Лабораторная работа №3 Доскоч Роман 2 курс 13 группа

Итерационные методы решения СЛАУ

Вариант 7

Постановка задачи

Даны две матрицы и стационарный итерационный метод. Для каждой матрицы необходимо выполнить следующие пункты.

1. Сгенерировать вектор b таким образом, чтобы точным решением СЛАУ Ax = b был вектор x = (1, 2, ..., n) T.

2. Реализовать решение полученных СЛАУ, построить диаграмму сходимости.

3. Вычислить матрицу B, соответствующую вашему методу.

4. Вычислить спектральный радиус матрицы B степенным методом.

5. Сделать выводы.

Метод релаксации, ω=1.2

A1 =

5.863818863708440 -0.277544116671075 -1.995537762171580 -0.137083711556688 0.206342778890080 -0.277544116671075 6.079151014088120 -0.744896794397772 -0.536319403140064 -2.700694409782450 -1.995537762171580 -0.744896794397772 5.680655755691160 0.247609252787904 -3.082572310388470 -0.137083711556688 -0.536319403140064 0.247609252787904 7.674360409136210 -0.412397742515552 0.206342778890080 -2.700694409782450 -3.082572310388470 -0.412397742515552 5.702013957376080

A2 =

537.537709718568000 -136.375861396643000 -18.367452727628500 228.051289483657000 113.636226542262000 -136.375861396642000 373.310283520029000 277.678528402533000 92.044049805949400 35.929463020161900 -18.367452727628500 277.678528402533000 518.767439631917000 4.644690779493230 -25.043206116413600 228.051289483657000 92.044049805949400 4.644690779493230 205.055439410727000 119.468823384937000 113.636226542262000 35.929463020161900 -25.043206116413600 119.468823384937000 519.329127718759000

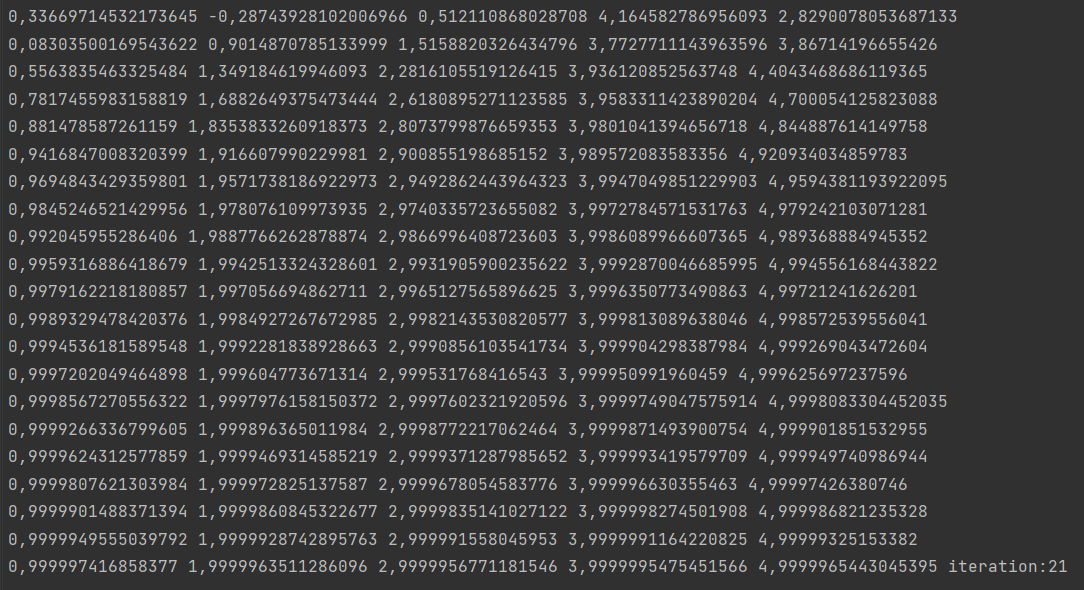
Решение

1.

Векторы b для решения х = (1, 2, 3, 4, 5)Т

|  |  |
| --- | --- |
| 1 матрица | 2 матрица |
| -0,1945036079248017  -6,002682133160657  -0,8657886246843756  28,16855816449398  12,417715844977963 | 1690,0699193883343  1991,103805175622  1986,6546555090931  1843,6393360016282  3184,886466366888 |

2.



На 21 итерации и на 1395 итерации для второй матрицы

Метод сошелся с точностью 0.00001

Код на C#

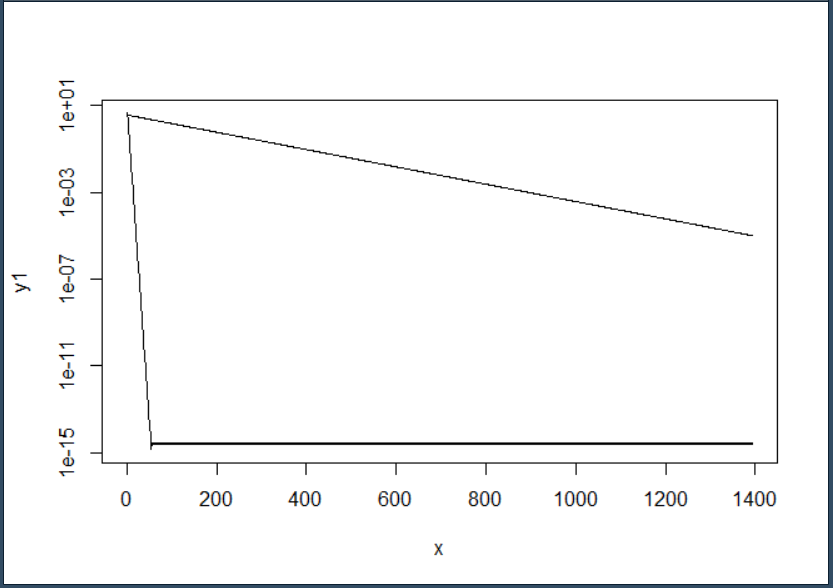
using System;  
using System.Globalization;  
using System.IO;  
using System.Linq;  
  
namespace Relaxasion

{  
 public static class Program

{  
 private const double **Omega** = 1.2;  
 private const string **APath** = "A1.txt";  
 private const string **BPath** = "b1.txt";  
 private const string **ResidualPath** = "residual1.txt";  
   
 private static double[][] A;  
 private static double[] b;  
 private static double[] x = new []{1.5, 1.5, 1.5, 1.5, 1.5};  
 private static double[] realX = new[] {1.0, 2.0, 3.0, 4.0, 5.0};  
  
 public static void Main()  
 {  
 var nfi = new NumberFormatInfo { NumberDecimalSeparator = "." };  
 var rowsA = File.ReadAllLines(**APath**);  
 var strB = File.ReadAllText(**BPath**);  
 using var writerResidual = new StreamWriter(**ResidualPath**);  
  
 *// чтение из файла данные* A = (from i in rowsA select i.Split(' ').Select(double.Parse).ToArray()).ToArray();  
 b = strB.Split().Select(double.Parse).ToArray();

var iteration = 1;  
 while (iteration < 1395)  
 {  
 for (var i = 0; i < 5; ++i) {  
 var nextX\_i = b[i];  
 for (var j = 0; j < 5; ++j) {  
 if (j == i)   
 continue;  
 nextX\_i -= A[i][j] \* x[j];  
 }  
  
 x[i] = (1 - **Omega**) \* x[i] + **Omega** \* nextX\_i / A[i][i];  
 }  
   
 foreach (var item in x)  
 Console.Write($"{item} ");  
   
 var residual = Residual();  
 Console.WriteLine();  
  
 writerResidual.WriteLine($"{residual.ToString(nfi)} ");  
 iteration++;  
 }  
  
 Console.WriteLine($"iteration:{iteration}");  
 }  
  
 private static double Residual() =>  
 Math.Sqrt(x.Select((t, i) => (t - realX[i]) \* (t - realX[i])).Sum());  
 }  
}

Диаграмма сходимости



1 матрица

2 матрица

Код на R

x <- 1:1394

y1 <- scan('residual1.txt')

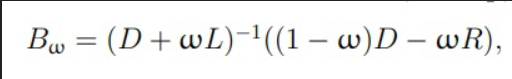
y2 <- scan('residual2.txt')

plot(x,y1,log="y", type = 'l')

lines(x,y2,log="y", type = 'l')

3.

Матрица B первой матрицы



-0.20000000 0.05679796 0.40837641 0.02805347 -0.04222698  
-0.01095722 -0.19688826 0.16941298 0.10740424 0.53079278  
-0.08603293 -0.00703841 -0.00119357 -0.02357948 0.71689455  
-0.00187493 -0.01502137 0.02300701 -0.18947866 0.08033616  
-0.05351783 -0.12024089 0.07977728 0.02808510 0.57556430

Матрица B второй матрицы

-0.20000000 0.30444568 0.04100353 -0.50910205 -0.25368168  
-0.08767561 -0.06653770 -0.87461819 -0.51905330 -0.22670317  
 0.04781829 0.05567346 0.36352609 0.30102358 0.19276689  
 0.31284130 -0.37197794 0.40650875 0.75083935 -0.24371001  
-0.02379965 0.03149128 -0.02933674 -0.01308201 -0.03613628

Код на R

w = 1.2

N = 5

A <- matrix(c(

  5.863818863708440,-0.277544116671075,-1.995537762171580,-0.137083711556688,0.206342778890080,

  -0.277544116671075,6.079151014088120,-0.744896794397772,-0.536319403140064,-2.700694409782450,

  -1.995537762171580,-0.744896794397772,5.680655755691160,0.247609252787904,-3.082572310388470,

  -0.137083711556688,-0.536319403140064,0.247609252787904,7.674360409136210,-0.412397742515552,

  0.206342778890080,-2.700694409782450,-3.082572310388470,-0.412397742515552,5.702013957376080),

  N,N,byrow = TRUE)

L = matrix(rep(0, N ^ 2), N, N, byrow = TRUE)

D = matrix(rep(0, N ^ 2), N, N, byrow = TRUE)

R = matrix(rep(0, N ^ 2), N, N, byrow = TRUE)

for(i in 1:N){

  D[i,i] = A[i,i]

  for (j in 1:N) {

    if (j < i)

      L[i,j] <- A[i,j]

    if (j > i)

      R[i,j] <- A[i,j]

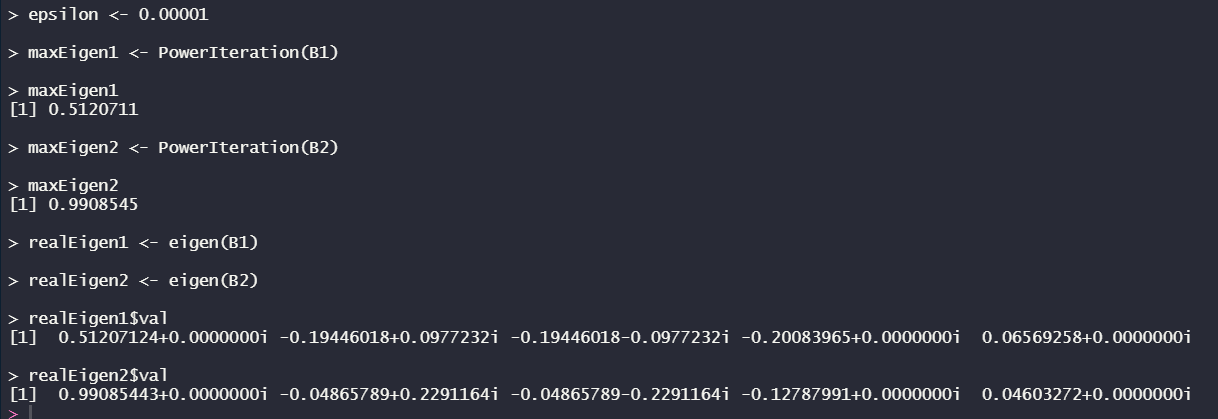
  }

}

B <- solve(D + w\*L) %\*% ((1-w)\*D - w\*R)

4.

Спектр первой и второй матриц В



Код на R

norma <- function(x) sqrt(sum(x^2))

normalize <- function(x) x / norma(x)

PowerIteration <- function(A){

  u <- normalize(y)

  Au <- A %\*% u

  lamda <- sum(Au \* u)

  while(norma(Au - lamda\*u) > epsilon){

    y <- Au

    u <- normalize(y)

    Au <- A %\*% u

    lamda <- sum(Au \* u)

  }

  lamda

}

B1 <- matrix(c(

  -0.2, 0.05679796, 0.40837641, 0.02805347, -0.04222698,

   -0.01095722, -0.19688826, 0.16941298, 0.10740424, 0.53079278,

   -0.08603293, -0.00703841, -0.00119357, -0.02357948, 0.71689455,

   -0.00187493, -0.01502137, 0.02300701, -0.18947866, 0.08033616,

   -0.05351783, -0.12024089, 0.07977728, 0.0280851, 0.5755643

  ),N,N,byrow = TRUE)

B2 <- matrix(c(

  -0.2, 0.30444568, 0.04100353, -0.50910205, -0.25368168,

   -0.08767561, -0.0665377, -0.87461819, -0.5190533, -0.22670317,

    0.04781829, 0.05567346, 0.36352609, 0.30102358, 0.19276689,

    0.3128413, -0.37197794, 0.40650875, 0.75083935, -0.24371001,

   -0.02379965, 0.03149128, -0.02933674, -0.01308201, -0.03613628

),N,N,byrow = TRUE)

y <- c(2, 1, 3, 2, 1)

epsilon <- 0.00001

maxEigen1 <- PowerIteration(B1)

maxEigen1

maxEigen2 <- PowerIteration(B2)

maxEigen2

realEigen1 <- eigen(B1)

realEigen2 <- eigen(B2)

realEigen1$val

realEigen2$val

5.

Выводы

Как видно из 4 пункта спектральный радиус матриц B меньше 1 значит итерационный процесс решения должен сходится но у 2 матрицы радиус ближе к 1 значит он будет медленее сходится что мы и видим на графике в 2 пункте.