Метод Гаусса решения линейной системы

Абрамов К.Е.

October 1, 2021

1. Мы решаем систему вида $Ax = b$, где A - матрица $n \times n$. Будем хранить матрицу в массиве построчно($a_{ij} = a[i * n + j]$, $i, j = 0, ..., n-1$), а вектор b в массиве b. Разобьем нашу матрицу на блоки $m \times m$, $m \times l$ и $l \times m$, $n = m * k + l$. :

$$A = \begin{pmatrix} A_{1,1}^{m\times m} & A_{1,2}^{m\times m} & --- & A_{1,k}^{m\times l} \\ A_{2,1}^{m\times m} & A_{2,2}^{m\times m} & --- & A_{1,k}^{m\times l} \\ --- & --- & --- & --- \\ A_{k-1,1}^{m\times m} & A_{k-1,2}^{m\times m} & --- & A_{k-1,k}^{m\times l} \\ A_{k,1}^{l\times m} & A_{k,2}^{l\times m} & --- & A_{k,k}^{l\times l} \end{pmatrix}.$$

2. Функции get_block и set_block:

get_block помещает блок с номером ij размера $m \times l$ в mas.

```
void get_block(int i, int j, int m, int l double* &C, double* &mas,){
 for(int q=0; i<m; q++)
  for(int p=0; i<l; p++)
   mas[q*l+p]=C[i*n*m + j*m + q*n + p];
}
```

set_block помещает блок размера $m \times l$ в позицию ij в **matrix**.

```
void set_block(int i, int j, int m, int l double* &C, double* &mas,){
 for(int q=0; i<m; q++)
  for(int p=0; i<l; p++)
   C[i*n*m + j*m + q*n +p]=mas[q*l+p]; }
```

3.Описание Алгоритма

a) Прямой ход Гаусса.

С помощью функции *get_block* копируем матрицу $A_{11}^{m\times m}$ во временный буфер C(фактически заносим ее в кэш). После этого считаем матрицу $D = C^{-1}$. Если C не обратима, то метод не применим. C^{-1} считаем обычным(не блочным) методом Гаусса. После этого умножаем первую строчку этой матрицы на D $((A_{11}^{m\times m})^{-1})$ слева.

$\boldsymbol{A_{1i} = (A_{11}^{m\times m})^{-1} * A_{1i}^{m\times m}}$, $i = 1, ..., k$

$\boldsymbol{A_{1k+1}^{m\times l} = (A_{11}^{m\times m})^{-1} * A_{1k+1}^{m\times l}}$, $i = 1, ..., k$

$E = get\_block(A_{1i})$

$E = D * E$

$set\_block(A_{1i})$

Далее мы должны обнулить первый столбец. $\boldsymbol{A_{ij} = A_{ij} - A_{i1}A_{1j}}$

После этого имеем

$$\begin{pmatrix} E^{m\times m} & A_{1,2}^{m\times m*} & --- & A_{1,k}^{m\times m*} & A_{1,k+1}^{m\times l*} \\ 0 & A_{2,2}^{m\times m} & --- & A_{2,k}^{m\times m} & A_{2,k+1}^{m\times l} \\ --- & --- & --- & --- & \\ 0 & A_{k,1}^{m\times m} & --- & A_{k,k}^{m\times m} & A_{k,k+1}^{m\times l} \\ 0 & A_{k+1,1}^{m\times m} & --- & A_{k+1,k}^{m\times m} & A_{k+1,k+1}^{m\times l} \end{pmatrix}$$

b)Обратный ход. После преобразований имеем матрицу вида:

$$\begin{pmatrix} E^{m\times m} & --- & A_{1k}^{m\times m} & A_{1,k+1}^{m\times l} \\ --- & --- & --- & --- \\ 0 & --- & E^{m\times n} & A_{k,k+1}^{m\times l} \\ 0 & --- & 0 & E_{k,k}^{l\times l} \end{pmatrix}$$

$$\boldsymbol{X_n^l = (B_n^l)}$$

$$\boldsymbol{X_i^m = (B_i^m) - \sum_{j=i+1}^{k} A_{ij}^{m\times n} X_j^m - A_{ik+1}^{m\times l} X_{k+1}^l}$$

4. Оценка числа операций.