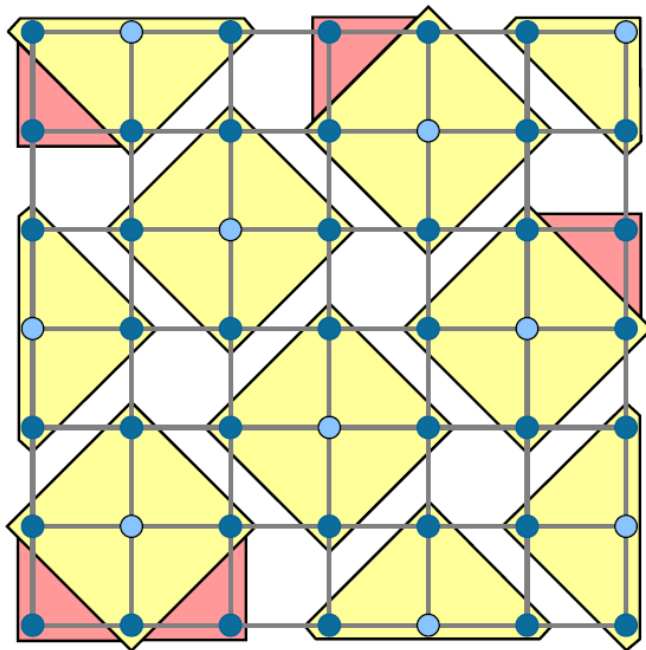
Phase 1:

- a) Pick root pt not adjacent to agg*
- b) Aggregate root and neighbors*

Phase 2:

Move pts into nearby aggs or new aggs

SA coarsening (5-pt Laplacian)



Phase 1:

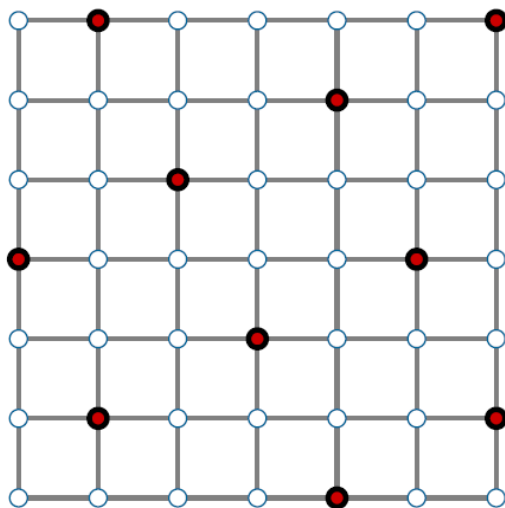
- a) Pick root pt not adjacent to agg
- b) Aggregate root and neighbors

Phase 2:

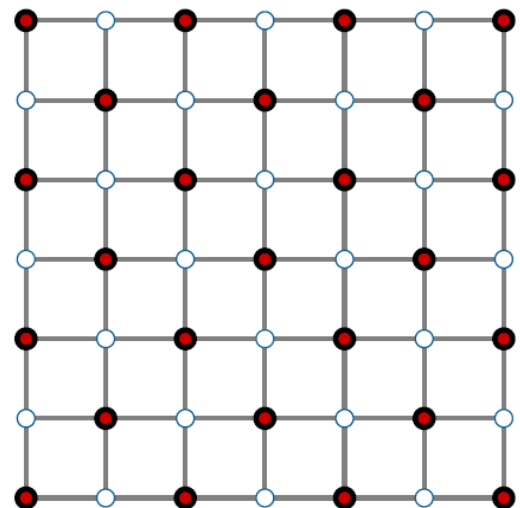
Move pts into nearby aggs or new aggs

SA coarsening is traditionally more aggressive than C-AMG coarsening (5-pt Laplacian example)

SA Seed Points (10)



C-AMG Grid (25)



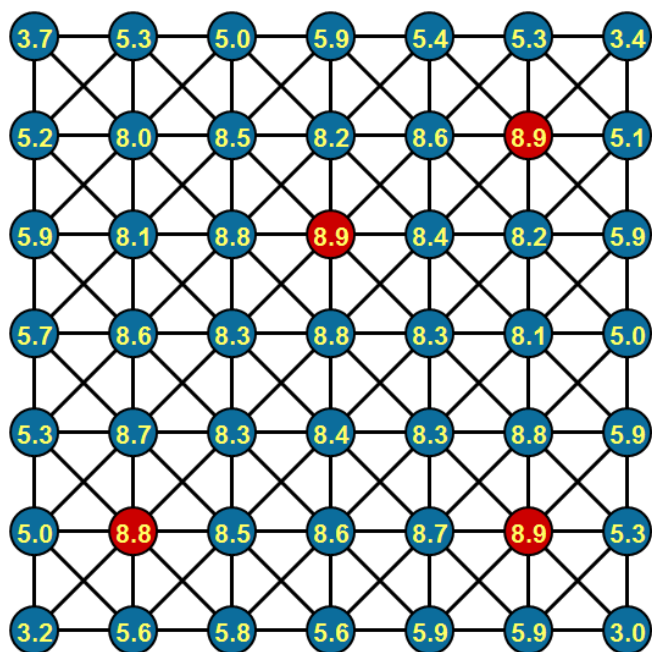
Operator complexities are usually smaller, too

Рисунок 22 — Метод агрегации в качестве селектора.

В результате работы метода агрегации красные узлы составляют сетку следующего уровня вложенности многосеточного алгоритма. Приводится сравнение с RS селектором.

PMIS селектор [3]. Достоинства: малая операторная сложность, параллельное исполнение.

PMIS select 1

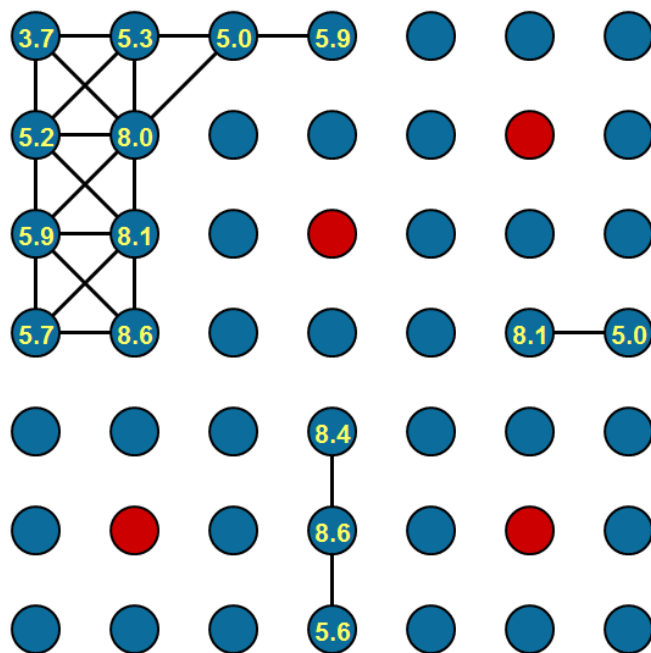


→ select C-pts with maximal measure locally

→ make neighbor F-pts

→ remove neighbor edges

PMIS: remove and update 1

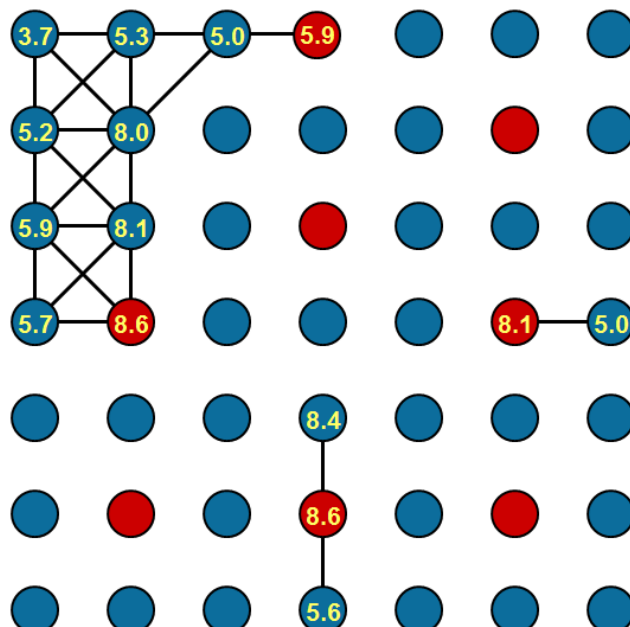


→ select C-pts with maximal measure locally

→ make neighbors F-pts

→ remove neighbor edges

PMIS: select 2

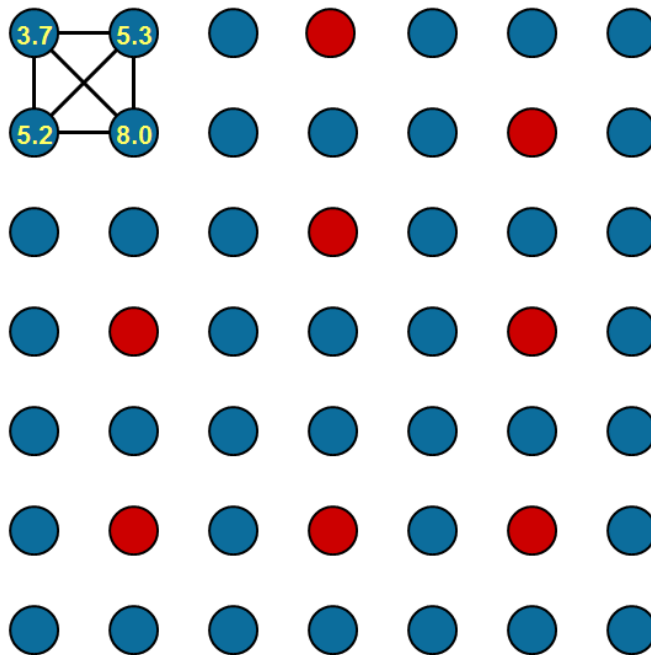


→ select C-pts with maximal measure locally

→ make neighbors F-pts

→ remove neighbor edges

PMIS: remove and update 2

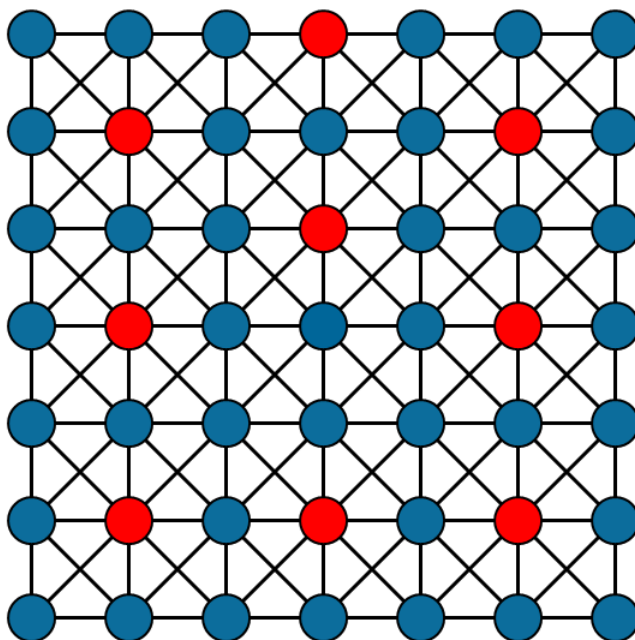


→ select C-pts with maximal measure locally

→ make neighbors F-pts

→ remove neighbor edges

PMIS: final grid



→ select C-pts with maximal measure locally

→ make neighbor F-pts

→ remove neighbor edges

Рисунок 23 — Работа PMIS алгоритма огрубления.

В результате параллельной работы PMIS алгоритма красные узлы составляют сетку следующего уровня вложенности многосеточного алгоритма.

В качестве алгоритма разреженного матричного умножения используется алгоритм Ф. Густавсона (IBM), а для операции транспонирования матрицы применяется сортировка с подсчетом Саусвела, работающая за линейное время.

SOLUTION Phase обычно состоит из V-цикла многосеточного метода в ходе работы которого используется та или иная релаксационная процедура. Релаксационными процедурами являются: метод Гаусса –Зейделя (Gauss-Seidel), метод нижней релаксации Якоби (damped Jacoby), треугольный кососимметричный метод, различные варианты неполного LU разложения: $ilu0$, $iluk$, $k=0,1,2..$, $ilut$ – ilu threshold, метод Чебышева, SPAI0, SPAI1, полиномиальные сглаживатели Рунге-Кутты 3 или 5 порядков, CF – версии сглаживателей которые при нисходящем подходе первыми обходят C узлы, а при восходящем F (F - релаксация). Часто пренебрегают предсглаживаниями и используют только постсглаживания.

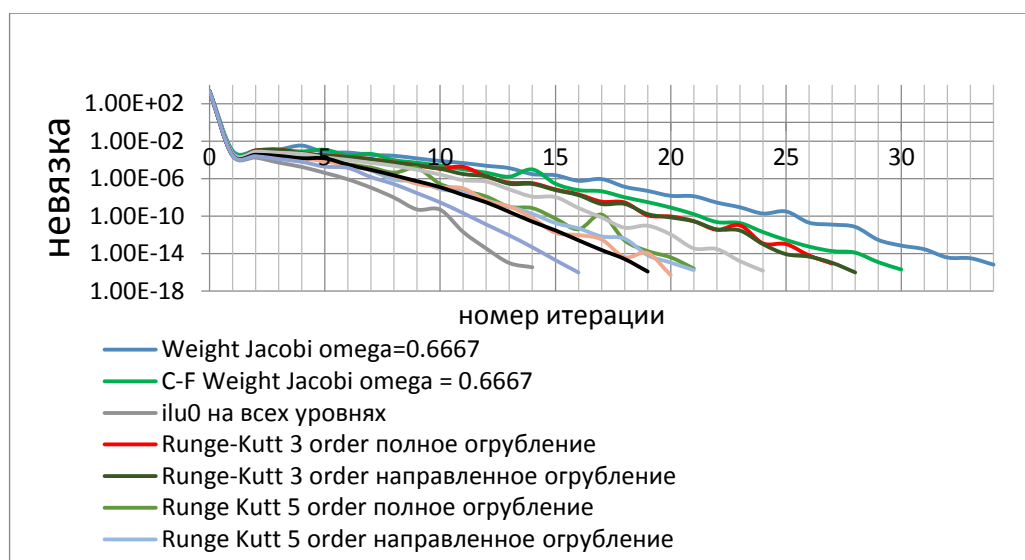


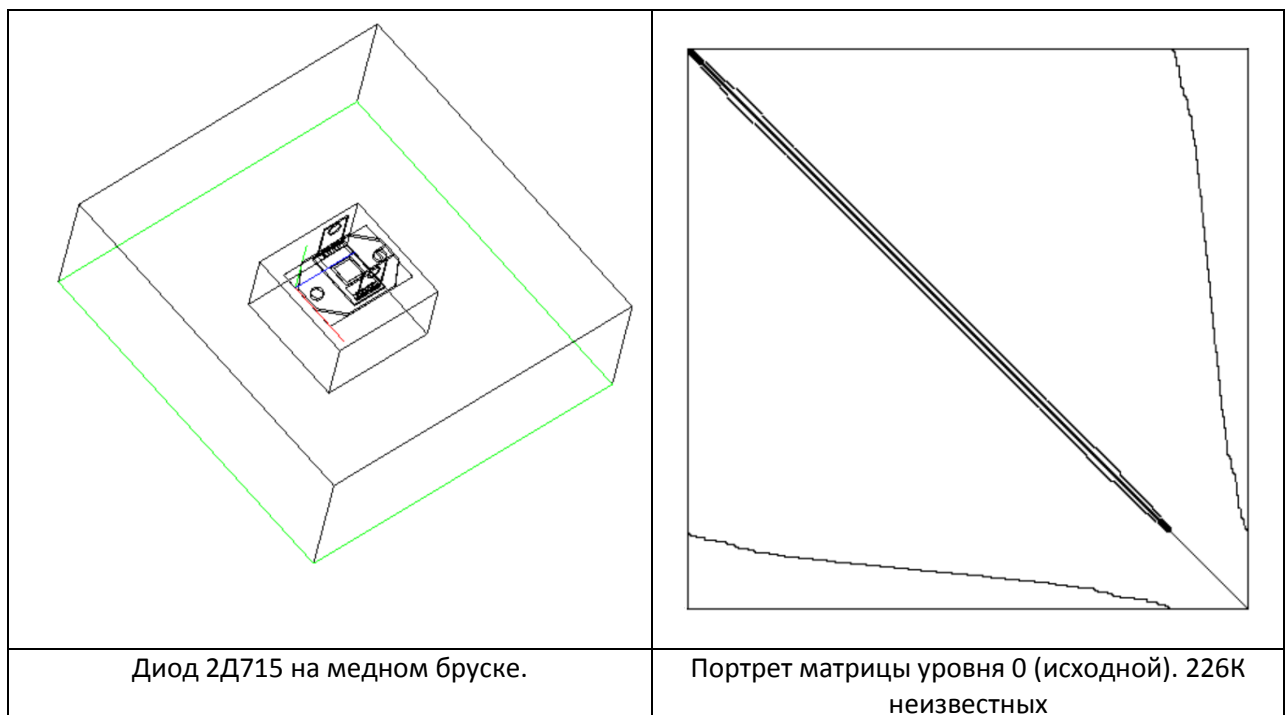
Рисунок 24 — Зависимость скорости сходимости алгебраического многосеточного метода от типа используемой сглаживающей процедуры.

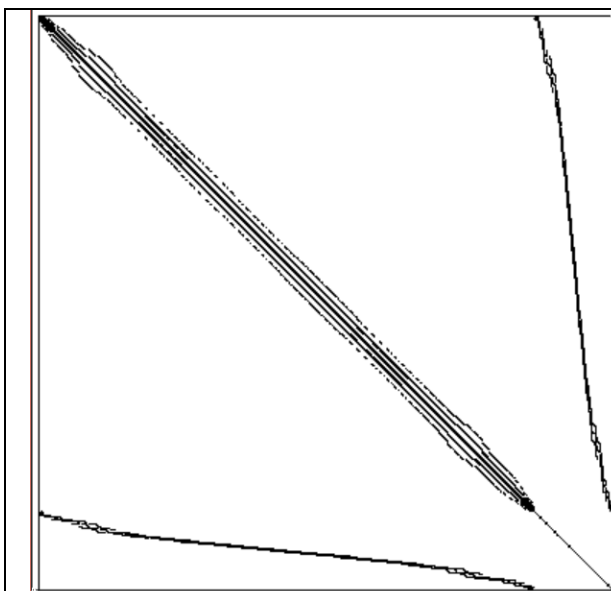
Алгебраические многосеточные методы (AMG) являются линейно масштабируемыми $O(n)$ в процессе решения для многих крупных разреженных линейных систем. Иерархии более грубых сеток прогрессивно сконструированы таким образом, чтобы использовать взаимодополняющие сглаживающие невязку релаксации и операторы грубосеточной

интерполяции. Высоко осциллирующая составляющая ошибки начального приближения уменьшается с помощью релаксации, в то время как плавная составляющая ошибки разрешается на грубой сетке -матрице и сокращается там. Однако, большие параллельные расходы на связь часто ограничивают масштабируемость параллельных реализаций AMG. Иерархия матриц в многосеточном методе, формируется с помощью матрицы каждого подробного уровня через тройное произведение разреженных матриц. В результате на грубых сетках часто имеется существенно большее число ненулевых элементов в строке, чем в оригинальном первоначальном операторе подробного уровня, тем самым создавая высокие параллельные расходы на связь связанные с разреженным умножением матрицы на вектор (SpMV) на грубых уровнях.

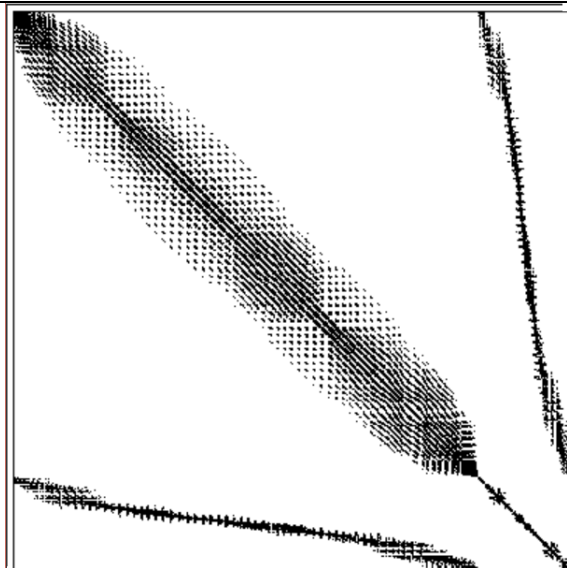
На рисунке 25 представлено, как меняется портрет матрицы СЛАУ при её ассемблировании внутри классического многосеточного метода Рунге и Стубена, $\theta=0.25$.

Портрет матрицы – графическое изображение её ненулевых коэффициентов.

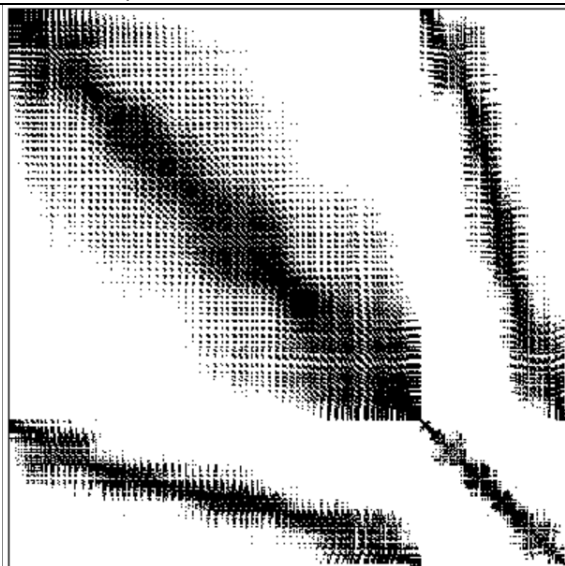




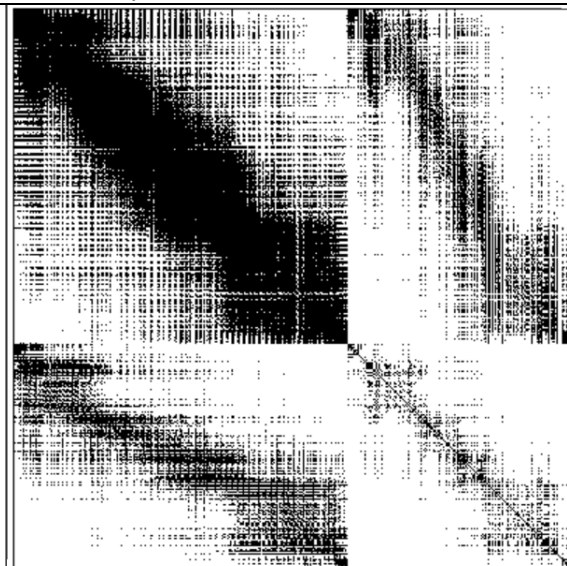
Уровень 1. 113K неизвестных



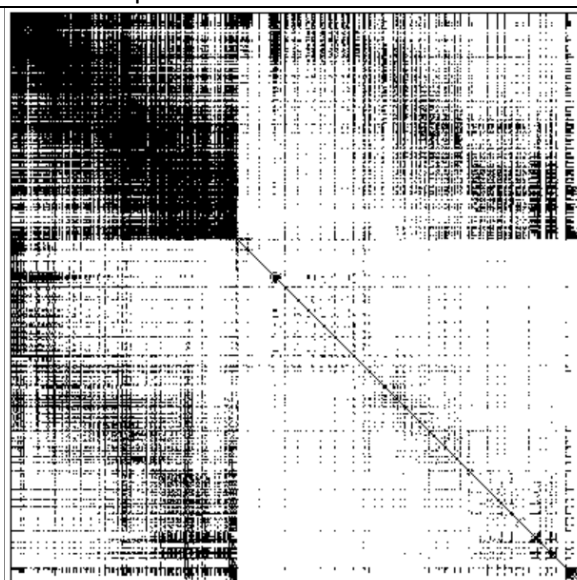
Уровень 2. 39K неизвестных.



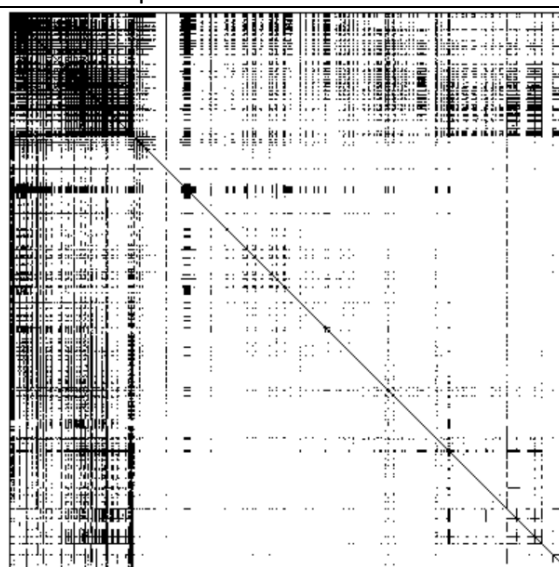
Уровень 3. 13K неизвестных.



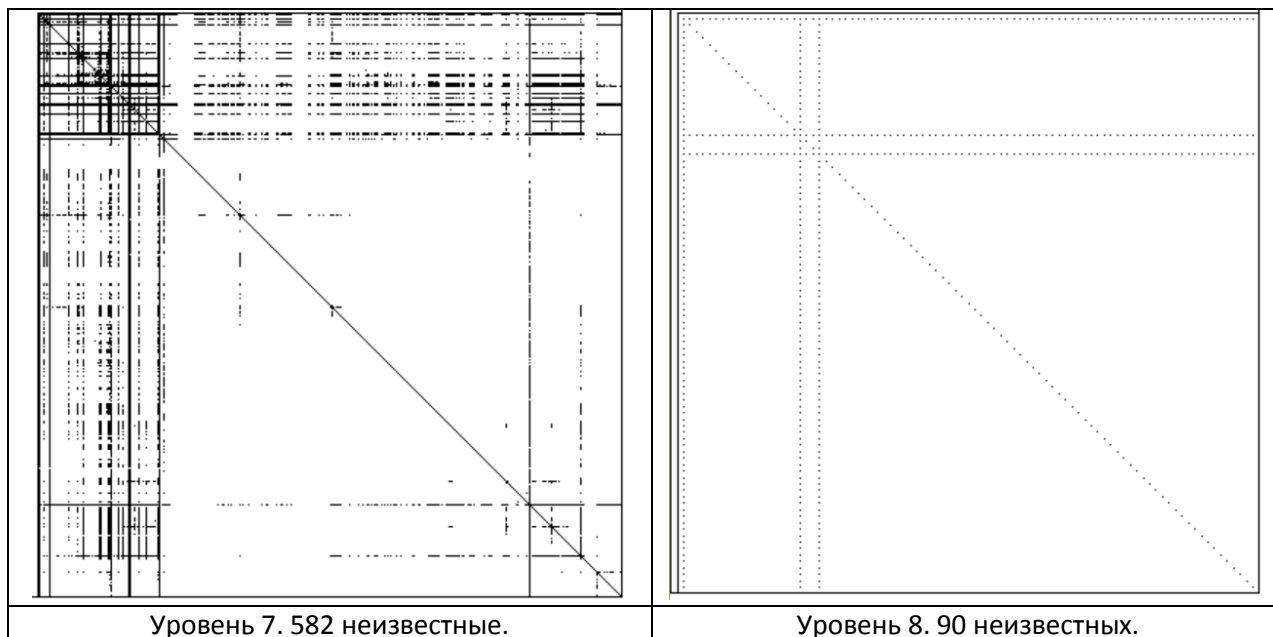
Уровень 4. 5K неизвестных.



Уровень 5. 2K неизвестных.



Уровень 6. 1K неизвестных.



Тепловое сопротивление диода. Время расчёта 12с 390мс. Тепловое сопротивление кристалл –основание медного бруса $\theta=0,57\text{ }^{\circ}\text{C/Wm}$. Тепловое сопротивление кристалл-основание корпуса $0,32\text{ }^{\circ}\text{C/Wm}$.

Рисунок 25 — Изменение портрета матрицы СЛАУ при её ассемблировании внутри многосеточного метода Рунге-Стубена, $\theta=0.24$.

Для уменьшения размера сеточного шаблона на глубоких уровнях вложенности применяют селекторы дающие меньшую операторную сложность. Для Рунге – Стубен алгоритма можно уменьшить операторную сложность, задав $\theta=0.9$, а потерю скорости сходимости скомпенсировать $ilu0$ релаксационной процедурой. Так же применяют поп Галёркин AMG.

Алгоритм решения уравнения теплопередачи

Алгоритм расчёта поля температур включая вакуумные промежутки.

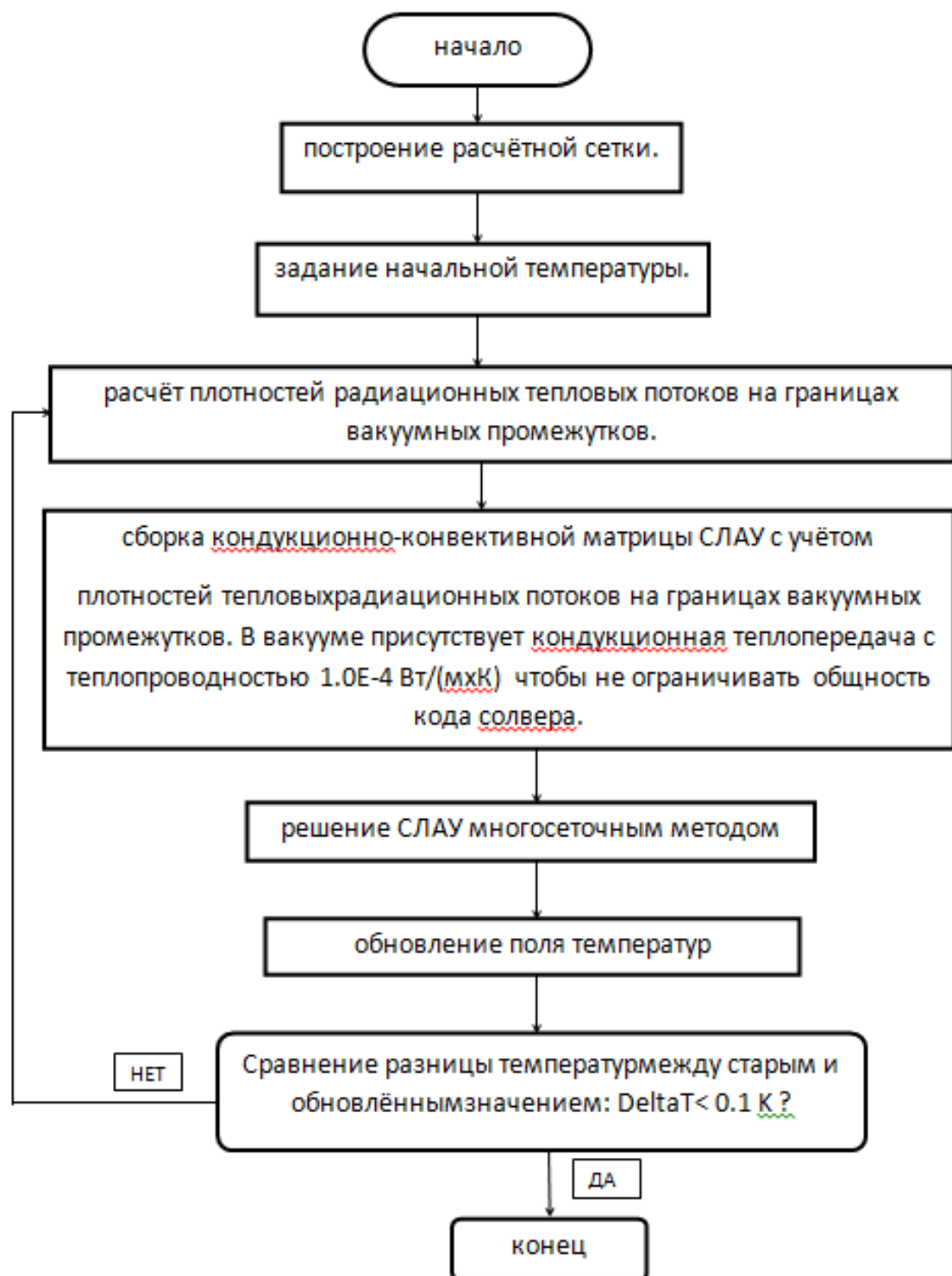


Рисунок 26 — Алгоритм решения уравнения теплопередачи.

Литература

1. Библиотека на языке FORTRAN. SPARSKIT 2. Y.Saad./ Итерационные методы для разреженных линейных систем, Ю.Саад, М. МГУ 2013 в переводе Х.Д. Икрамова.
2. An Algebraic Multigrid Tutorial. Robert D. Falgout.
3. Reducing Complexity in Algebraic Multigrid. Hans De Sterck, Ulrike Meier Yang.
4. Моделирование перехода между регулярным и маховским отражением ударных волн с помощью неявной схемы на основе методов LU-SGS и BiCGStab. В.Е. Борисов, А.Е. Луцкий.
5. ANES/20XE : Код для численного моделирования процессов гидродинамики и тепломассообмена. Версия 2.20.
6. Методы ускорения газодинамических расчётов на неструктурированных сетках. Под редакцией проф. В.Н. Емельянова, М. ФИЗМАТЛИТ, 2014
7. С.Патанкар Численные методы теплообмена и динамики жидкости. М.:ЭНЕРГОАТОМИЗДАТ, 1984.
8. М.Э.Рояк, Ю.Г.Соловейчик, Э.П.Шурина Сеточные методы решения краевых задач математической физики. Учебное пособие. Новосибирск 1998.
9. http://tesis.com.ru/own_design/flowvision/calc_mesh.php