Поиск обратной матрицы с помощью блочного метода Холецкого.

Теоретическая основа метода:

В случае обычного метода Холецкого, мы ищем представление симметричной матрицы A в виде $A = R^t * D * R$, где R - верхнетреугольная положительно опеределенная матрица, а D - диагональная матрица с ± 1 на диагонали. Исходя из такого разложения не сложно получить формулы для вычисления элементов матриц D и R:

$$d_{ii} = sgn(a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2 * d_{kk})$$

$$r_{ii} = \sqrt{|a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2 * d_{kk}|}$$

$$r_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} r_{ki} * d_{kk} * r_{kj}}{r_{ii} * d_{ii}}$$

Но если мы хотим эффективно использовать ресурсы кэш памяти процессора, то нам необходимо представить матрицы A, R, D в блочном виде и оперировать с блоками, тогда в случае $A = (A_{ij})$, $R = (R_{ij})$ формулы примут следующий вид:

$$A_{ij} = \sum_{k=1}^{l} (R^t)_{ik} * D_{kk} * R_{kj} = \sum_{k=1}^{i} R_{ki}^t * D_{kk} * R_{kj}$$

Отсюда получаем формулы для пересчета R_{ii} и D_{ii}:

$$R_{ii}^t \cdot D_{ii} \cdot R_{ii} = A_{ii} - \sum_{k=1}^{i-1} R_{ki}^t * D_{kk} * R_{ki}$$

Заметим, что правая часть равенства - симметричная матрица, т.е. R_{ii} - это верхнетреугольная матрица из разложения Холецкого правой части, которое мы уже умеем вычислять не в блочном случае, а D_{ii} - диагональная матрица из разложения. После получения R_{ii} можно приступить к вычислению R_{ij}:

$$R_{ij} = D_{ii}^{-1} * R_{ii}^{-1} * (A_{ij} - \sum_{k=1}^{i-1} R_{ki}^t * D_{kk} * R_{kj})$$

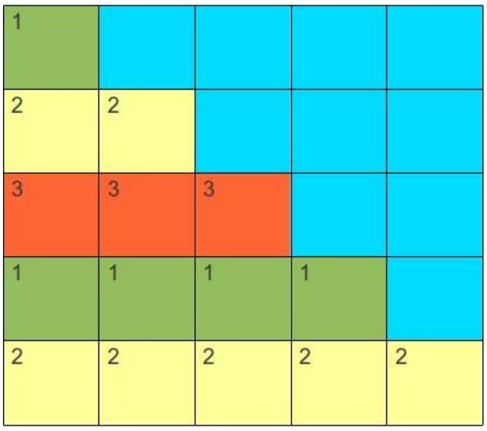
Здесь мы рассмотрели строчный вариант метода Холецкого, где R_{ij} вычисляются по строкам. Проведя расчеты по этим формулам, мы получим верхнетреугольную матрицу R, к такой матрице обратная матрица легко ищется последовательным исключением строк. Тогда искомая матрица A^{-1} будет иметь вид: $A^{-1} = R^{-1} * D^{-1} * (R^{-1})^t$.

Параллельная версия блочного метода Холецкого на MPI.

Блочный алгоритм мы рассмотрели, далее мы считаем каждый блок единичным неделимым элементом, т.е. мы получаем матрицу размером $n \cdot n$, где $n = \lceil \frac{N}{M} \rceil$, N - размер матрицы, M - размер блока. Дальше будем рассматривать только симметричные положительно определенные матрицы A, тогда A можно представить в виде $A = R \cdot R^T$, где R - нижнетреугольная положительно определенная матрица.

Разбиение матрицы по процессорам:

Процессор номер i будет хранить строки с номерами $i, i + P, i + 2 \cdot P, ...$, где P - общее количество процессоров. Для $n = 5$ и $P = 3$ разбиение примет вид:



Из разложения $A = R \cdot R^T$ ясно, что

$$r_{ji} \cdot r_{ii}^T = a_{ji} - \sum_{k=1}^{i-1} r_{jk} * r_{ik}^T \;, i < j \qquad (*)$$

Будем вычислять матрицу R по столбцам, т.е. на i-м шаге будем вычислять i-ый столбец. Причём j-ый процессор будет вычислять r_{ji}, т.е. тот элемент, который находится у него в памяти. Для того, чтобы воспользоваться формулой $(*)$ нужна ещё i-ая строка, которая будет разослана всем процессорам на i-м шаге. Совершенно аналогично поступим и при обращении матрицы R, а именно правая часть будет разбита по процессорам так же.

Вычисление произведения матриц:
Теперь вычислим $A^{-1} = R^{-T} \cdot R^{-1}$, поскольку матрица A - симметричная, то каждый элемент матрицы A - это произведение 2-х строк матрицы R^{-T}. Будем делать так - по одной строке рассылаем на все процессоры, где она умножается на все строки, которые хранятся в этом процессоре.

Количество и объём пересылок:
1) Разложение Холецкого:
Количество: $n = \frac{N}{M}$
Объём: $\sum_{i=1}^{n} i \cdot M^2 \simeq \frac{N^2}{2}$
2) Обращение матрицы R:
Количество: $n = \frac{N}{M}$
Объём: $\sum_{i=1}^{n} i \cdot M^2 \simeq \frac{N^2}{2}$
3) Перемножение матриц $R^{-1} \cdot R^{-T}$:
Количество: $2 \cdot n = 2 \cdot \frac{N}{M}$
Объём: $2 \cdot \sum_{i=1}^{n} i \cdot M^2 \simeq N^2$