Нижегородский государственный университет
им. Н.И.Лобачевского

Факультет Вычислительной математики и кибернетики

Параллельные численные методы

Решение разреженных СЛАУ

При поддержке компании Intel

Баркалов К.А.,
Кафедра математического обеспечения ЭВМ

Содержание

- Разреженные матрицы
  - Понятие, форматы хранения
- Разложение Холецкого для разреженных матриц
  - Проблема заполнения фактора, переупорядочивание матрицы как метод снижения заполненности
- Графовая модель системы уравнений
  - Последовательность графов исключения, дерево исключения
- Переупорядочивание матрицы
  - Метод минимальной степени
  - Метод вложенных сечений
  - Ресурсы для паралеллизма
- Примеры работы алгоритмов переупорядочивания

Разреженные матрицы

- Разреженная матрица – это матрица порядка *N*, в которой:
  – Малый процент ненулевых элементов; или
  – *O*(*N*) ненулевых элементов; или
  – В каждой строке не более $k<<N$ ненулевых элементов.
- Разреженная матрица – это матрица, для которой:
  – Есть алгоритмы, учитывающий разреженность, более эффективные, чем «плотные» алгоритмы.
- Обработка разреженных матриц: нетривиальные алгоритмы
- Хранение разреженных матриц: особый формат
  – матрица размера *NxN*,
  – *NZ* ненулевых элементов
  – $NZ << N^2$

Координатный формат

- Матрица хранится в виде трех массивов:
  - Массив значений *Value*
  - Массив номеров строк *Row*
  - Массив номеров столбцов *Col*

A

1				2	
		3	4		
			8		5
	7	1			6

Структура хранения:

1	2	3	4	8	5	7	1	6

Value

0	0	1	1	3	3	5	5	5

Row

0	4	2	3	3	5	1	2	5

Col

Координатный формат

- Объем памяти для хранения в разных форматах:
  - Плотный: $M = 8\ N^2$ байт
  - Координатный: $M = 8\ NZ + 4\ NZ + 4\ NZ = 16\ NZ << 8\ N^2$.
- Координатный формат:
  - Упорядоченный (по строкам/столбцам)
    - Быстрый доступ к строкам/столбцам
    - Перепаковки при вставке/удалении элементов
  - Неупорядоченный
    - «Переборный» доступ к элементам
    - Быстрая вставка/удаление элементов
- В целом координатный формат прост в реализации, но недостаточно эффективен (память, быстродействие)

Разреженный строчный формат

- Широко известен как CRS (Compressed Row Storage) или CSR(Compressed Sparse Rows);
- Матрица хранится в виде трех массивов:
  - Массив значений *Value* (построчно, сверху вниз);
  - Массив номеров столбцов *Col;*
  - Массив индексов начала строк *RowIndex* в массиве *Col*.
- RowIndex[i] указывает на начало *i*-ой строки
- Элементы строки *i* в массиве Value находятся по индексам от RowIndex[i] до RowIndex[i + 1] – 1 включительно
  - Обрабатываются пустые строки (RowIndex[i] = RowIndex[i + 1])
  - Единообразно обрабатывается последняя строка (RowIndex[N+1]=NZ).

Разреженный строчный формат

- Объем памяти для хранения в разных форматах:
  - Плотный: $M = 8\ N^2$ байт
  - Координатный: $M = 16\ NZ$.
  - CRS : $M = 8\ NZ + 4\ NZ + 4\ (N + 1) = 12\ NZ + 4\ N + 4$

A

1				2	
		3	4		
			8		5
	7	1			6

Структура хранения:

1	2	3	4	8	5	7	1	6

Value

0	4	2	3	3	5	1	2	5

Col

0	2	4	4	6	6	9

RowIndex

Модификации формата CRS

- В CRS строки рассматриваются по порядку (быстрый доступ к строке), но элементы внутри строки могут быть:
  - Упорядочены (быстрый поиск элемента, нужно поддерживать упорядоченность);
  - Не упорядочены (переборный поиск, упорядоченность не поддерживается).
- CRS с четырьмя массивами.
  - Три массива – аналогично CRS.
  - Четвертый массив хранит индексы элементов, идущих в конце строки.
  - строки могут не быть упорядочены (перестановка строк проводится без перепаковки, только изменением индексов.

Разреженный столбцовый формат

- Широко известен как CCS (Compressed Column Storage) или CSR(Compressed Sparse Columns);
- Матрица хранится в виде трех массивов:
  – Массив значений *Value* (построчно, сверху вниз);
  – Массив номеров строк *Row;*
  – Массив индексов начала столбцов *ColIndex*.
- ColIndex[j] указывает на начало *j*-го стоблца
- j-й столбец в массиве Value находится по индексам от ColIndex[j] до ColIndex[j + 1] – 1 включительно
  – Обрабатываются пустые столбцы (ColIndex[j] = ColIndex[j + 1])
  – Единообразно обрабатывается последний столбец (ColIndex[N+1]=NZ).

Разреженный столбцовый формат

- Объем памяти CCS равен CRS
- Быстрый доступ к столбцам
- Модификации – аналогично CRS

A

1				2	
		3	4		
			8		5
	7	1			6

Структура хранения:

1	7	3	1	4	8	2	5	6

Value

0	5	1	5	1	3	0	3	5

Row

0	1	2	4	6	7	9

ColIndex

Другие форматы хранения

- Ленточная матрица
  - Храним только ленту в виде набора массивов.
- Матрица с плотной диагональю
  - Диагональ хранится в массиве *Diag.*
  - Остальные элементы матрицы хранятся в разреженном формате
- Симметричная матрица
  - Храним только верхний треугольник (в разреженном формате)
- Блочная матрица с плотными блоками
  - Блоки хранятся в разреженном формате
  - Элементы внутри блока хранятся в плотном формате

Умножение матрицы на вектор

- Исходные данные:
  - Разреженная матрица $A(N \times N)$ в формате CRS
  - Плотный вектор x
- Найти: $c=Ax$, $$c_i = \sum_{j=1}^{N} a_{ij} x_j$$
- Формат CRS - эффективный доступ к строкам A

```
For i:= 1 to N
  y[i]:= 0;
  for  j:= RowIndex[i] to (RowIndex[i + 1] – 1)
     y[i]:= y[i] + Value[j] * x[Col[j]];
endfor
```

Транспонирование матрицы

- Дано: разреженная матрица $A(N \times N)$ в формате CRS
- Найти: A^T в формате CRS
- Алгоритм
  – Сформируем N «целых» и N «вещественных» векторов.
  – В цикле просматриваем все строки исходной матрицы, для каждой строки – все ее элементы.
  – Пусть A[i][j] = *v*. Тогда добавим числа *i* и *v* в *j*-ые «целый» и «вещественный» вектора. Тем самым в векторах мы сформируем строки транспонированной матрицы.
  – Скопируем данные из векторов в CRS-структуру транспонированной матрицы (**Col** и **Value**), попутно формируя массив **RowIndex**.

Транспонирование матрицы

	3		7
		8	
9		15	16

A

3	7	8	9	15	16

Value

1	3	2	0	2	3

Col

0	2	3	3	6

RowIndex

3	
0	
1	3
0	3

IntVectors

9	
3	
8	15
7	16

DoubleVectors

A^T

			9
3			
	8		15
7			16

9	3	8	15	7	16

Value

3	0	1	3	0	3

Col

0	1	2	4	6

RowIndex

Умножение разреженных матриц

- Дано: $A(N \times N)$, $B(N \times N)$ в формате CRS
- Найти: C=AB в формате CRS
- Алгоритм
  - Транспонировать матрицу B, т.е. вычислить B^T.
  - Инициализировать структуру данных для матрицы C.
  - Последовательно перемножить каждую строку матрицы A на каждую из строк матрицы B^T, записывая в C полученные результаты и формируя ее структуру. При умножении строк реализовать сопоставление с целью выделения пар ненулевых элементов.

Умножение разреженных строк матриц A и B

- Простейший вариант: для каждого элемента строки матрицы A перебираем все элементы строки матрицы B^T, пока не будет найден элемент с таким же значением в массиве **Col** или не кончится строка.

```
for k:=A.RowIndex[i] to (A.RowIndex[i+1]-1)
  for l:=B.RowIndex[j] to (B.RowIndex[j+1]-1)
    if A.Col[k]=B.Col[l] then
        sum = sum + A.Value[k]* B.Value[l];
        break;
     endif
endfor
```

Умножение разреженных строк матриц A и B

- Вектора упорядочены!
- Оптимизированный вариант (аналог слияния двух отсортированных массивов в один с сохранением упорядочивания)
  - Встать на начало обоих векторов (**ks = …**, **ls = …**).
  - Сравнить текущие элементы **A.Col[ks]** и **B.Col[ls]**.
  - Если значения совпадают, просуммировать **A.Value[ks] * B. Value[ls]** и увеличить оба индекса, в противном случае – увеличить один из индексов, в зависимости от того, какое значение больше.

Умножение разреженных строк матриц A и B

```
ks:=A.RowIndex[i]; kf:=A.RowIndex[i + 1] - 1
ls:= B.RowIndex[j]; lf:= B.RowIndex[j + 1] - 1
while ((ks <= kf) && (ls <= lf))
  if A.Col[ks]<B.Col[ls] then ks:=ks+1;
  else if A.Col[ks]>B.Col[ls] then ls:=ls+1;
     else
       sum:=sum + A.Value[ks]*B.Value[ls];
        ks:=ks+1;ls:=ls+1;
     endif;
  endif
endwile
```

Разложение Холецкого

- Рассмотрим систему уравнений $Ax{=}b$
- Идея метода: $A{=}LL^T$, где L – нижняя треугольная матрица

```
for i = 1 to n
    for k = 1 to i – 1
        a_ii = a_ii – (a_ik)^2;
    a_ii = sqrt(a_ii);
    for j = i+1 to n
        for k = 1 to i – 1
            a_ji = a_ji – a_ik a_jk;
        a_ji = a_ji / a_ii;
    end
end
```

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$$

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

- Будут ли особенности в случае разреженной матрицы A?

Пример решения разреженной системы

- Пример: стреловидная матрица (5×5)
- Решение: $x^T=[2, 2, 1, 8, 0.5]$
- Фактор *L* претерпевает *заполнение* (обычно)
  - – доп. память
  - – доп. действия
    - при разложении
    - при обратном ходе
  - – плохие свойства для распараллеливания.

$$\begin{bmatrix} 4 & 1 & 2 & 0.5 & 2 \\ 1 & 0.5 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0.5 & 0 & 0 & 0.625 & 0 \\ 2 & 0 & 0 & 0 & 16 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 17 \\ 3 \\ 7 \\ 6 \\ 12 \end{bmatrix}$$

$$L = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 \\ 0.25 & -0.25 & -0.5 & 0.5 & 0 \\ 1 & -1 & -2 & -3 & 1 \end{bmatrix}$$

Пример решения разреженной системы

- Переупорядочим уравнения (перенумеруем переменные) $y_i = x_{i+1}, y_5 = x_1$
- Проведем факторизацию
- Решение: y^T=[2, 1, 8, 0.5, 2].
- Фактор разрежен в той же мере, что и исходная матрица:
  - экономия памяти
  - экономия операция
  - появляется ресурс для паралеллизма!

$$\begin{bmatrix} 0.5 & 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 0 & 2 \\ 0 & 0 & 0.625 & 0 & 0.5 \\ 0 & 0 & 0 & 16 & 2 \\ 1 & 2 & 0.5 & 2 & 4 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \\ 6 \\ 12 \\ 17 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.707 & 0 & 0 & 0 & 0 \\ 0 & 1.732 & 0 & 0 & 0 \\ 0 & 0 & 0.79 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 1.414 & 1.154 & 0.632 & 0.5 & 0.129 \end{bmatrix}$$

Решение разреженных систем

- В общем случае: домножение на матрицу перестановки
- Матрица перестановки *P*: в каждой строке и столбце матрицы находится лишь один единичный элемент

$$Ax=b \rightarrow (PAP^T)(Px)=Pb$$

- В примере

Матрица *Q* дает схожий результат

$$P=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Q=\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Решение разреженных систем

- Общая схема решения:
  - *Переупорядочивание* (поиск матрицы перестановки P);
  - *Символическое разложение* (построение портрета L);
  - *Численное разложение* (вычисление значений L);
  - *Обратный ход* (решение двух треугольных систем).
- Основная проблема – построение матрицы P.
  - задача NP-трудная;
  - используются эвристические алгоритмы
    - метод минимальной степени;
    - метод вложенных сечений.
- Рассмотрим графовую модель задачи разложения

Сведения из теории графов

- *Неориентированный граф G* можно представить в виде упорядоченной пары множеств

$$G=(V, E),\ V=\{1,...,n\},\ E=\{(e_1,e_2)\subseteq V\times V\}$$

V – множество вершин, E – множество ребер.

- Вершины x, y – смежные, если $(x,y)\in E$.
- Множество вершин, смежных данной вершине x

$$Adj(x)=\{y\in V:(x,y)\in E\}$$

- Мощность данного множества – степень вершины x

$$deg(x)=|Adj(x)|$$

Сведения из теории графов

- *Клика* – подмножество вершин, каждые две из которых соединены ребром (полный подграф исходного графа).
- Граф $G=(V, E)$ можно задать с помощью *матрицы смежности* – матрицы $A=(a_{ij})$, $n\times n$, такой, что

$$a_{ij} = \begin{cases} \neq 0, (i, j) \in E, \\ 0, \quad (i, j) \notin E. \end{cases}$$

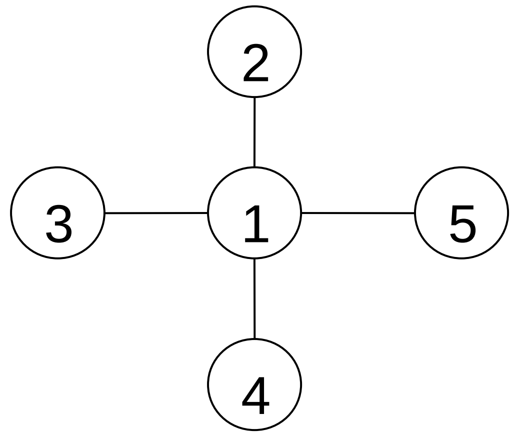
- Любую симметричную матрицу можно рассматривать как матрицу смежности неориентированного графа.

$Ax=b \rightarrow$ граф с матрицей смежности A.

Пример графового представления

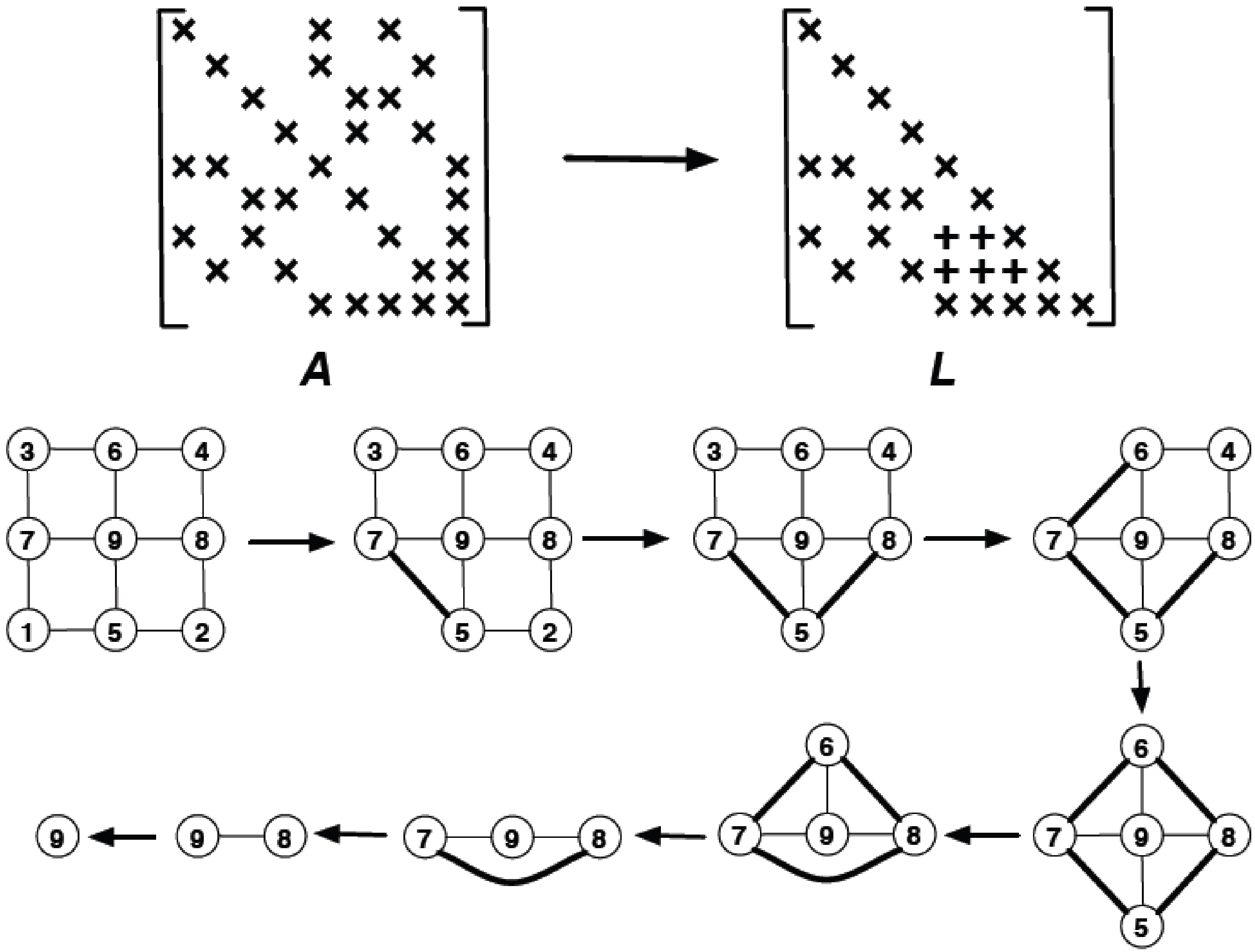
- Стреловидная матрица $\leftrightarrow$ граф с матрицей смежности

$$\begin{bmatrix} * & * & * & * & * \\ * & * & & & \\ * & & * & & \\ * & & & * & \\ * & & & & * \end{bmatrix}$$



- Из графа исключены петли, соответствующие диагональным элементам матрицы

Последовательность графов



Parallel Numerical Algorithms, Department of Computer Science
University of Illinois at Urbana-Champaign, http://www.cse.illinois.edu/courses/cs554/

Метод минимальной степени

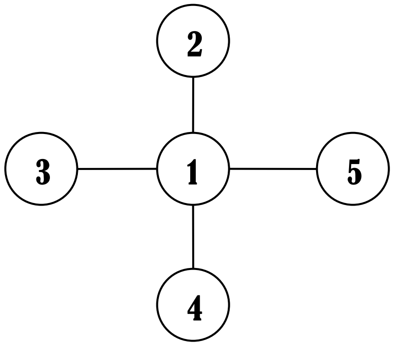
1. Инициализация – $G_0=(V,E)$, $i=1$;
2. В графе G_{i-1} выбрать узел x с минимальной степенью
3. Построить $G_i=(V_i,E_i)$, исключив из G_{i-1} узел x. Номер исключенной вершины записать в перестановку π.
4. Если $i>|V|$ то стоп

   иначе $i=i+1$, перейти на шаг 2.

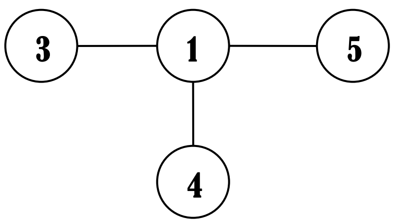
- Если вершин с минимальной степенью несколько, выбирается любая.

Метод минимальной степени

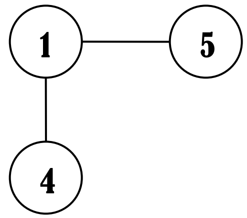
- Пример работы метода



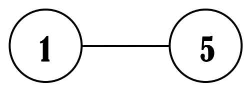
$G_0, \pi=\{\varnothing\}$



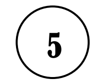
$G_1, \pi=\{2\}$



$G_2, \pi=\{2,3\}$



$G_3, \pi=\{2,3,4\}$



$G_4, \pi=\{2,3,4,1\}$

$G_5=\varnothing, \pi=\{2,3,4,1,5\}$

Метод минимальной степени

- Матрица, соответствующая $\pi=\{2,3,4,1,5\}$

$$P=\begin{bmatrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\1&0&0&0&0\\0&0&0&0&1\end{bmatrix}\qquad PAP^T=\begin{bmatrix}0.5&0&0&1&0\\0&3&0&2&0\\0&0&0.625&0.5&0\\1&2&0.5&4&2\\0&0&0&2&16\end{bmatrix}$$

Другая матрица перестановки

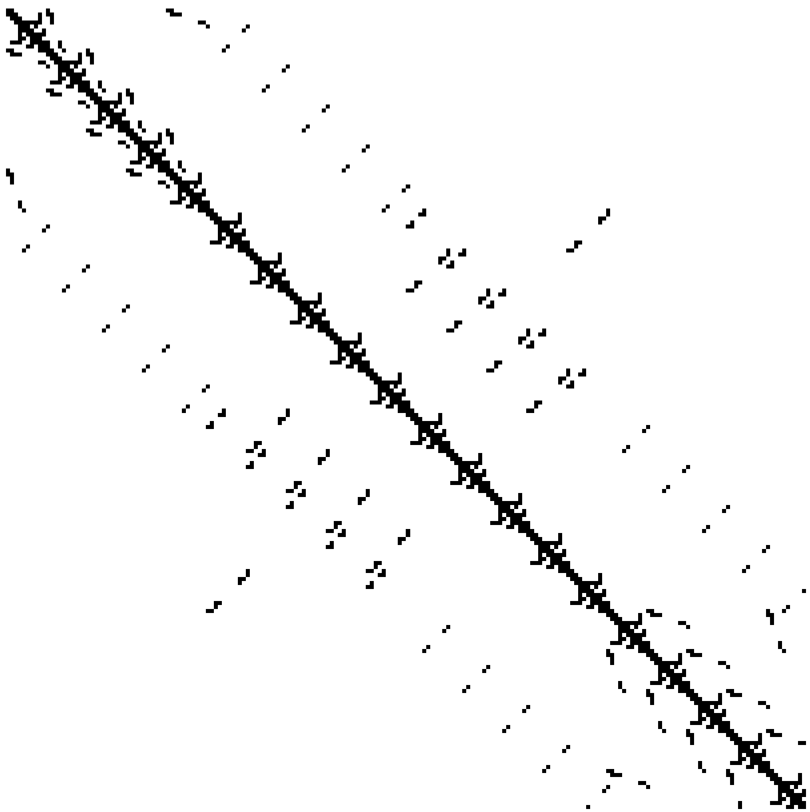
L разрежен так же, как и PAP^T

$$L=\begin{bmatrix}0.707&0&0&0&0\\0&1.732&0&0&0\\0&0&0.79&0&0\\1.414&1.154&0.632&0.516&0\\0&0&0&3.873&1\end{bmatrix}$$

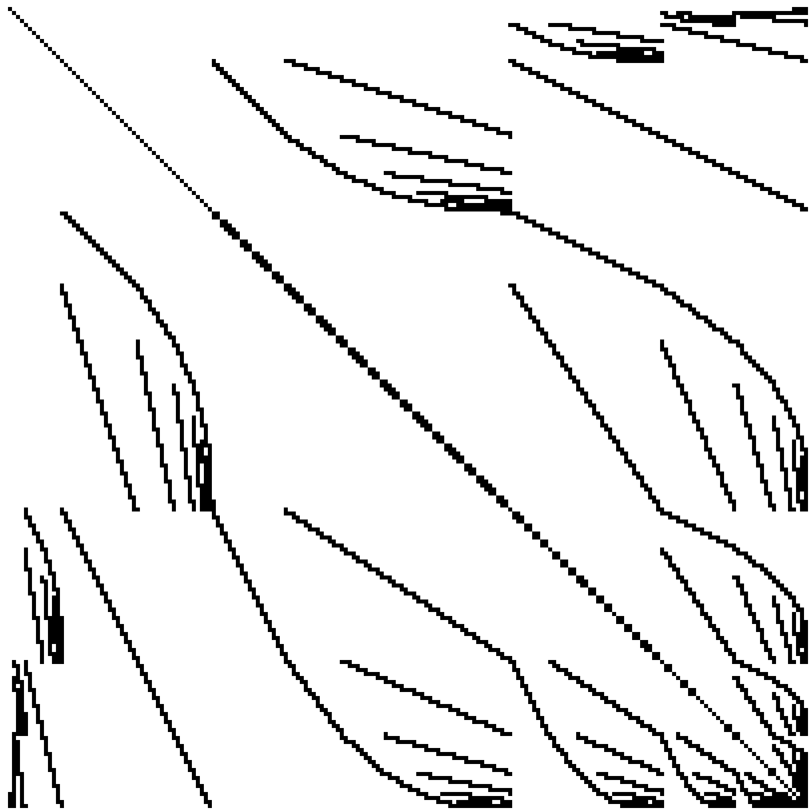
Модификации метода минимальной степени

- Метод минимальной степени в его исходном виде не используется в силу его значительной трудоемкости.
- Широко распространены две его модификации:
  – *множественный метод минимальной степени* (*Multiple Minimum Degree, MMD*, реализован в Intel MKL).
    если на некотором шаге алгоритма нашлось несколько вершин с минимальной степенью, то можно одновременно удалить все из них, не являющиеся соседями
  – *приближенный метод минимальной степени* (*Approximate Minimum Degree, AMD*)
    используется эвристическое правило: степень вершины не превосходит сумму степеней ее соседей

Тестовые матрицы

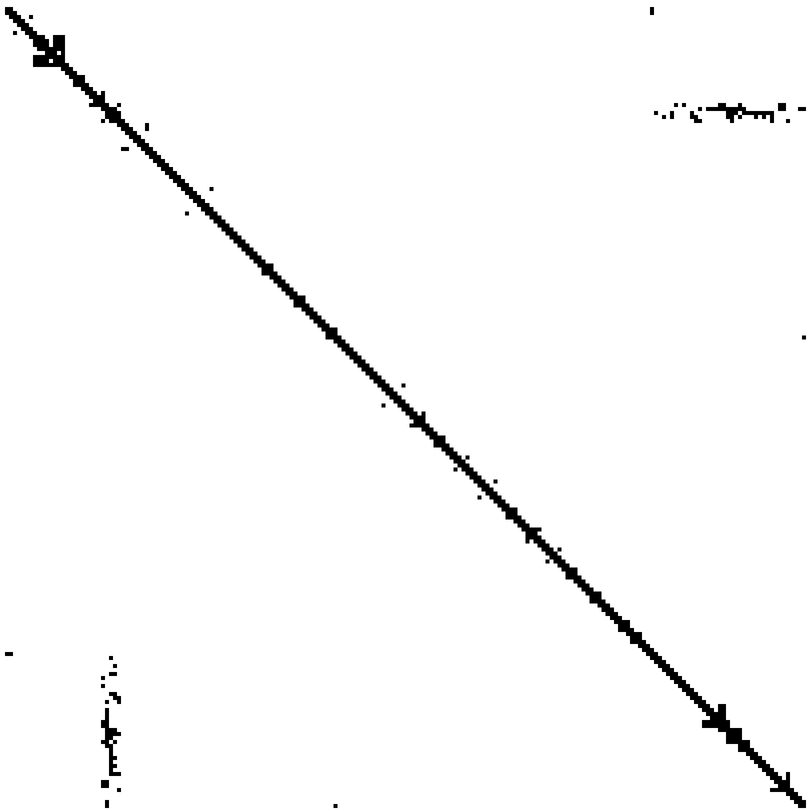


Матрица shallow_water2,
n=81 920, nz=204 800

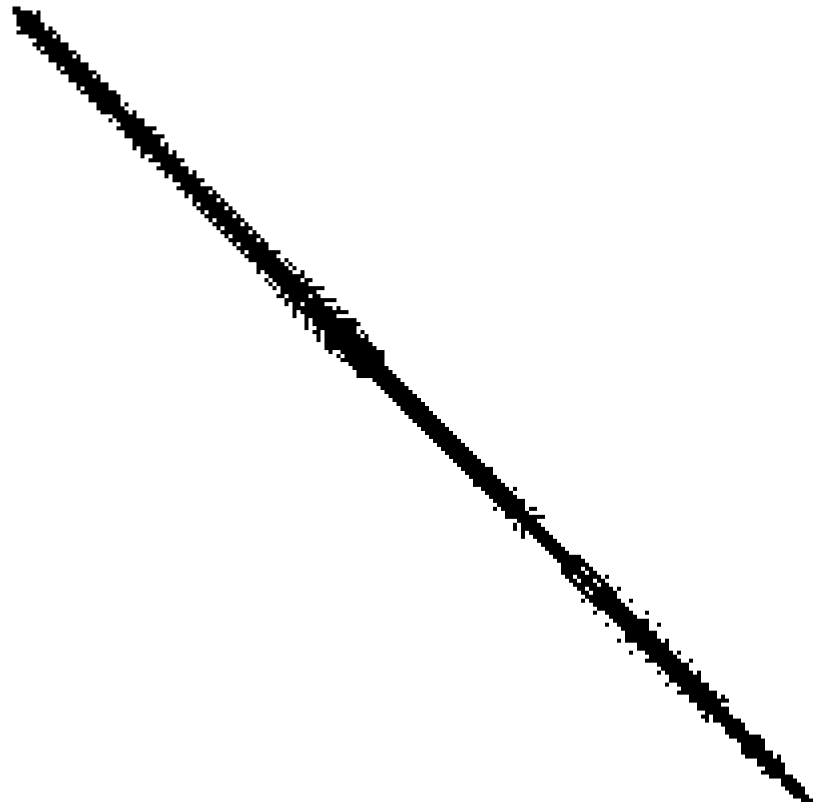


Матрица parabolic_fem,
n=525 825, nz =2 100 225

Тестовые матрицы

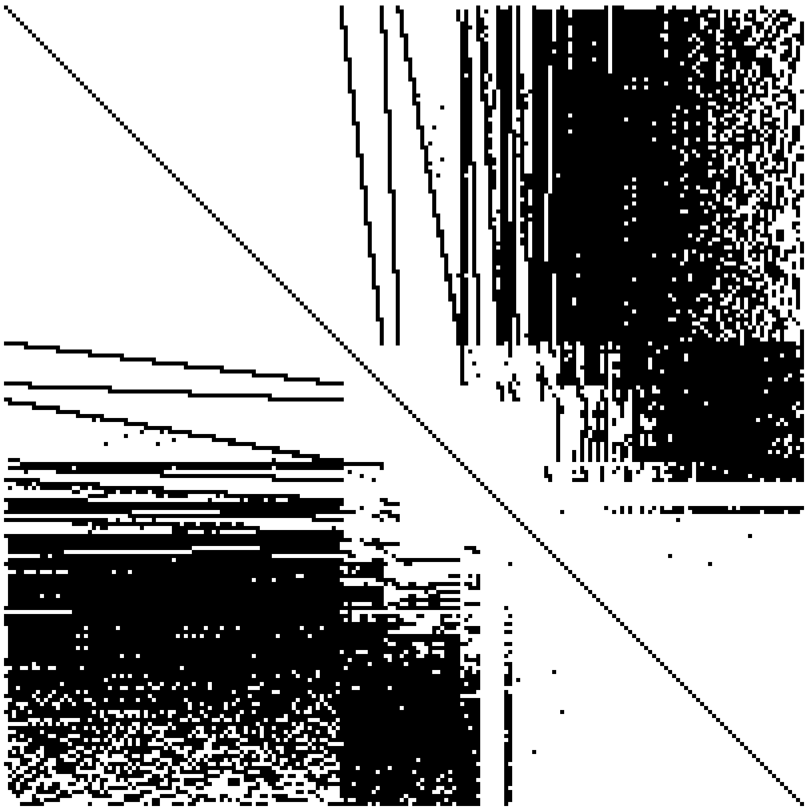


Матрица pwtk,
n=217 918, nz=5 871 175

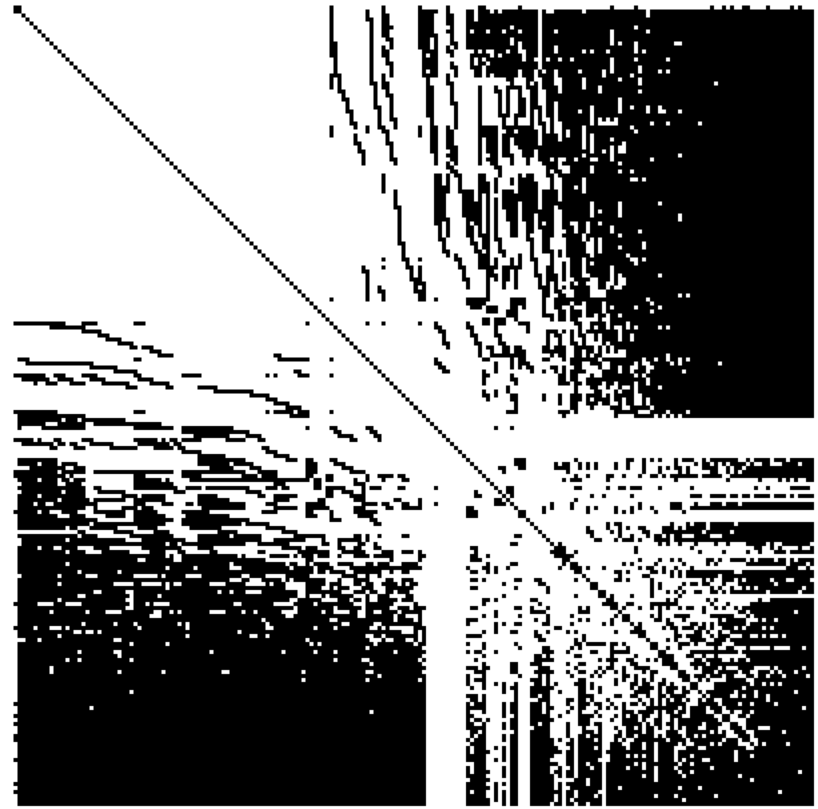


Матрица cfd2,
n=123 440, nz=1 604 423

Тестовые матрицы после MMD

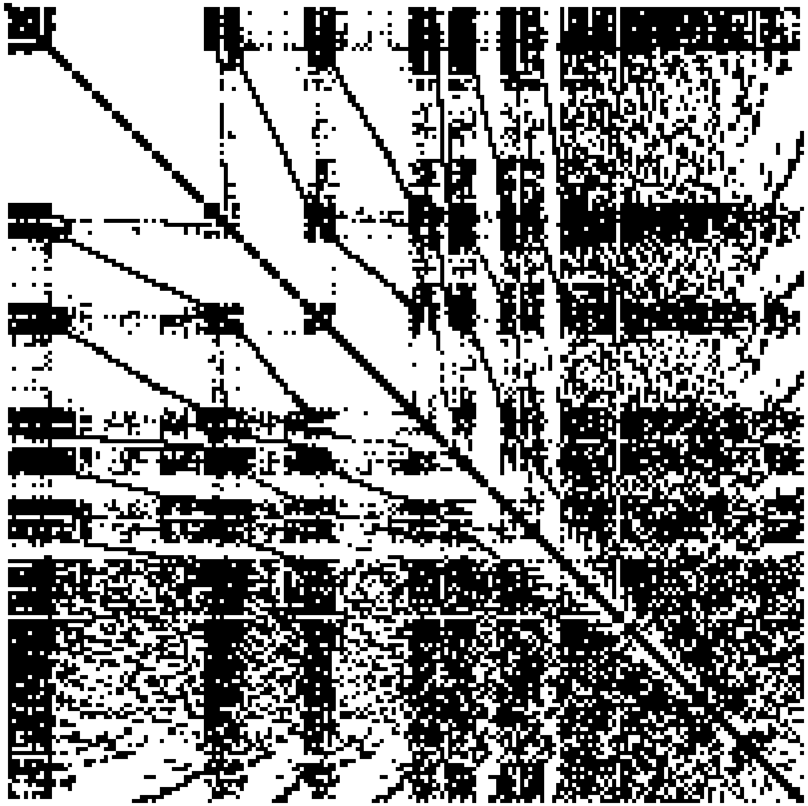


Матрица shallow_water2,
n=81 920, nz=204 800

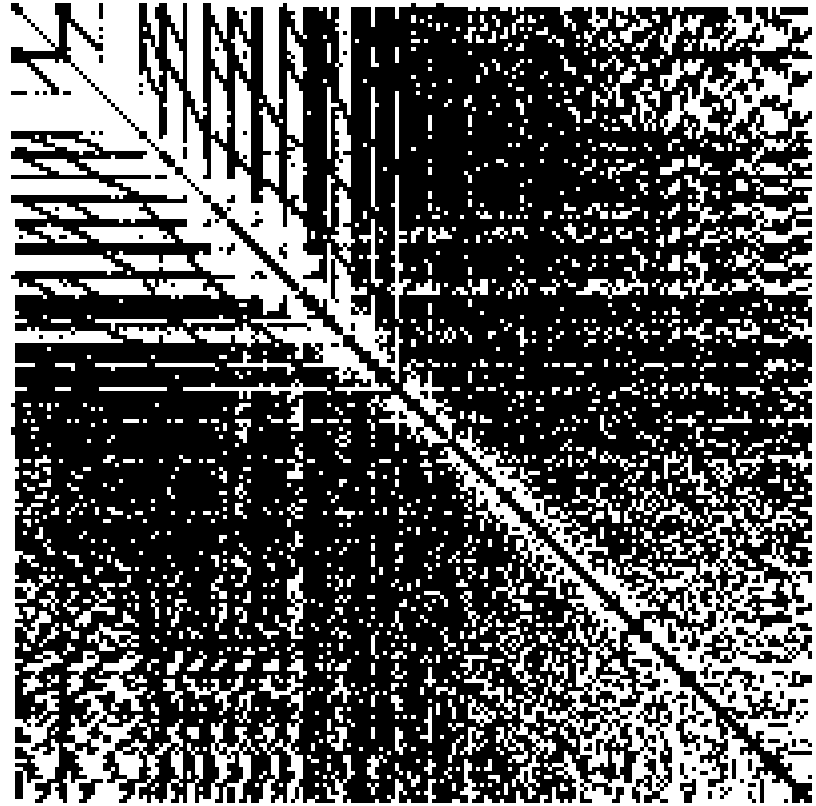


Матрица parabolic_fem,
n=525 825, nz =2 100 225

Тестовые матрицы после MMD

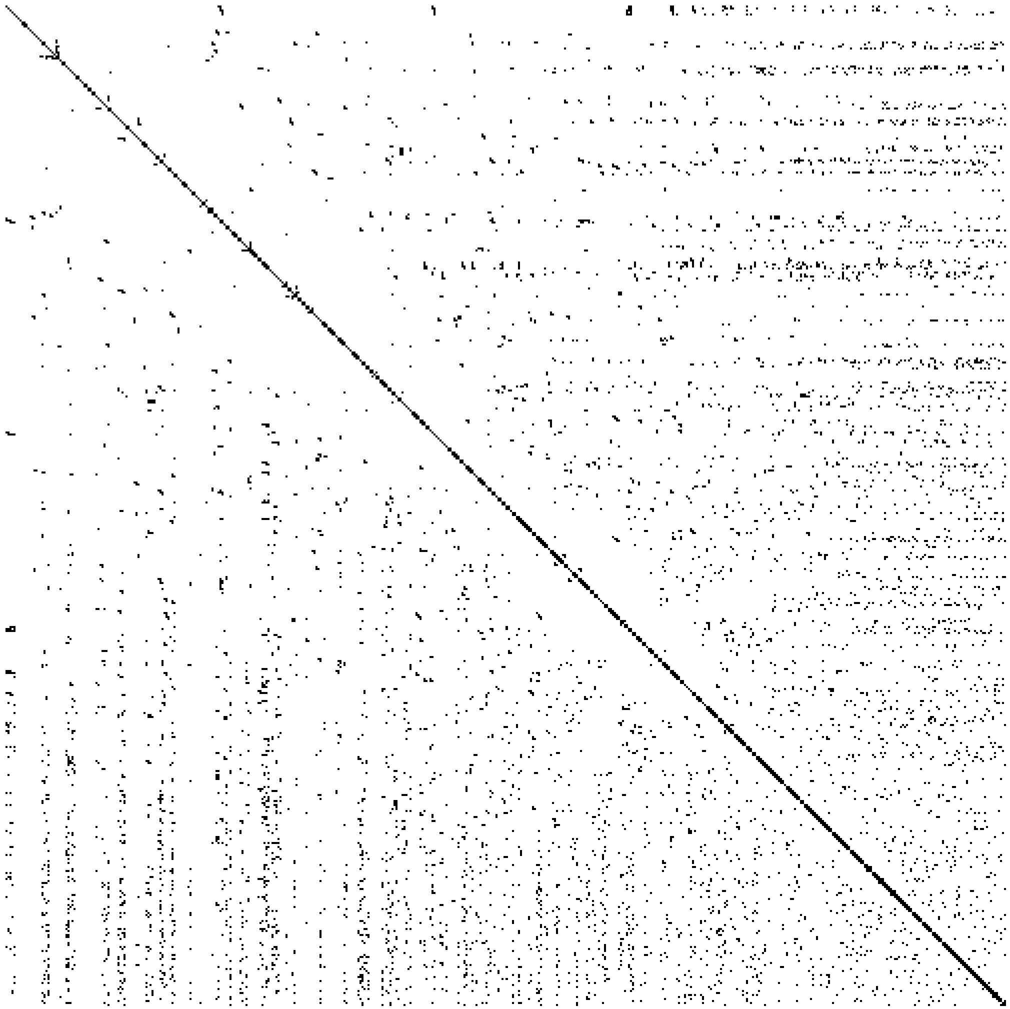


Матрица pwtk,
n=217 918, nz=5 871 175



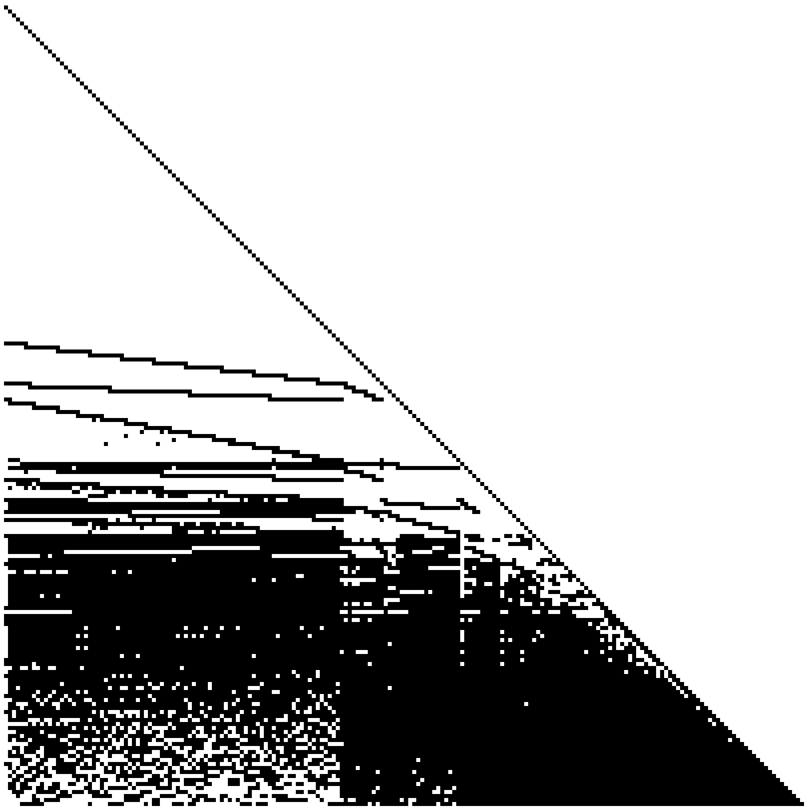
Матрица cfd2,
n=123 440, nz=1 604 423

Тестовые матрицы после MMD

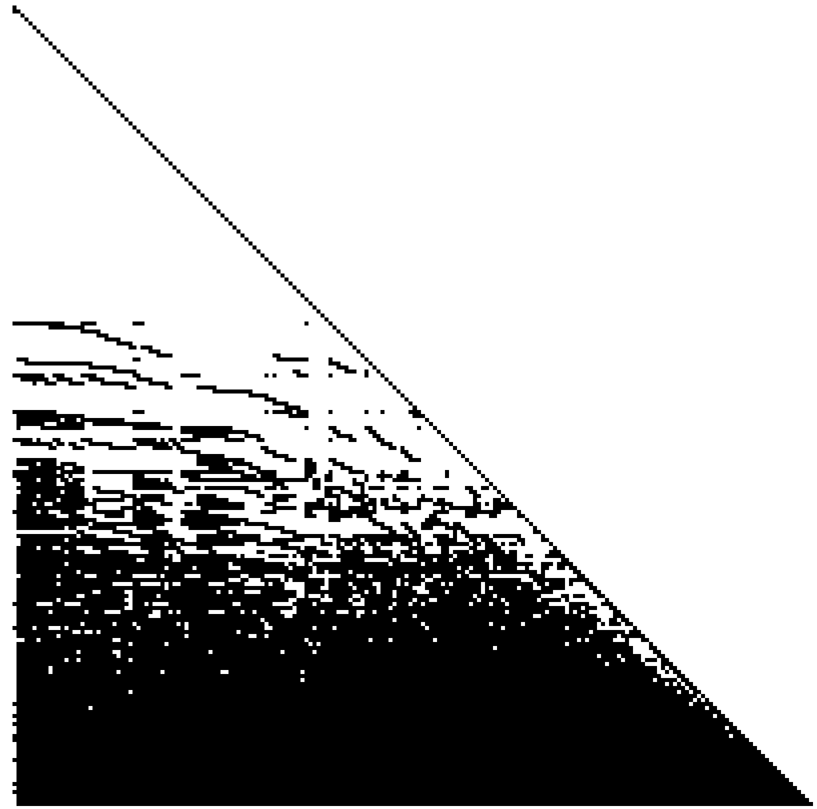


Матрица parabolic_fem, правый нижний угол

Тестовые матрицы после разложения

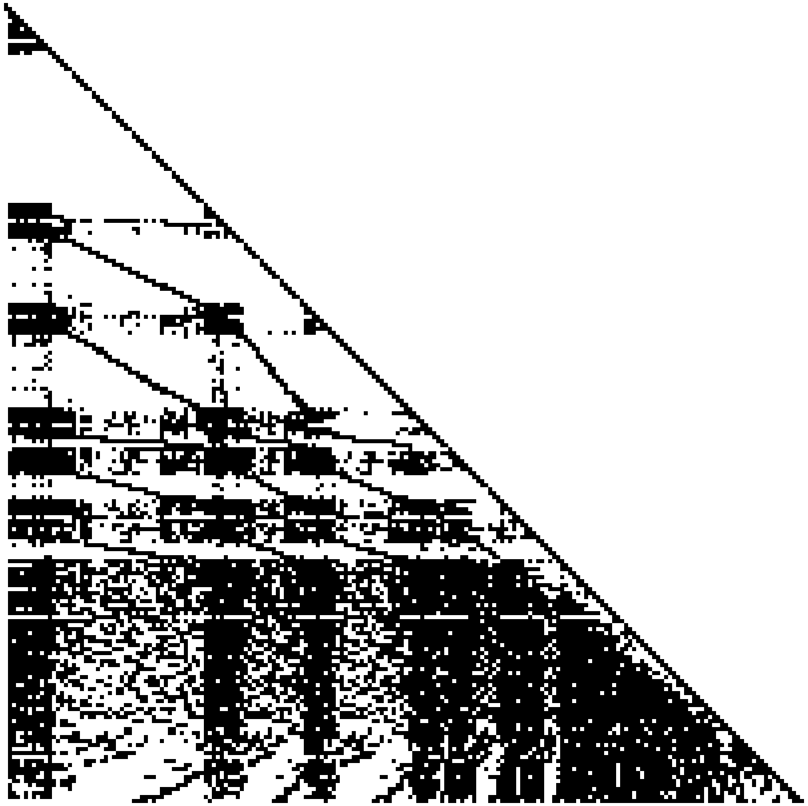


Матрица shallow_water2,
n=81 920, *nz*=2 525 184

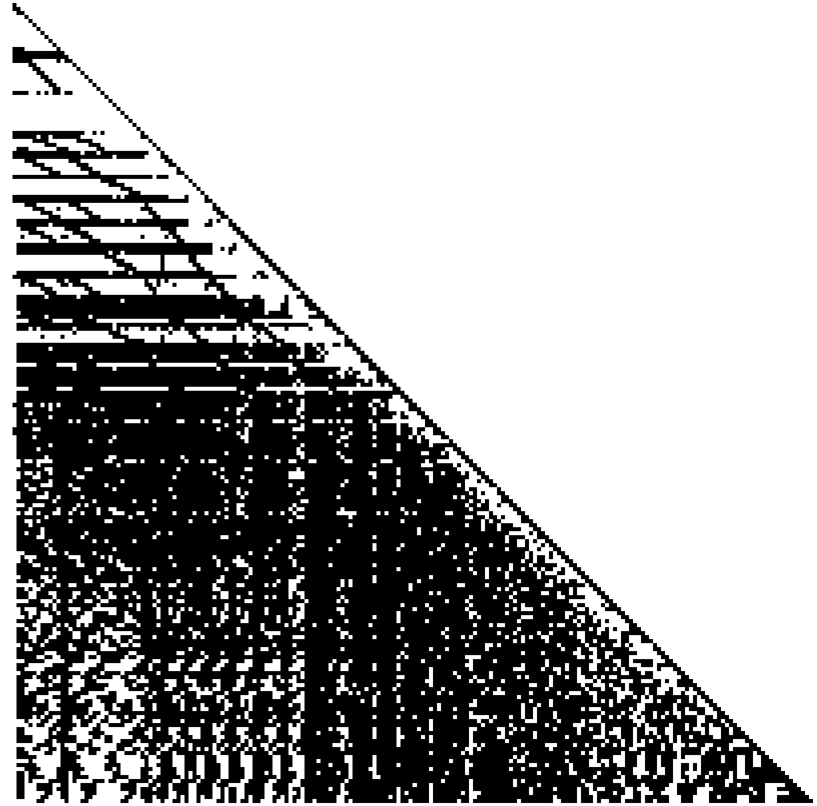


Матрица parabolic_fem,
n=525 825, *nz* =23 582 508

Тестовые матрицы после разложения

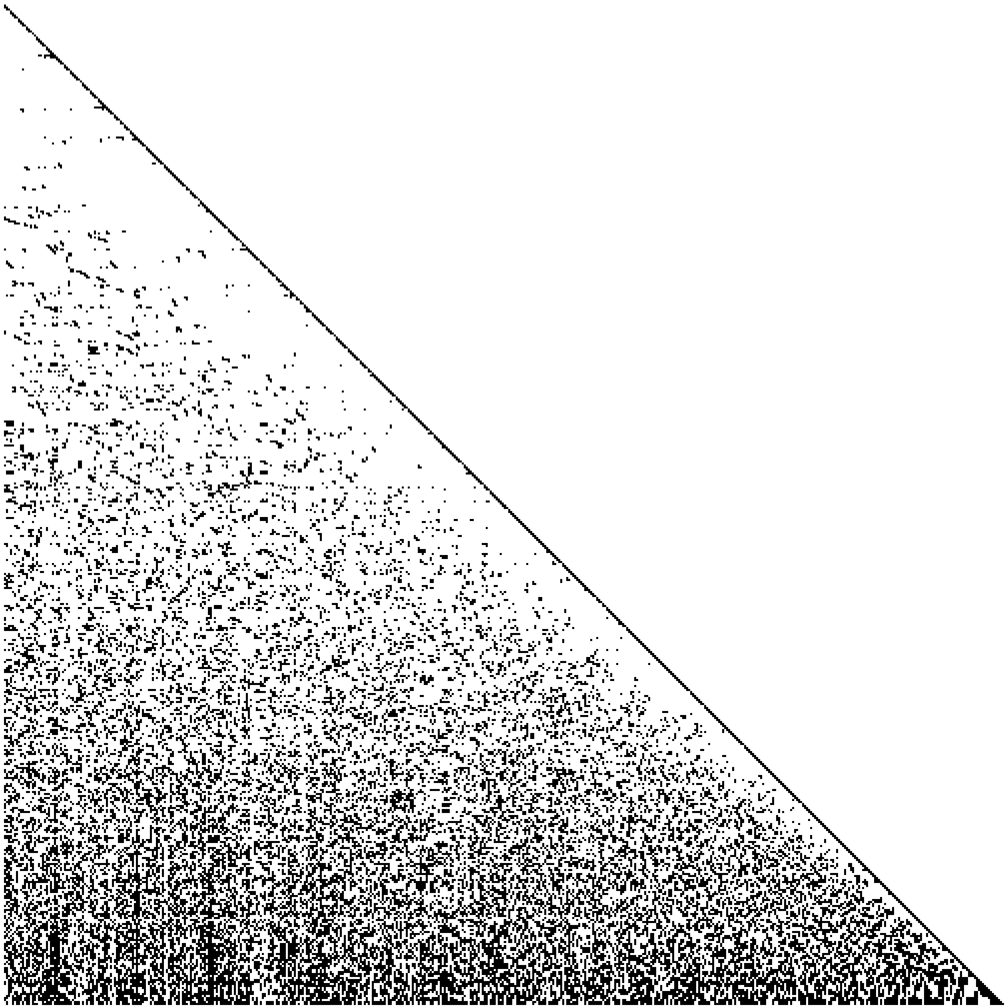


Матрица pwtk,
n=217 918, nz=60 711 075



Матрица cfd2,
n=123 440, nz=66 828 397

Тестовые матрицы после разложения



Матрица parabolic_fem
после разложения,
правый нижний угол

Сравнение коэффициентов заполнения

- Приведены числовые значения коэффициентов заполнения исходной матрицы *A*, ее факторов *L* (без переупорядочивания) и *L'* (переупорядочивание MMD).

Матрица	*A*	*L*	*L'*
shallow_water2	4,88281E-05	0,00687	0,00075
pwtk	0,00024268	0,00803	0,00256
parabolic_fem	1,32902E-05	0,16134	0,00017
cfd2	0,000202489	0,02049	0,00877

Метод вложенных сечений (nested dissection, ND)

- Граф называется *связным,* если для каждой пары его вершин найдется путь, который их связывает.
- В противном случае граф *несвязный*, и состоит из двух и более *компонент связности*.
- *Разделитель* – множество вершин, удаление которых (вместе с инцидентными им ребрами) приводит к увеличению числа связных компонент.
- *Минимальный разделитель* – никакое его собственное подмножество не является разделителем.
- *Разрезающая вершина* – разделитель, состоящий из одной вершины.

Метод вложенных сечений

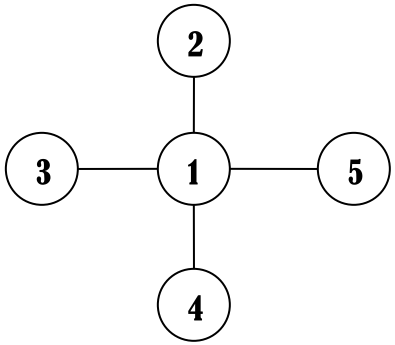
- Пусть матрице A соответствует граф $G(V,E)$

1. В графе G находим разделитель $S=\{v_k\}$
2. Удаляем S (получаем две компоненты связности G_1,G_2) номер k записываем в перестановку
3. Рекурсивно применяем алгоритм к G_1, G_2.

- Если разделитель состоит из нескольких вершин – удаляем их все, и записываем номера в перестановку.
- Как выбрать разделитель?
  - Выбирать небольшой (минимальный) разделитель;
  - G_1, G_2 должны быть примерно одинакового размера.

Метод вложенных сечений

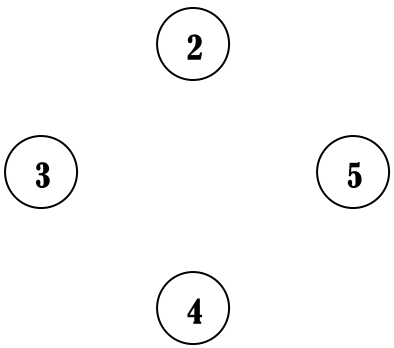
- Пример работы метода



G_1,
$\pi=\{\varnothing\}$

$\pi=\{2,3,4,5,1\}$

$$P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$



$G_2\, G_3\, G_4\, G_5$,
$\pi=\{1\}$

Уровни смежности графа

- *Разбиением* графа называется группировка вершин графа в попарно непересекающиеся подмножества S_0, S_1, … S_m.
- Важный класс разбиений – *уровни смежности*
- L_0, L_1, … L_m, - уровни смежности графа V, если

$$Adj(L_0) \subseteq L_1,$$

$$Adj(L_m) \subseteq L_{m-1},$$

$$Adj(L_i) \subseteq L_{i-1} \cup L_{i+1},\ 0<i<m.$$

где $Adj(x)$ для $x \in V$ – множеством смежных x вершин.

- Каждый уровень L_i, $0<i<m$, является разделителем графа.

Говорят о выявлении *структуры уровней смежности* (СУС)

Уровни смежности графа

- Разбиение на уровни смежности называют *корневым* с корнем L_0, если $L_0 \subset V$, а каждое следующее множество смежно с объединением предыдущих, т.е.

$$L_i = Adj\left(\bigcup_{j=0}^{i-1} L_j\right)$$

- Если L_0 состоит из единственной вершины *u*, то структура уровней имеет корень в вершине *u*.
- Число *m* называют длиной структуры уровней смежности, а *ширина* структуры определяется как максимальное количество вершин, составляющих каждый уровень

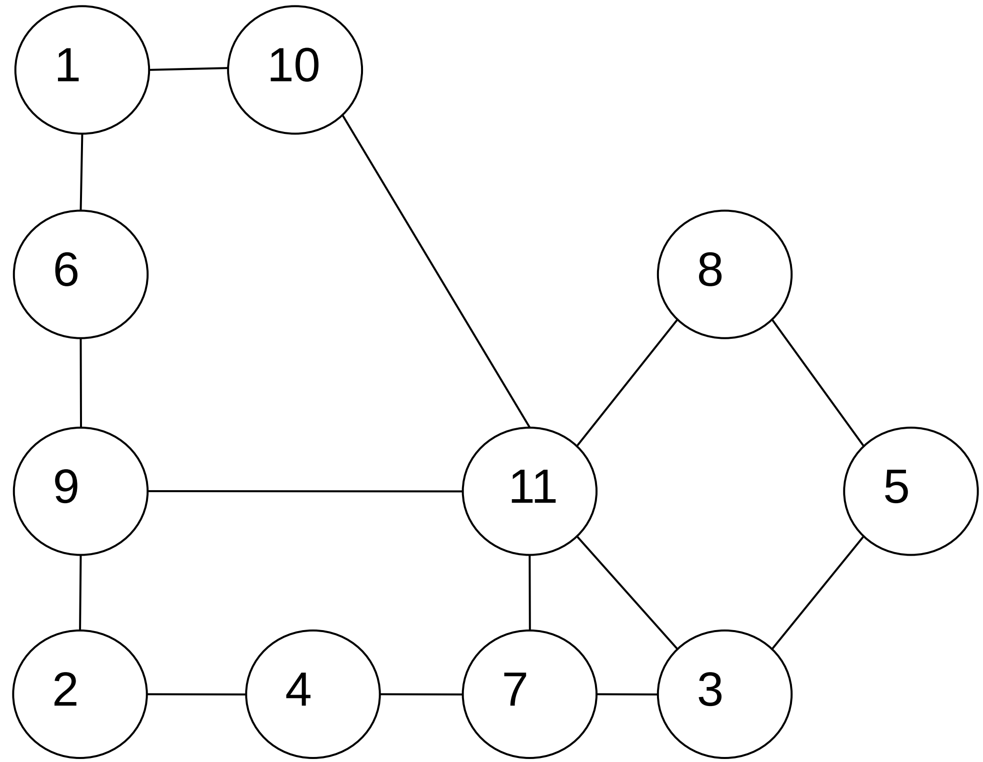
Выявление структуры уровней смежности

1. В графе V выбирается некоторая вершина $u \subset V$ в качестве корневой. Полагаем $i=0$, $L_0=\{u\}$.
2. Помечаем данную вершину (присваиваем ей первый номер)
3. Перебираем все вершины из L_i, и для каждой из них определяем непомеченную смежную ей вершину. Они помечаются (нумеруются по порядку) и помещаются в L_{i+1}.
4. Если множество непомеченных вершин пусто, то алгоритм завершен, иначе полагаем $i=i+1$, и переходим на шаг 3.

Пример структуры уровней смежности

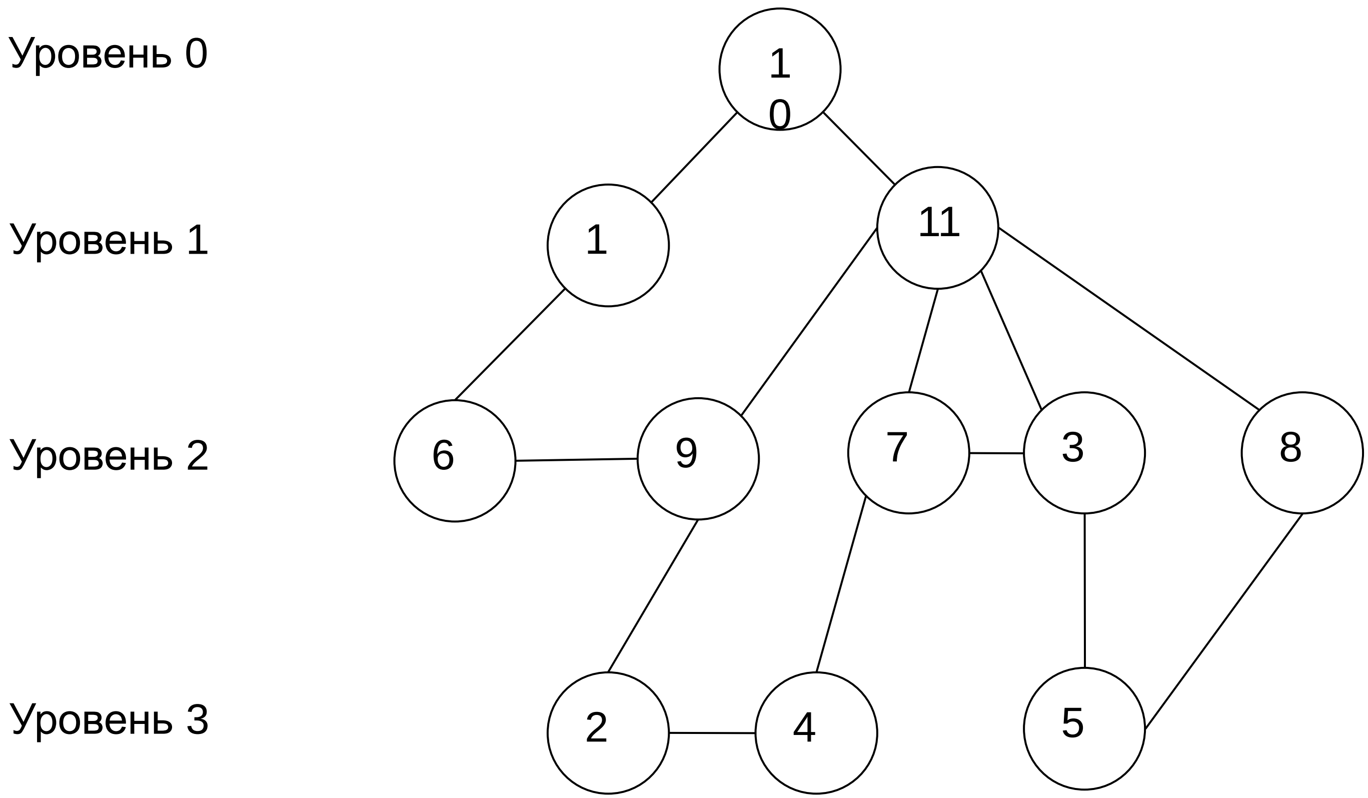
$$
\begin{bmatrix}
* & & & & & * & & & & * & \\
& * & & * & & & & & * & & \\
& & * & & * & & * & & & & * \\
& * & & * & & & * & & & & \\
& & * & & * & & & * & & & \\
* & & & & & * & & & * & & \\
& & * & * & & & * & & & & * \\
& & & & * & & & * & & & * \\
& * & & & & * & & & * & & * \\
* & & & & & & & & & * & * \\
& & * & & & & * & * & * & * & *
\end{bmatrix}
$$

Портрет матрицы



Графовое представление

Пример структуры уровней смежности



Поиск минимального разделителя

- Длина кратчайшего пути между $u,v \in V$, соединяющего эти вершины, *расстояние* $d(u,v)$,.
- $\mathrm{diam}(G)=\max\{d(u,v),\ u,v \in V\}$ – *диаметр* графа
- $e(u) = \max\{d(u,v), v \in V\}$ – *эксцентриситет вершины*
- Если $\mathrm{diam}(G)=e(u)$, то вершина u – *периферийная.*
- *Если* $d(u,v)=e(u)$, и $e(v)=e(u)$, то u – *псевдопериферийная*.

- Известный метод выбора подходящего разделителя, состоит в построении для графа структуры уровней смежности с корнем в псевдопериферийной вершине и выборе разделителя из среднего уровня.

Поиск псевдопериферийной вершины

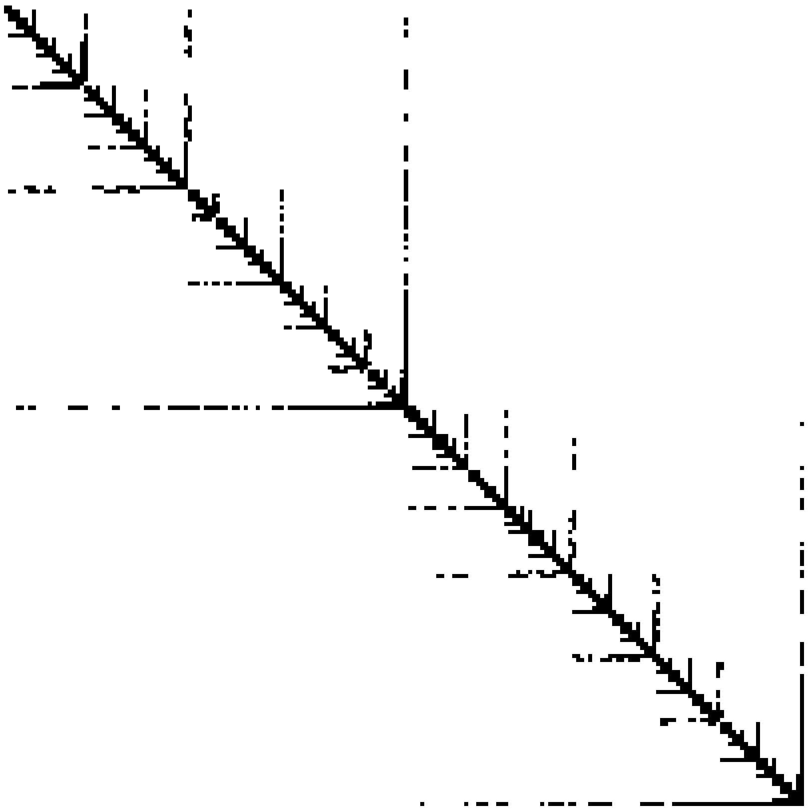
1. Выбрать некоторую вершину $r \subset V$ в качестве корневой.
2. Определить СУС $\Lambda(r)=\{L_0, L_1, \dots L_{m(r)}\}$ с корнем в r.
3. Выбрать в последнем уровне $L_{m(r)}$ вершину x с минимальной степенью.
4. Определить СУС $\Lambda(x)=\{L_0, L_1, \dots L_{m(x)}\}$ с корнем в x.
5. Если $m(x)>m(r)$, то положить $r \leftarrow x$, и перейти на шаг 3.
6. Узел x является псевдопериферийным.

Формальное описание алгоритма

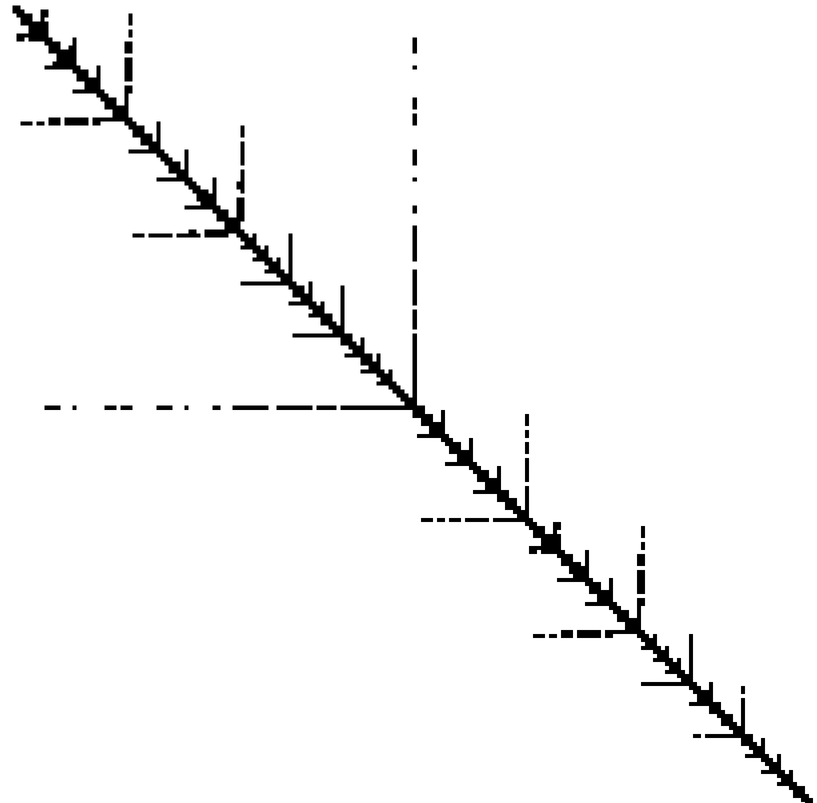
Пусть $G=G(V,E)$ – граф, ассоциированный с матрицей A.

1. Положить $R=V$, $n=|V|$.
2. Найти в $G(R)$ связную компоненту $G(C)$ и построить для нее СУС с корнем в псевдопериферийной вершине r, т.е. $\Lambda(r)=\{L_0, L_1, \dots L_m\}$.
3. Если $m\leq 2$, то положить $S=C$ и перейти к шагу 4. Иначе положить $j=(m+1)/2$ и определить разделитель $S\subset L_j$ как
$$S=\left\{y\in L_j : Adj(y)\cap L_{j+1}\neq\varnothing\right\}$$
4. Перенумеровать узлы разделителя S числами от n до $n-|S|+1$. Положить $R\leftarrow R-S$ и $n\leftarrow n-|S|$. Если $R\neq\varnothing$, то перейти на шаг 2.

Тестовые матрицы после ND

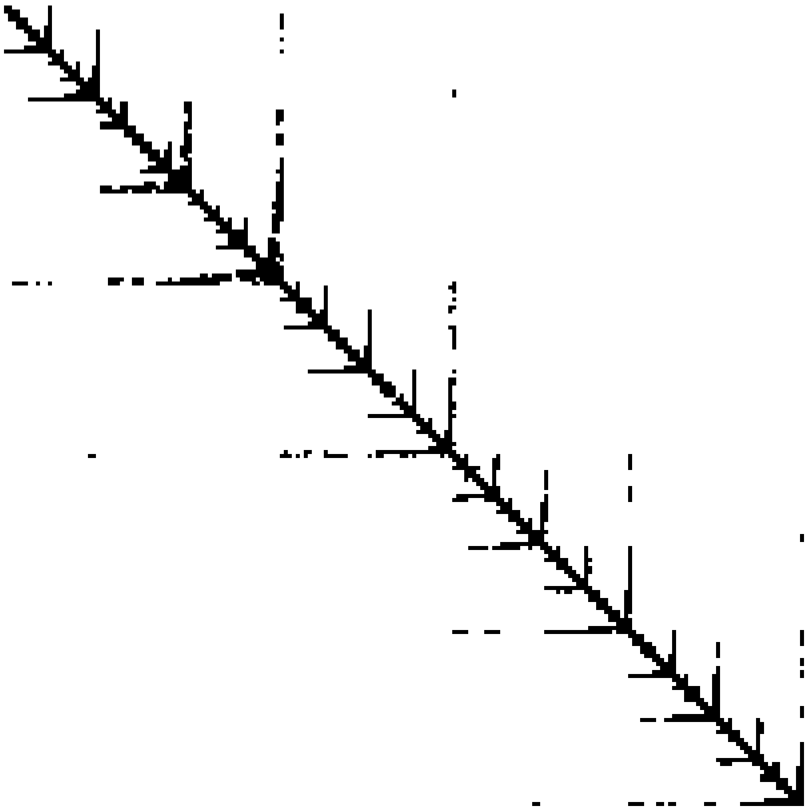


Матрица shallow_water2,
n=81 920, nz=204 800

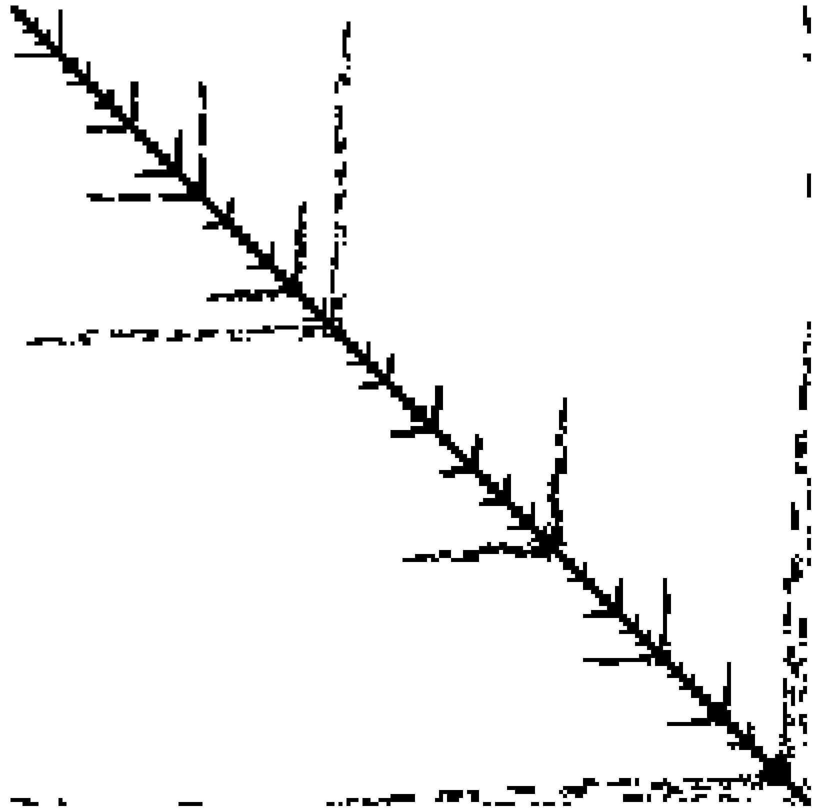


Матрица parabolic_fem,
n=525 825, nz =2 100 225

Тестовые матрицы после ND

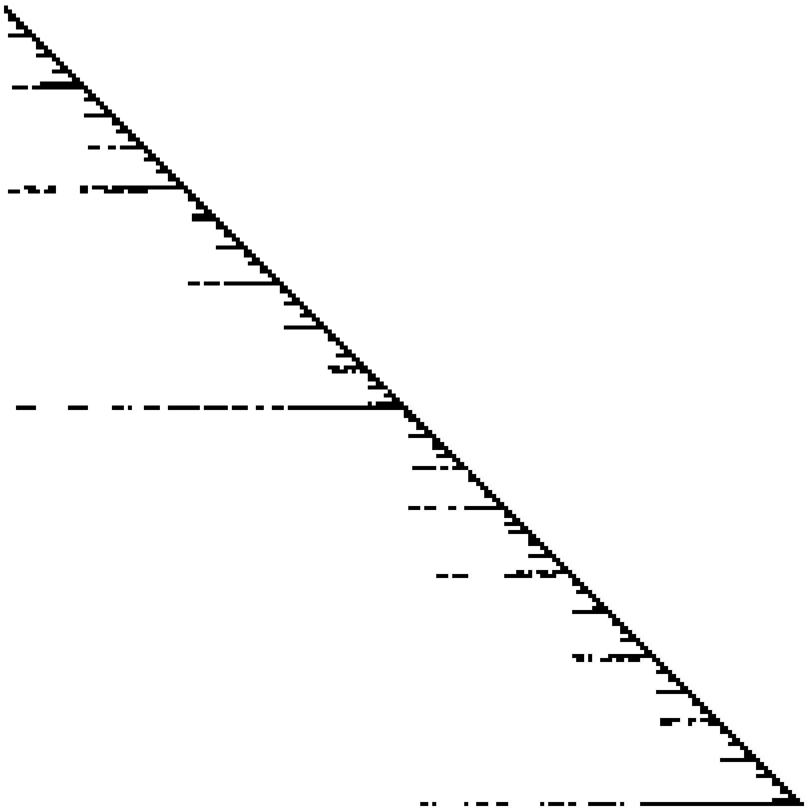


Матрица pwtk,
n=217 918, nz=5 871 175

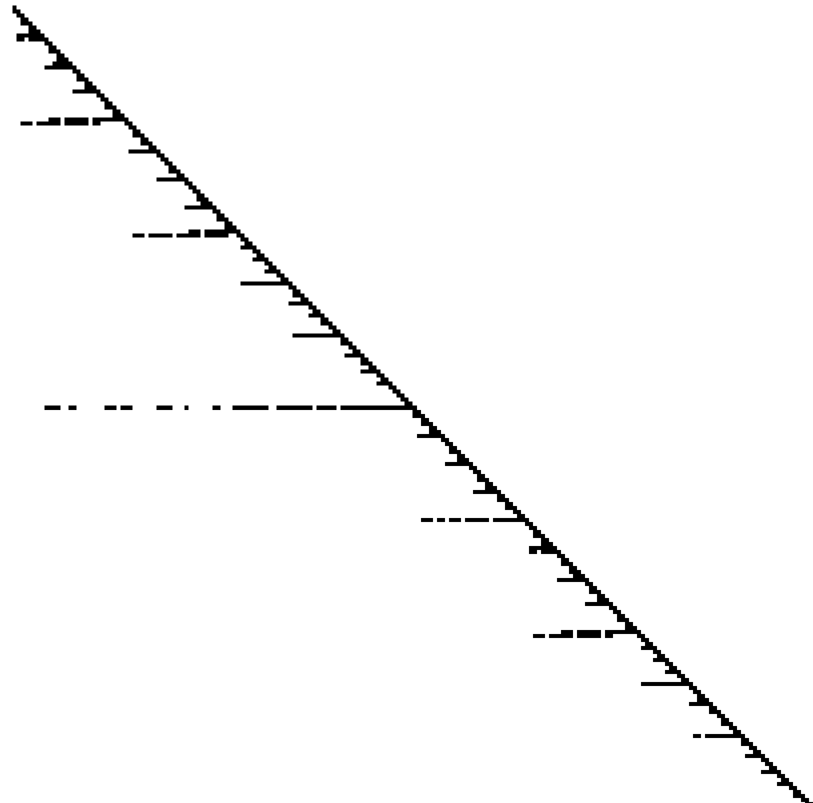


Матрица cfd2,
n=123 440, nz=1 604 423

Тестовые матрицы после разложения

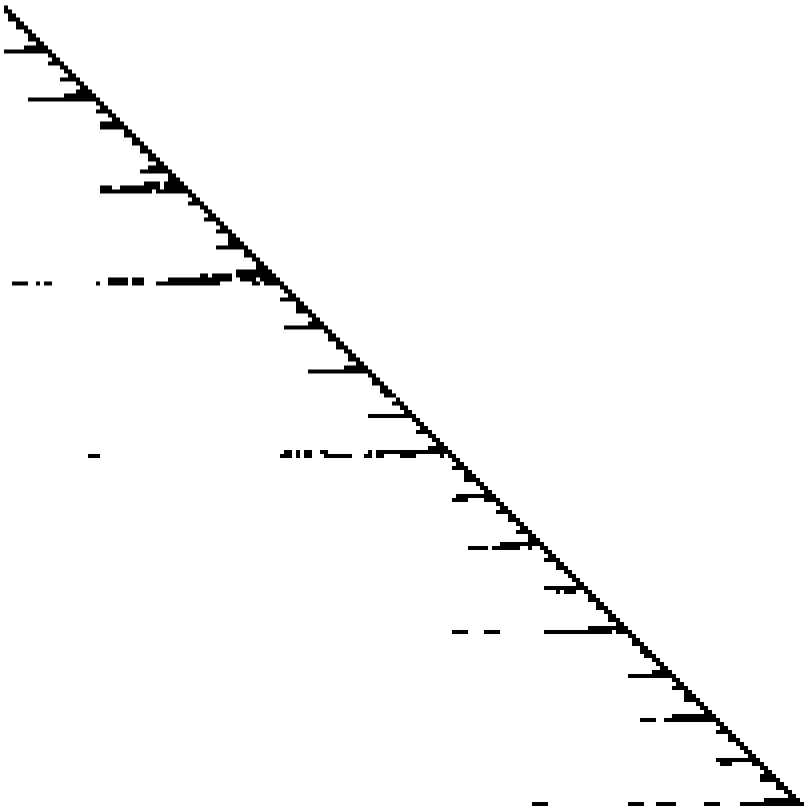


Матрица shallow_water2,
n=81 920, nz=2 183 332

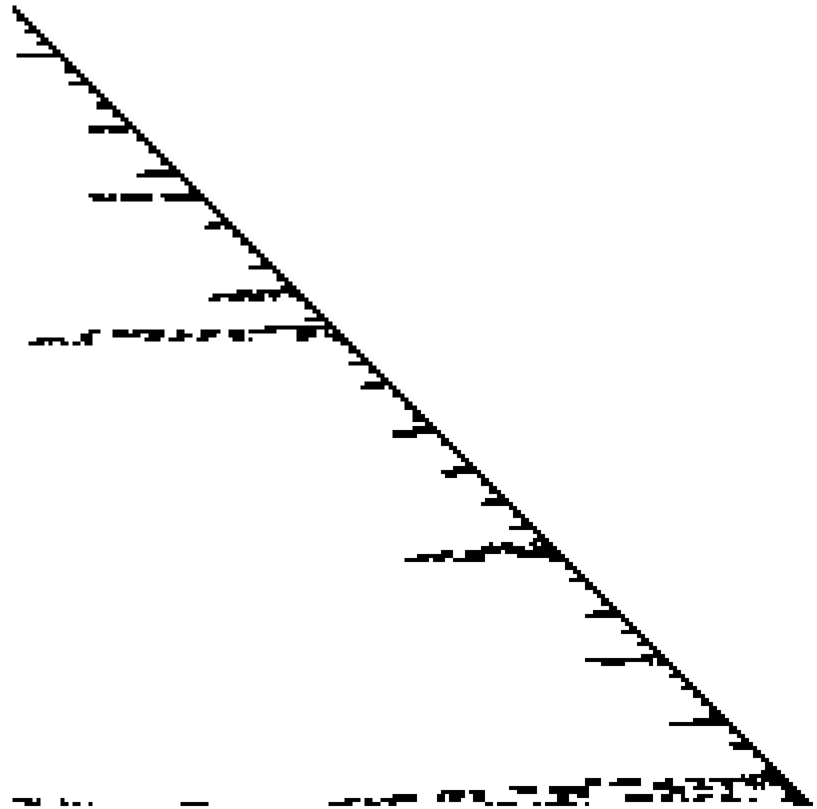


Матрица parabolic_fem,
n=525 825, nz =26 494 693

Тестовые матрицы после разложения



Матрица pwtk,
n=217 918, nz=66 843 598



Матрица cfd2,
n=123 440, nz=156 075 674

Сравнение коэффициентов заполнения

- Приведены числовые значения коэффициентов заполнения исходной матрицы *A*, ее факторов *L* (без переупорядочивания), *L'* (переупорядочивание ND) , *L*′ (переупорядочивание MMD).

Матрица	*A*	*L*	*L'*	*L''*
shallow_water2	4,88281E-05	0,00687	0,00065	0,00075
pwtk	0,00024268	0,00803	0,00281	0,00256
parabolic_fem	1,32902E-05	0,16134	0,00019	0,00017
cfd2	0,000202489	0,02049	0,02048	0,00877

Символическое разложение

- Символическое разложение – построение портрета матрицы *L*.

$$
\begin{array}{c}
\begin{bmatrix}
* & & & & & & & & & & \\
 & * & & & & & & & & & \\
 & & * & & & & & & & & \\
 & * & & * & & & & & & & \\
 & & * & & * & & & & & & \\
* & & & & & * & & & & & \\
 & & * & * & & & * & & & & \\
 & & & & * & & & * & & & \\
 & * & & & & * & & & * & & \\
* & & & & & & & & & * & \\
 & & * & & & & * & * & * & * & *
\end{bmatrix} \\
\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{matrix}
\end{array}
\qquad
\begin{array}{c}
\begin{bmatrix}
* & & & & & & & & & & \\
 & * & & & & & & & & & \\
 & & * & & & & & & & & \\
 & * & & * & & & & & & & \\
 & & * & & * & & & & & & \\
* & & & & & * & & & & & \\
 & & * & * & x & & * & & & & \\
 & & & & * & & x & * & & & \\
 & * & & x & & * & x & x & * & & \\
* & & & & & x & & & x & * & \\
 & & * & & x & & * & * & * & * & *
\end{bmatrix} \\
\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{matrix}
\end{array}
$$

- Как определить места новых ненулевых элементов?

Символическое разложение

- Элементы 7-го столбца вычисляются как

$$l_{ji} = \frac{1}{l_{ii}}\left(a_{ji} - \sum_{k=1}^{i-1} l_{ik} l_{jk}\right),\ i=7,\ j=8,9,10,11$$

- Портрет 7-го столбца получается как объединение портретов столбцов *k*=3,4,5,7 (остальные значения в строке 7 – нули)
- В общем случае для *j*-го столбца приходится выполнить слияние, вообще говоря, *j* списков.
- Известно, что надо объединять не все портреты столбцов 1, 2, …,(*j*–1), а только те, в которых ненулевой элемент в *j*-й строке является первым ненулевым ниже диагонали.
- В примере это – портреты столбцов *k*=4,5,7, портрет 5-го столбца содержит ненулевые элементы 3-го столбца

Численное разложение

- Пусть нужно обработать столбец *i* (например, *i*=7).
- Известно расположение ненулевых элементов столбца (в примере – строки *j*=7, 8, 9 и 11).
- Чтобы их вычислить их, нужно:
  – просмотреть строку *i*,
  – установить, какие столбцы имеют ненулевые элементы в этой строке (в примере – столбцы 3, 4, 5)
  – найти в этих столбцах ненулевые элементы, строчные индексы которых не меньше *i*
  – провести умножение и вычитание в соответствии с формулой

$$l_{ji} = \frac{1}{l_{ii}}\left(a_{ji} - \sum_{k=1}^{i-1} l_{ik} l_{jk}\right)$$

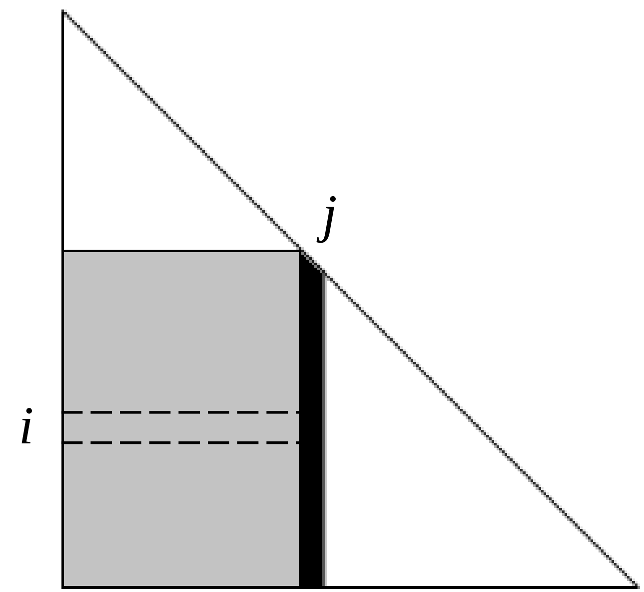
Параллелизм в разложении Холецкого

- Уровни параллелизма (на примере столбцового алгоритма)
  - Нижний: вычисление компоненты l_{ij};
  - Средний: вычисление j-го столбца;
  - Верхний: вычисление нескольких столбцов L параллельно (только для разреженных матриц!).

■ модифицируемые элементы

▒ значения, используемые для модификации

□ обращений к элементам не производится



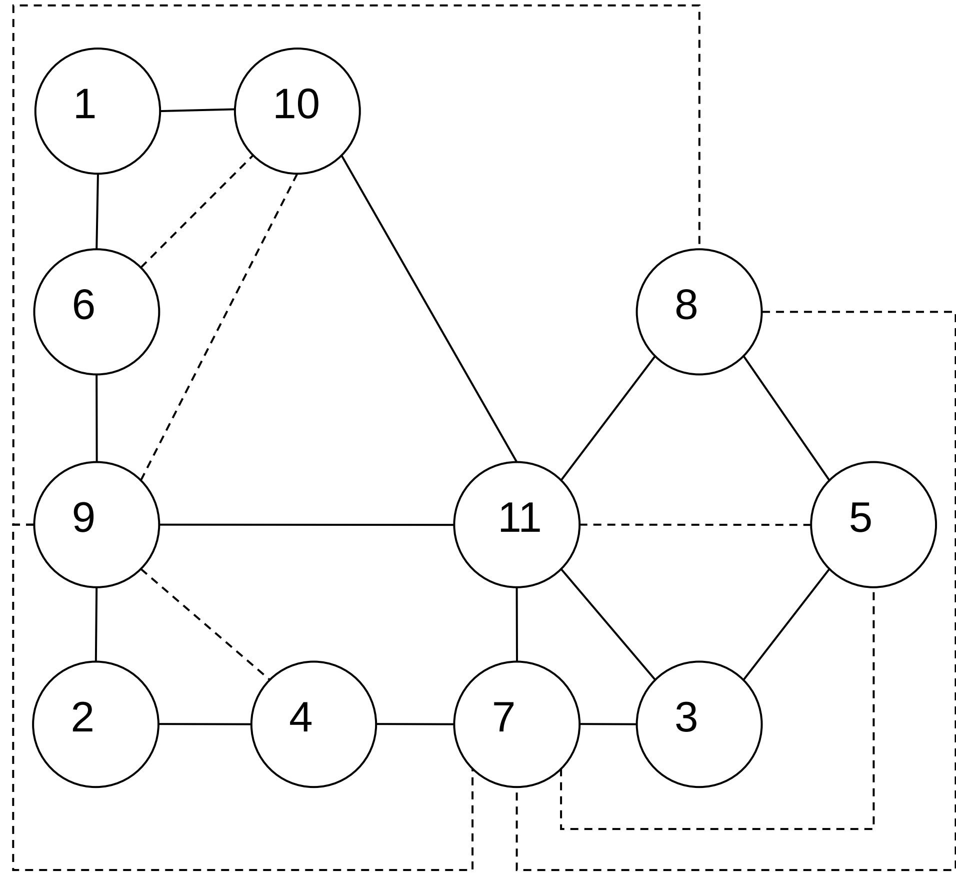
Дерево исключения

- $предок(j) = \min\{i : l_{ij} \neq 0,\ i > j\}$
  номер строки первого ненулевого элемента ниже диагонали
- *F(A)* – граф заполнения, получаемый из исходной матрицы смежности *A*, в которую добавлены появившиеся элементы из *L*.
- Дерево исключения *T*(*A*) – граф с *n* вершинами, где вершины i, j соединены дугой ($i>j$), если $i=предок(j)$
- Для графа *G*(*A*), где *A* – матрица $n \times n$, дерево исключения будет иметь корень в *n*-й вершине.
- *T(A)* является *остовным деревом* с корнем в *n*-й вершине для графа заполнения *F(A).*

Граф заполнения

$$
\begin{bmatrix}
* &&&&&&&&&& \\
& * &&&&&&&&& \\
&& * &&&&&&&& \\
& * && * &&&&&&& \\
&& * && * &&&&&& \\
* &&&&& * &&&&& \\
&& * & * & \mathrm{x} && * &&&& \\
&&&& * && \mathrm{x} & * &&& \\
& * && \mathrm{x} && * & \mathrm{x} & \mathrm{x} & * && \\
* &&&&& \mathrm{x} &&& \mathrm{x} & * & \\
&& * && \mathrm{x} && * & * & * & * & *
\end{bmatrix}
$$

Фактор *L*

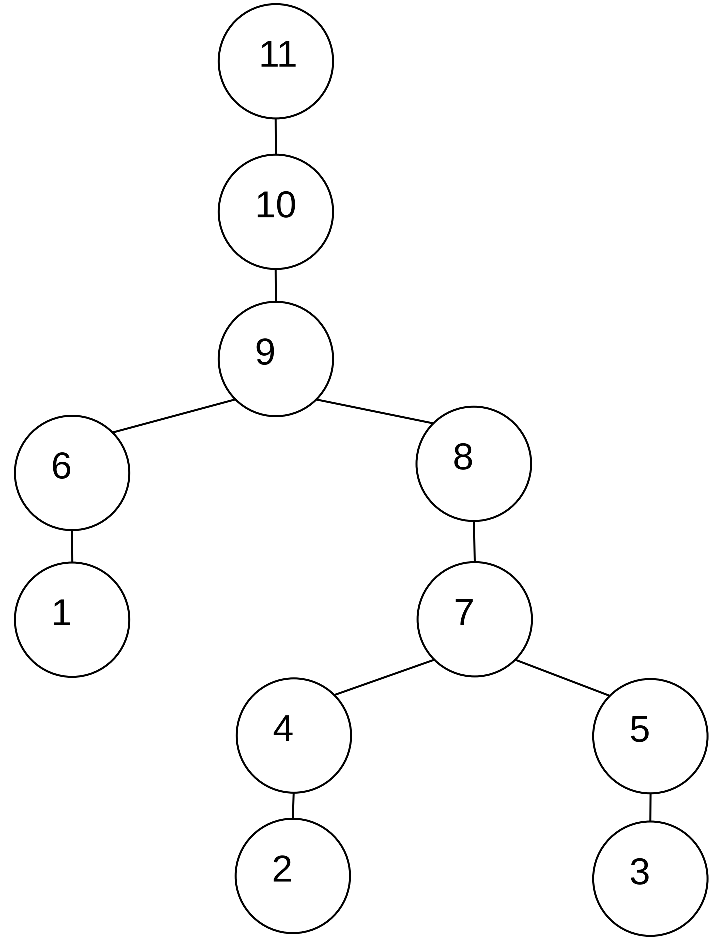


Граф заполнения

Дерево исключения

$$
\begin{bmatrix}
* & & & & & & & & & & \\
 & * & & & & & & & & & \\
 & & * & & & & & & & & \\
 & * & & * & & & & & & & \\
 & & * & & * & & & & & & \\
* & & & & & * & & & & & \\
 & & * & * & x & & * & & & & \\
 & & & & * & & x & * & & & \\
 & * & & x & & * & x & x & * & & \\
* & & & & & x & & & x & * & \\
 & & * & & x & & * & * & * & * & *
\end{bmatrix}
$$

Фактор *L*



Дерево исключения

Дерево исключения

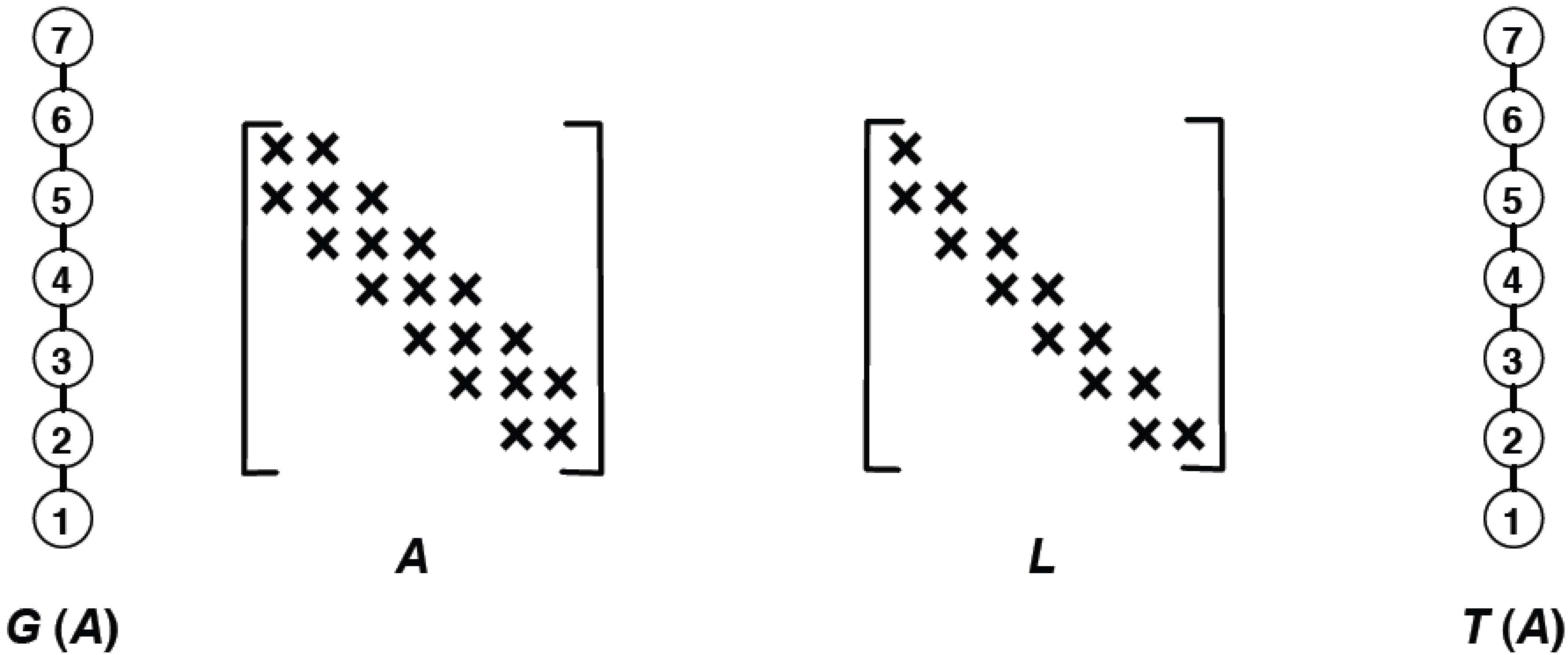
- Построение *T*(*A*) фактически происходит на этапе символического разложения.
- Дерево исключения показывает зависимость между столбцами при построении фактора *L*, тем самым определяя возможный паралеллизм задачи.
- Столбцы фактора *L*, принадлежащие непересекающимся поддеревьям *T*(*A*), могут быть вычислены независимо друг от друга, т.е. – параллельно.
- В частности, столбцы, соответствующие листьям T(A), всегда могут быть вычислены параллельно.

Дерево исключения

- *Высота* дерева исключения определяет наибольший последовательный участок вычислений
- Ширина дерева исключения определяет степень возможного паралеллизма
- Для параллельного выполнения желательно иметь невысокое, широкое, хорошо сбалансированное дерево.
- Так как структура дерева исключения зависит от переупорядочивания матрицы, алгоритм должен как сохранять разреженность, так и способствовать параллелизму.

Пример: трехдиагональная матрица

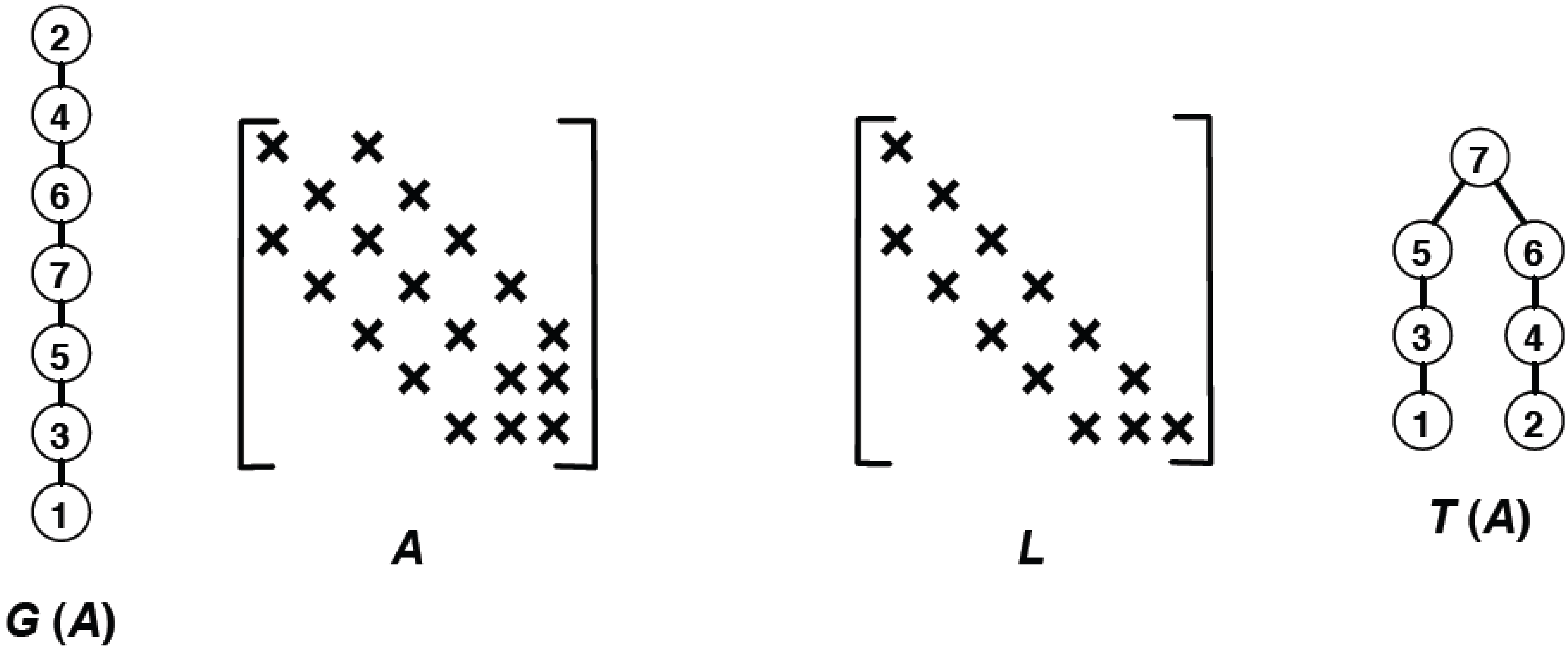
- Разложение без переупорядочивания



Parallel Numerical Algorithms, Department of Computer Science
University of Illinois at Urbana-Champaign, http://www.cse.illinois.edu/courses/cs554/

Пример: трехдиагональная матрица

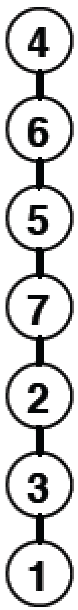
- Переупорядочивание методом минимальной степени



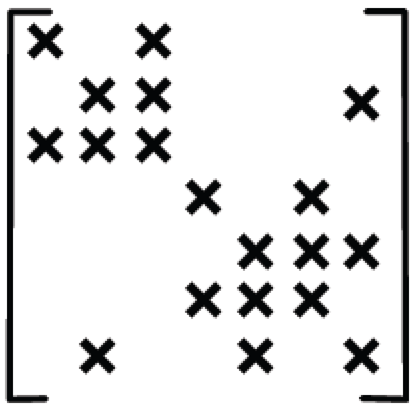
Parallel Numerical Algorithms, Department of Computer Science
University of Illinois at Urbana-Champaign, http://www.cse.illinois.edu/courses/cs554/

Пример: трехдиагональная матрица

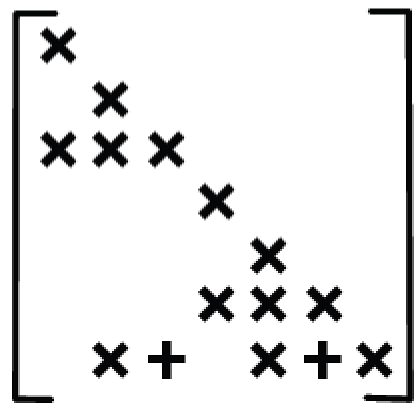
- Переупорядочивание методом вложенных сечений



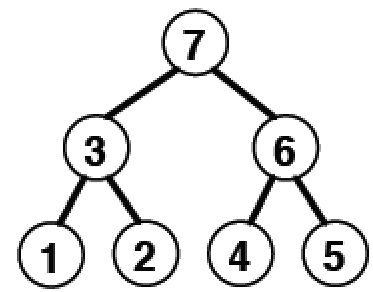
G (A)



A



L

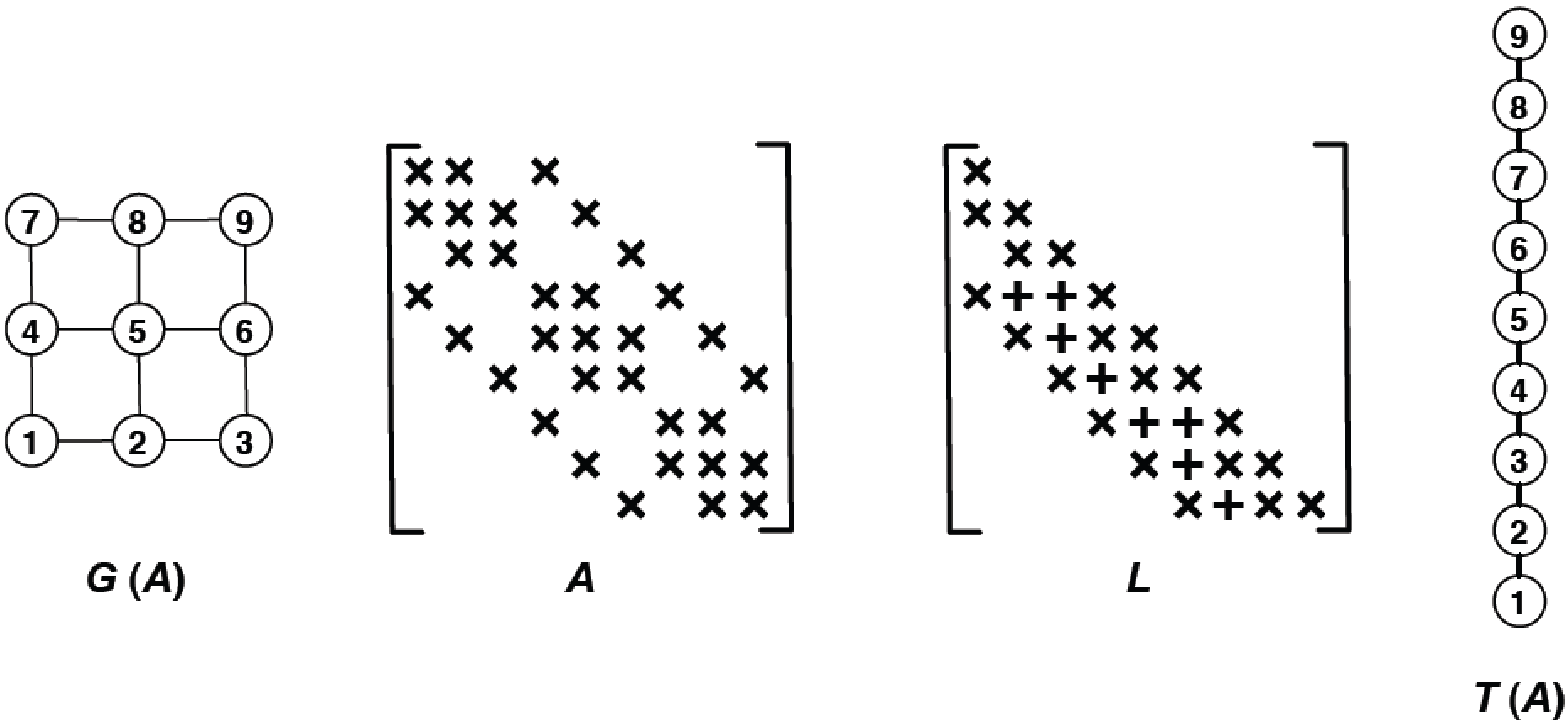


T (A)

Parallel Numerical Algorithms, Department of Computer Science
University of Illinois at Urbana-Champaign, http://www.cse.illinois.edu/courses/cs554/

Пример: пятидиагональная матрица

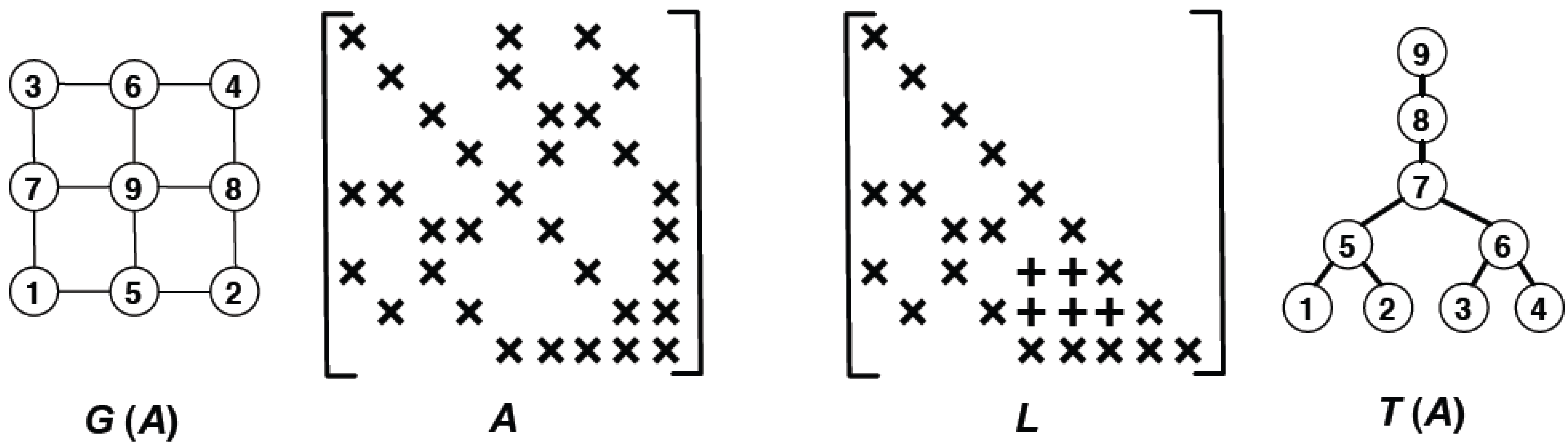
- Разложение без переупорядочивания



Parallel Numerical Algorithms, Department of Computer Science
University of Illinois at Urbana-Champaign, http://www.cse.illinois.edu/courses/cs554/

Пример: пятидиагональная матрица

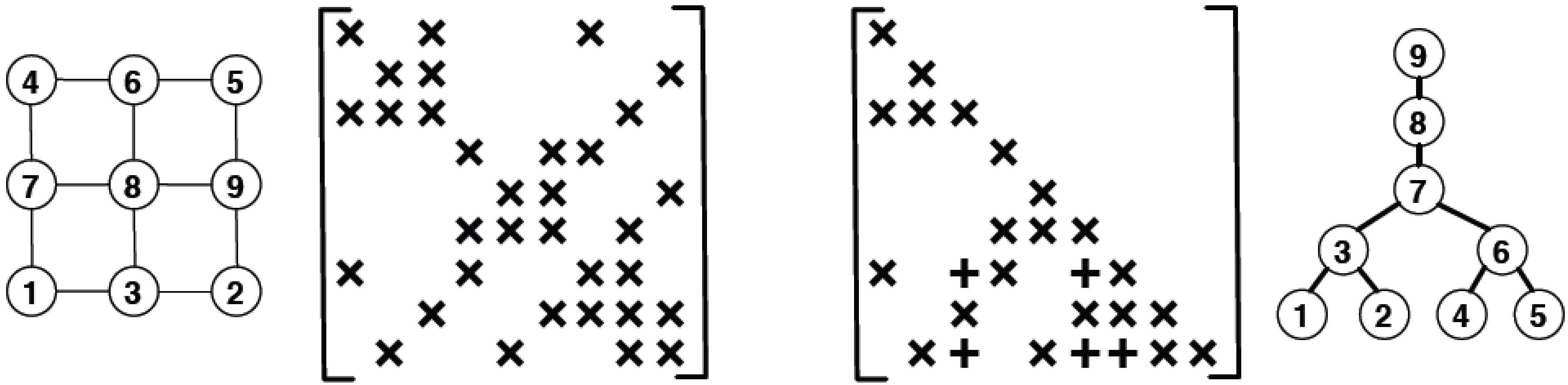
- Переупорядочивание методом минимальной степени



Parallel Numerical Algorithms, Department of Computer Science
University of Illinois at Urbana-Champaign, http://www.cse.illinois.edu/courses/cs554/

Пример: пятидиагональная матрица

- Переупорядочивание методом вложенных сечений



G (*A*) *A* *L* *T* (*A*)

Parallel Numerical Algorithms, Department of Computer Science
University of Illinois at Urbana-Champaign, http://www.cse.illinois.edu/courses/cs554/

Результаты экспериментов

- Сравним высоту дерева исключения, получающегося при переупорядочивании матриц методами минимальной степени (MD) и вложенных сечений (ND).

	ND	MD
shallow_water2	930	1459
pwtk	5823	8062
parabolic_fem	2826	4891
cfd2	6802	10723

- Переупорядочивание по методу ND дает дерево исключения с меньшей высотой, чем с использованием MD – лучшие свойства для распараллеливания

Результаты экспериментов

- Время работы последовательного алгоритма T_1 – символического разложения, T_2 – численного разложения и T_3 – обратного хода.

	T_1	T_2	T_3
shallow_water2	0,12	0,37	0,01
pwtk	4,96	53,87	0,26
parabolic_fem	1,62	7,78	0,12
cfd2	4,78	91,04	0,23

- Наибольшее время - T_2, поэтому будем распараллеливать этап численного разложения
- Распараллеливание будем проводить, выделяя непересекающиеся поддеревья в дереве исключения и параллельно обрабатывая соответствующие столбцы матрицы

Результаты экспериментов

- Время t (сек) и ускорение S параллельного численного разложения для разных матриц

Матрица	1 поток	Параллельное численное разложение							
		2 потока		4 потока		6 потоков		8 потоков	
		t	S		S	t	S	t	S
shallow_water2	0,34	0,23	1,47	0,22	1,57	0,20	1,69	0,22	1,57
pwtk	61,42	36,94	1,66	32,8	1,87	32,3	1,90	31,5	1,95
parabolic_fem	8,36	5,35	1,56	4,40	1,90	4,60	1,82	4,65	1,80
cfd2	86,05	74,9	1,15	67,4	1,28	68,8	1,25	62,6	1,38

Заключение

- Рассмотренные вопросы
  - Понятие, форматы хранения разреженной матрицы
  - Разложение Холецкого для разреженных матриц
  - Проблема заполнения фактора, переупорядочивание матрицы как метод снижения заполненности
  - Графовая модель системы уравнений
    - Последовательность графов исключения, дерево исключения
  - Переупорядочивание матрицы
    - Метод минимальной степени
    - Метод вложенных сечений
    - Ресурсы для паралеллизма
  - Примеры работы алгоритмов

Литература

1. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. – М.: Мир, 1984.
2. Писсанецки С. Технология разреженных матриц. — М.: Мир,1988.
3. Белов С.А., Золотых Н.Ю. Численные методы линейной алгебры. – Н.Новгород, Изд-во ННГУ, 2005.

Ресурсы сети Интернет

4. Intel Math Kernel Library Reference Manual. [http://software.intel.com/sites/products/documentation/hpc/mkl/mklman.pdf].

Авторский коллектив

- Баркалов Константин Александрович,
  к.ф.-м.н., старший преподаватель кафедры Математического обеспечения ЭВМ факультета ВМК ННГУ.
  barkalov@fup.unn.ru

- Коды учебных программ разработаны Маловой Анной и Сафоновой Яной