Вариант 8

Общая постановка задачи

Написать программу, которая решает линейные задачи наименьших квадратов методом QR-разложения. Пусть дана квадратная матрица A и вектор b размерности n. Необходимо решить n задач НК вида

$$\operatorname{argmin} \|A_k x^k - b\|,$$

где A_k — матрица, составленная их первых k столбцов матрицы A,
x^k — вектор размерности k=1, 2, ..., n.

Возможны два типа матриц: обычная полная матрица или матрица в форме Хессенберга. В первом случае для построения QR-разложения следует использовать преобразования отражения, во втором — преобразования вращения. В любом случае вычисления необходимо реализовать экономно: на каждом следующем шаге (k + 1) нужно использовать имеющиеся данные с предыдущих k шагов. Матрицы отражений и вращений в памяти хранить не следует. В качестве результатов работы необходимо вывести:

1. Векторы $\left(x^k\right)^T$, k=1, 2, ..., n.
2. Евклидовы нормы невязок $r^k = \|A_k x^k - b\|$
3. Описание алгоритма с соответствующими фрагментами кода.

8

$$\begin{pmatrix} 16 & 47 & -57 & 62 & -8 & -87 & -84 & 11 & 4 & -23 \\ -4 & -90 & -97 & 94 & -20 & -64 & -75 & 48 & 92 & 69 \\ 0 & 44 & -65 & -69 & -27 & -35 & -56 & -49 & -26 & -78 \\ 0 & 0 & 83 & 70 & 18 & 88 & -35 & -32 & 82 & -57 \\ 0 & 0 & 0 & -97 & 56 & 73 & -74 & -14 & -16 & 99 \\ 0 & 0 & 0 & 0 & 71 & -52 & 36 & -6 & 77 & 50 \\ 0 & 0 & 0 & 0 & 0 & -57 & 41 & 61 & -51 & 73 \\ 0 & 0 & 0 & 0 & 0 & 0 & -42 & -18 & 79 & -56 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & -86 & 78 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12 & -82 \end{pmatrix} \begin{pmatrix} -119 \\ -47 \\ -361 \\ 217 \\ 27 \\ 176 \\ 67 \\ -37 \\ 0 \\ -70 \end{pmatrix}$$

Ответ:

x0 -6.30882 0 0 0 0 0 0 0 0 0

r0 474.781

x1 -0.875679 -1.32897 0 0 0 0 0 0 0 0

r1 460.239

x2 13.2491 -3.3852 2.96831 0 0 0 0 0 0 0

r2 209.317

x3 16.9087 -4.01295 3.09617 -0.37396 0 0 0 0 0 0

r3 205.228

x4 4.69676 -1.96339 2.13256 0.670322 2.20177 0 0 0 0 0

r4 124.567

x5 9.07807 -3.1608 3.35047 -0.0466879 1.7296 -1.0537 0 0 0 0

r5 79.6767

x6 9.61344 -2.91412 2.73113 0.169955 1.91411 -0.609143 0.348697 0 0 0

r6 77.025

x7 2.32097 -1.05351 1.6976 0.84783 2.49396 0.19912 0.428126 0.978014 0 0

r7 70.4653

x8 2.9117 -1.33072 1.89533 0.791892 2.46235 0.0300389 0.361978 0.798964 -0.0662109 0

r8 70.323

x9 1 1 1 1 1 1 1 1 1 1

r9 3.25612e-14

Исходный код

```
#include <iostream>

using namespace std;

int main()
{
	setlocale(LC_ALL, "rus");

	int n = 0;
	cout << "n: " << endl;
	cin >> n;

	long double** A = new long double* [n];
	long double** startA = new long double* [n];
	long double* b = new long double[n];
	long double* start_b = new long double[n];
```

```
	long double* x = new long double[n];
	long double* r = new long double[n];
	long double sin = 0, cos = 0, rAnswer = 0;

	for (int i = 0; i < n; i++)
	{
		A[i] = new long double[n];
		startA[i] = new long double[n];
	}
	cout << endl << "A: " << endl;
	for (int i = 0; i < n; i++)
	{
		for (int j = 0; j < n; j++)
		{
			cin >> A[i][j];
			startA[i][j] = A[i][j];
		}
	}

	cout << endl << "b: " << endl;

	for (int i = 0; i < n; i++)
	{
		cin >> b[i];
		start_b[i] = b[i];
		x[i] = 0;
		r[i] = 0;
	}

	cout << endl;

	long double num_1 = 0, num_2 = 0;

	for (int i = 0; i < n - 1; i++)
```

```
	{
		sin = A[i + 1][i] / (sqrt(A[i][i] * A[i][i] + A[i + 1][i] * A[i + 1][i]));
		cos = A[i][i] / (sqrt(A[i][i] * A[i][i] + A[i + 1][i] * A[i + 1][i]));
		for (int j = i; j < n; j++)
		{
			num_1 = cos * A[i][j] + sin * A[i + 1][j];
			num_2 = cos * A[i + 1][j] - sin * A[i][j];
			A[i][j] = num_1;
			A[i + 1][j] = num_2;
		}
		A[i + 1][i] = 0;
		num_1 = cos * b[i] + sin * b[i + 1];
		num_2 = cos * b[i + 1] - sin * b[i];
		b[i] = num_1;
		b[i + 1] = num_2;
	}

	num_1 = 0;

	for (int k = 0; k < n; k++)
	{
		for (int i = k; i >= 0; i--)
		{
			if (A[i][i] == 0)
			{
				x[i] = 0;
			}
			else {
				num_1 = b[i];
				for (int j = 0; j <= k; j++)
				{
					num_1 -= A[i][j] * x[j];
				}
```

```
                        x[i] = num_1 / A[i][i];
                  }
            }
            for (int i = 0; i < n; i++)
            {
                  for (int j = 0; j <= k; j++)
                  {
                        r[i] += startA[i][j]*x[j];
                  }
                  r[i] -= start_b[i];
                  rAnswer += r[i] * r[i];
            }
            cout << "x" << k << " ";
            for (int j = 0; j < n; j++)
            {
                  cout << x[j] << " ";
                  x[j] = 0;
                  r[j] = 0;
            }
            cout << endl;
            cout << "r" << k << " " << sqrt(rAnswer) << endl;
            rAnswer = 0;
      }
}
```

Описание алгоритма

В начале находим sin и cos по формулам sin = β/√(α^2+β^2), cos = α /√(α^2+β^2), далее зануляем столбцы в матрице A до верхнетреугольной. Фрагмент кода:

```
            sin = A[i + 1][i] / (sqrt(A[i][i] * A[i][i] + A[i + 1][i] *
A[i + 1][i]));
            cos = A[i][i] / (sqrt(A[i][i] * A[i][i] + A[i + 1][i] * A[i
+ 1][i]));
            for (int j = i; j < n; j++)
            {
                  num_1 = cos * A[i][j] + sin * A[i + 1][j];
                  num_2 = cos * A[i + 1][j] - sin * A[i][j];
```

```
                A[i][j] = num_1;
                A[i + 1][j] = num_2;
        …
```

Также изменяем и вектор b:

```
num_1 = cos * b[i] + sin * b[i + 1];
        num_2 = cos * b[i + 1] - sin * b[i];
        b[i] = num_1;
        b[i + 1] = num_2;
```

Далее высчитываем x, для каждого шага и находим r.

```
for (int k = 0; k < n; k++)
    {
        for (int i = k; i >= 0; i--)
        {
            if (A[i][i] == 0)
            {
                x[i] = 0;
            }
            else {
                num_1 = b[i];
                for (int j = 0; j <= k; j++)
                {
                    num_1 -= A[i][j] * x[j];
                }
                x[i] = num_1 / A[i][i];
            }
        }
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j <= k; j++)
            {
                r[i] += startA[i][j]*x[j];
            }
            r[i] -= start_b[i];
            rAnswer += r[i] * r[i];
```

```
            }
            cout << "x" << k << " ";
            for (int j = 0; j < n; j++)
            {
                  cout << x[j] << " ";
                  x[j] = 0;
                  r[j] = 0;
            }
            cout << endl;
            cout << "r" << k << " " << sqrt(rAnswer) << endl;
            rAnswer = 0;
      }
```