Доскоч Роман Дмитриевич

Лабораторная работа №2

Вариант 2

Постановка задачи

Написать программу, которая решает линейные задачи наименьших квадратов методом QR-разложения. Пусть дана квадратная матрица A в форме Хессенберга и вектор b размерности n. В качестве результатов работы необходимо вывести:

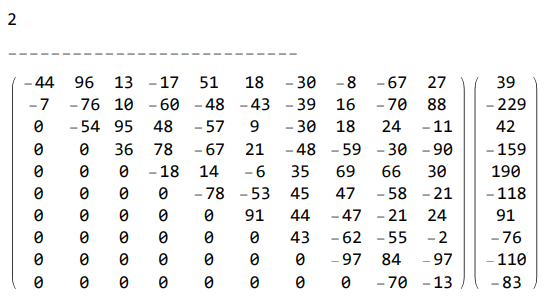
1. Векторы (xk)T , k=1, 2, ..., n.

2. Евклидовы нормы невязок rk = ||Akxk – b||2

3. Описание алгоритма с соответствующими фрагментами кода.

Решение

Исходная матрица в форме Хессенберга



Ответ:

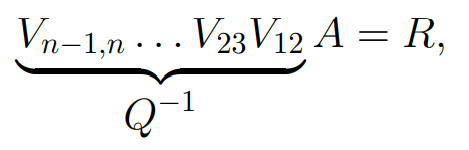
(x1)T  = (-0,05693 \_ \_ \_ \_ \_ \_ \_ \_ \_ ) (x2)T = ( 3,08814 1,69094 \_ \_ \_ \_ \_ \_ \_ \_ ) (x3)T = ( 4,21878 2,14835 0,86536 \_ \_ \_ \_ \_ \_ \_ ) (x4)T = (19,08687 7,07698 7,62130 -5,97548 \_ \_ \_ \_ \_ \_ ) (x5)T = (15,44385 5,11570 6,52222 -4,33614 1,37841 \_ \_ \_ \_ \_ ) (x6)T = (16,78165 5,68421 6,83612 -5,34412 0,68276 1,02855 \_ \_ \_ \_ ) (x7)T = (23,92819 8,34018 8,65618 -8,01952 -0,38399 1,59206 -1,15844 \_ \_ \_ ) (x8)T = ( 7,28166 2,22210 3,17785 -1,48240 1,52574 1,50200 0,45647 1,37476 \_ \_ ) (x9)T = (-5,45500 -1,70790 -0,66140 2,38846 2,05741 1,27831 1,63315 1,72270 0,79505 \_ ) (x10)T = (1,00000 1,00000 1,00000 1,00000 1,00000 1,00000 1,00000 1,00000 1,00000 1,00000 )

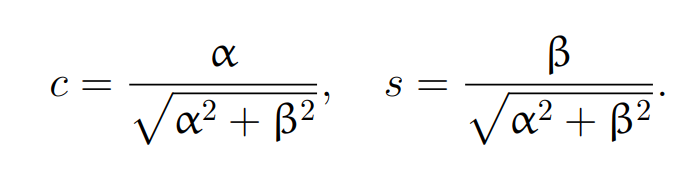
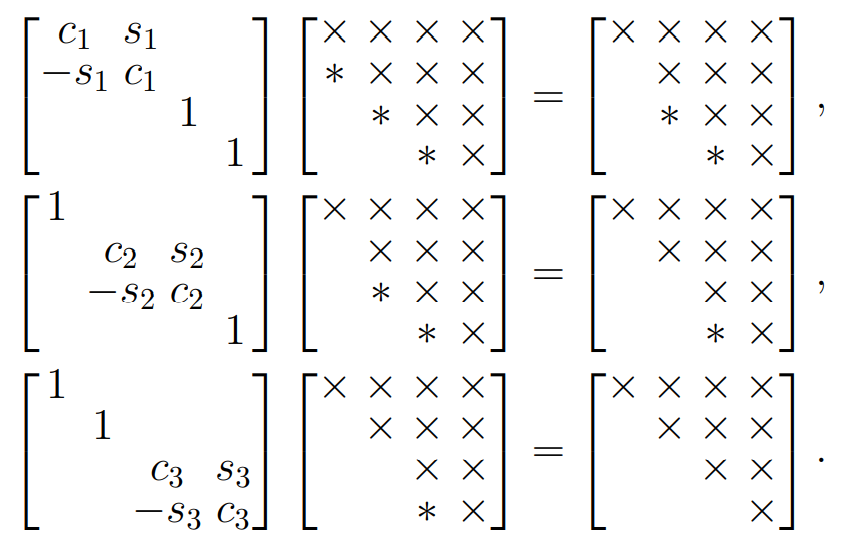
r = (405.05625 364.00677 356.42145 234.66658 208.28943 186.05878 178.88116 97.38314 34.55305 0)

Алгоритм

Шаг 1:

Приводим начальную матрицу к верхне-треугольной методом вращений. То есть получаем R матрицу из QR – разложения.

Где матрицы V – матрицы элементарного вращения.



R

Q

V

Код:

for (var i = 0; i < **n** - 1; ++i)

{  
 var sin = A[i + 1][i] / Math.Sqrt(Math.Pow(A[i][i], 2) + Math.Pow(A[i + 1][i], 2));  
 var cos = A[i][i] / Math.Sqrt(Math.Pow(A[i][i], 2) + Math.Pow(A[i + 1][i], 2));  
  
 for (var j = i; j < **n**; ++j)  
 Rotate(A[i][j], A[i + 1][j], sin, cos);  
 Rotate(B[i], B[i + 1], sin, cos);  
   
 A[i + 1][i] = 0;  
}

private static void Rotate(double p1, double p2, double sin, double cos)  
{  
 var tmp = p1;  
 p1 = cos \* p1 + sin \* p2;  
 p2 = cos \* p2 - sin \* tmp;  
}

Шаг 2:

Решаем матрицу R N раз методом обратной подстановки так как она треугольная.

for (var k = 0; k < **N**; ++k)   
{  
 for (var i = k; i >= 0; --i)  
 {  
 if (A[i][i] == 0) continue;  
   
 var ll = B[i];  
 for (var j = 0; j <= k; ++j)  
 ll -= A[i][j] \* x[j];  
   
 x[i] = ll / A[i][i];  
 }  
  
 PrintAns(Residual(k));  
}

Нахождение k – ой невязки rk = ||Akxk – b||2

private static double Residual(int k)  
{  
 var rAns = 0.0;  
 for (var i = 0; i < **N**; ++i)   
 {  
 for (var j = 0; j <= k; ++j)   
 r[i] += sourceA[i][j] \* x[j];   
 r[i] -= sourceB[i];  
 rAns += Math.Pow(r[i], 2);  
 }  
  
 return Math.Sqrt(rAns);  
}

Вывод ответа:

private static void PrintAns(double residual)  
{  
 Console.Write("( ");  
 for (var j = 0; j < **N**; ++j)   
 {  
 Console.Write((x[j] != 0 ? $"{x[j]:F5} " : "\_ ").PadLeft(10));  
 x[j] = r[j] = 0; // обнуляю массивы для новой итерации  
 }  
 Console.WriteLine($") {residual:F7}");  
}

Код целиком

using System;  
using System.IO;  
using System.Linq;  
  
namespace Lab\_2\_MV  
{  
 public static class Program  
 {  
 private const int **N** = 10;  
 private static double[][] A;  
 private static double[] B;  
 private static double[][] sourceA;  
 private static double[] sourceB;  
 private static readonly double[] x = new double[**N**];  
 private static readonly double[] r = new double[**N**];  
   
 public static void Main()  
 {

// Считывание данных  
 var rowsA = File.ReadAllLines("H.txt");  
 var strB = File.ReadAllText("b.txt");  
   
 A = (from i in rowsA select i.Split(' ').Select(double.Parse).ToArray()).ToArray();

// сохраняю исходный массив для формулы rk = ||Akxk – b||2   
 sourceA = (from i in rowsA selecti.Split('').Select(double.Parse).ToArray())

.ToArray();

B = strB.Split().Select(double.Parse).ToArray();  
 sourceB = strB.Split().Select(double.Parse).ToArray();

// Повороты, приведение к треугольной матрице  
 for (var i = 0; i < **N** - 1; ++i)   
 {  
 var cos = A[i][i] / Math.Sqrt(Math.Pow(A[i][i], 2) + Math.Pow(A[i + 1][i], 2));  
 var sin = A[i + 1][i] / Math.Sqrt(Math.Pow(A[i][i], 2) + Math.Pow(A[i + 1][i], 2));  
  
 for (var j = i; j < **N**; ++j)  
 Rotate(ref A[i][j], ref A[i + 1][j], sin, cos);  
 Rotate(ref B[i], ref B[i + 1], sin, cos);  
   
 A[i + 1][i] = 0;  
 }

// Решение rk = ||Akxk – b||2   
 for (var k = 0; k < **N**; ++k)   
 {  
 for (var i = k; i >= 0; --i)  
 {  
 if (A[i][i] == 0) continue;  
 var ll = B[i];  
 for (var j = 0; j <= k; ++j)  
 ll -= A[i][j] \* x[j];  
   
 x[i] = ll / A[i][i];  
 }  
  
 PrintAns(Residual(k));  
 }  
 }

private static void Rotate(ref double p1, ref double p2, double sin, double cos)  
 {  
 var tmp = p1;  
 p1 = cos \* p1 + sin \* p2;  
 p2 = cos \* p2 - sin \* tmp;  
 }  
  
 private static double Residual(int k)  
 {  
 var rAns = 0.0;  
 for (var i = 0; i < **N**; ++i)   
 {  
 for (var j = 0; j <= k; ++j)   
 r[i] += sourceA[i][j] \* x[j];  
 r[i] -= sourceB[i];  
 rAns += Math.Pow(r[i], 2);  
 }  
  
 return Math.Sqrt(rAns);  
 }  
  
 private static void PrintAns(double residual)  
 {  
 Console.Write("( ");  
 for (var j = 0; j < **N**; ++j)   
 {  
 Console.Write((x[j] != 0 ? $"{x[j]:F5} " : "\_ ").PadLeft(10));  
 x[j] = r[j] = 0;  
 }  
 Console.WriteLine($") {residual:F7}");  
 }  
 }  
}