Московский авиационный институт
(Национальный исследовательский университет)
Факультет "Информационные технологии и прикладная математика"

Лабораторные работы
по курсу “Численные методы”

Студент: Живалев Е.А.

Группа: М8О-306Б

Преподаватель: Ревизников Д.Л.

Оценка: ____________

Дата: ____________

Москва
2021

1 Лабораторная работа №1

1.1 Часть 1

Задание

Реализовать алгоритм LU-разложения матриц (с выбором главного элемента) в виде программы. Используя разработанное программное обеспечение, решить систему линейных алгебраических уравнений(СЛАУ). Для матрицы СЛАУ вычислить определитель и обратную матрицу.

$$\begin{cases} 3x_1 - 8x_2 + x_3 - 7x_4 = 96 \\ 6x_1 + 4x_2 + 8x_3 + 5x_4 = -13 \\ -x_1 + x_2 - 9x_3 - 3x_4 = -54 \\ -6x_1 + 6x_2 + 9x_3 - 4x_4 = 82 \end{cases}$$

Резултат

```
Matrix L:
1.00000 0.00000 0.00000 0.00000
0.50000 1.00000 0.00000 0.00000
-1.00000 -1.00000 1.00000 0.00000
-0.16667 -0.16667 -0.58333 1.00000
Matrix U:
6.00000 4.00000 8.00000 5.00000
0.00000 -10.00000 -3.00000 -9.50000
0.00000 0.00000 14.00000 -8.50000
0.00000 0.00000 0.00000 -8.70833
P * L * U:
3.00000 -8.00000 1.00000 -7.00000
6.00000 4.00000 8.00000 5.00000
-1.00000 1.00000 -9.00000 -3.00000
-6.00000 6.00000 9.00000 -4.00000
Ax = b solution:
-3.00000
-6.00000
8.00000
-7.00000
Determinant of matrix A: 7315.00000
Inverse of matrix A:
0.06220 0.13110 0.10198 -0.02146
-0.02392 0.08804 0.13001 0.05441
0.01914 0.00957 -0.06972 0.03076
-0.08612 -0.04306 -0.11483 -0.06699
Product of matrix A and its' inverse matrix
1.00000 0.00000 0.00000 -0.00000
-0.00000 1.00000 0.00000 0.00000
-0.00000 -0.00000 1.00000 -0.00000
0.00000 0.00000 0.00000 1.00000
```

1.2 Часть 2

Задание

Реализовать метод прогонки в виде программы, задавая в качестве входных данных ненулевые элементы матрицы системы и вектор правых частей. Используя разработанное программное обеспечение, решить СЛАУ с трехдиагональной матрицей.

$$\begin{cases} 8x_1 + 4x_2 = 48 \\ -5x_1 + 22x_2 + 8x_3 = 125 \\ -5x_2 - 11x_3 + x_4 = -43 \\ -9x_3 - 15x_4 + x_5 = 18 \\ x_4 + 7x_5 = -23 \end{cases}$$

Результат

```
Ax = b solution:
3.00000
6.00000
1.00000
-2.00000
-3.00000
```

1.3 Часть 3

Задание

Реализовать метод простых итераций и метод Зейделя в виде программ, задавая в качестве входных данных матрицу системы, вектор правых частей и точность вычислений. Используя разработанное программное обеспечение, решить СЛАУ. Проанализировать количество итераций, необходимое для достижения заданной точности.

$$\begin{cases} 20x_1 + 5x_2 + 7x_3 + x_4 = -117 \\ -x_1 + 13x_2 - 7x_4 = -1 \\ 4x_1 - 6x_2 + 17x_3 + 5x_4 = 49 \\ -9x_1 + 8x_2 + 4x_3 - 25x_4 = -21 \end{cases}$$

Результат

```
Ax = b solution:
Matrix alpha l1 norm: 0.922941
Simple iterations method took 20 iterations
-7.99998
1.99997
3.99999
4.99997
Exact solution:
-8.00000
2.00000
4.00000
5.00000
```

```
Ax = b solution:
Matrix alpha l1 norm: 0.852371
Seidel's method took 10 iterations
-7.99998
1.99998
4.00000
4.99998
Exact solution:
-8.00000
2.00000
4.00000
5.00000
```

1.4 Часть 4

Задание

Реализовать метод вращений в виде программы, задавая в качестве входных данных матрицу и точность вычислений. Используя разработанное программное обеспечение, найти собственные значения и собственные векторы симметрических матриц. Проанализировать зависимость погрешности вычислений от числа итераций.

$$\begin{pmatrix} 0 & -7 & 7 \\ -7 & -9 & -5 \\ 7 & -5 & -1 \end{pmatrix}$$

Резултат

```
Jacobi's rotation method took 6 iterations
Eigenvalues:
a_1 = 10.3014
a_2 = -12.9621
a_3 = -7.3393
Eigenvectors:
x_1 = (0.686013, 0.413773, -0.59848)
x_2 = (-0.405322, 0.900429, 0.157928)
x_3 = (0.604235, 0.134236, 0.785417)
A * x_1:
7.06690
-4.17537
6.22447
a_1 * x_1:
7.06689
-4.17538
6.22447
A * x_2:
-5.36335
-11.67145
-1.73997
a_2 * x_2:
-5.36336
```

```
-11.67145
-1.73998
A * x_3:
4.39242
-1.15908
-5.76442
a_3 * x_3:
4.39242
-1.15908
-5.76442
```

1.5 Часть 5

Задание

Реализовать алгоритм QR–разложения матриц в виде программы. На его основе разработать программу, реализующую QR–алгоритм решения полной проблемы собственных значений произвольных матриц, задавая в качестве входных данных матрицу и точность вычислений. С использованием разработанного программного обеспечения найти собственные значения матрицы.

$$\begin{pmatrix} 5 & 8 & -2 \\ 7 & -2 & -4 \\ 5 & 8 & -1 \end{pmatrix}$$

Результат

```
QR Algorithm took 18 iterations
Eigenvalues:
a_1 = (17.5518,0)
a_2 = (-1.74559,-1.50105)
a_3 = (-1.74559,1.50105)
a_4 = (-0.446158,0)
a_5 = (0.385581,0)
```

2 Лабораторная работа №2

2.1 Часть 1

Задание

Реализовать методы простой итерациии Ньютона решения нелинейных уравнений в виде программы, задавая в качестве входных данных точность вычислений. С использованием разработанного программного обеспечения найти положительный корень нелинейного уравнения(начальное приближение определить графически). Проанализировать зависимость погрешности вычислений от количества итераций.

$$cosx + 0,25x - 0,5 = 0$$

Результат

```
Root found by Newton's method:
Newton's method took 3 iterations
4.14608
Root found by simple iterations method:
Simple iterations method took 6 iterations
1.42715
```

2.2 Часть 2

Задание

Реализовать методы простой итерациии Ньютона решения систем нелинейных уравнений в виде программы, задавая в качестве входных данных точность вычислений. С использованием разработанного программного обеспечения решить систему нелинейных уравнений(при наличии нескольких решений найти то из них, в котором значения неизвестных являются положительными); начальное приближение определить графически. Проанализировать зависимость погрешности вычислений от количества итераций.

$$\begin{cases} -cosx_2 + x_1 = 1 \\ -\lg(x_1 + 1) + x_2 = 3 \end{cases}$$

Результат

```
Solution found by Newton's method:
Newton's method took 38 iterations
x1: 0.444513 x2: 2.15974
Solution found by simple iterations method:
Simple iterations method took 15 iterations
x1: 0.44456 x2: 2.15972
```

3 Лабораторная работа №3

3.1 Часть 1

Задание

Используя таблицу значений Y_i функции $y = f(x)$, вычисленных в точках X_i, $i = 0, \ldots, 3$ построить интерполяционные многочлены Лагранжа и Ньютона, проходящие через точки $\{X_i, Y_i\}$. Вычислить значение погрешности интерполяции в точке X^*

$$y = ln(x), \quad a)\ X_i = 0.2, 0.6, 1.0, 1.4; \quad \text{б)}\ X_i = 0.2, 0.6, 1.0, 1.4; \quad X^* = 0.8$$

Результат

```
Lagrange's interpolation result for the first set of points: -0.207779
Actual value is -0.223144
Absolute error is 0.0153645
```

```
Newton's interpolation result for the second set of points: -0.207779
Actual value is -0.223144
Absolute error is 0.0153645
```

3.2 Часть 2

Задание

Построить кубический сплайн для функции, заданной в узлах интерполяции, предполагая, что сплайн имеет нулевую кривизну при $x = x_0, x = x_4$. Вычислить значение функции в точке $x = x^*$

$$X^* = 0.8$$

i	0	1	2	3	4
x_i	0.1	0.5	0.9	1.3	1.7
f_i	-2.3026	-0.69315	-0.10536	0.26236	0.53063

Результат

```
Result of interpolation: -0.197081
Spline #1, range: [0.1, 0.5], a: -2.3026, b: 4.67291, c: 0, d:-4.05806
Spline #2, range: [0.5, 0.9], a: -0.69315, b: 2.72505, c: -4.86967, d:4.32684
Spline #3, range: [0.9, 1.3], a: -0.10536, b: 0.906197, c: 0.32254, d:-0.724456
Spline #4, range: [1.3, 1.7], a: 0.26236, b: 0.81649, c: -0.546807, d:0.455672
```

3.3 Часть 3

Задание

Для таблично заданной функции путем решения нормальной системы МНК найти приближающие многочлены а)1-ой и б) 2-ой степени. Для каждого из приближающих многочленов вычислить сумму квадратов ошибок. Построить графики приближаемой функции и приближающих многочленов.

i	0	1	2	3	4	5
x_i	0.1	0.5	0.9	1,3	1.7	2.1
y_i	-2.3026	-0.69315	-0.10536	0.26236	0.53063	0.74194

Результат

```
Approximation using first order polynom
Sum of the squares of residuals is 0.98536
Resulting function: -1.77445 + 1.37584 * x
Approximation using second order polynom
Sum of the squares of residuals is 0.171399
Resulting function: -2.46044 + 3.40612 * x + -0.922855 * x^2
```

3.4 Часть 4

Задание

Вычислить первую и вторую производную от таблично заданной функции $y_i = f(x_i), i = 0, 1, 2, 3, 4$ в точке $x = X^*$.

$$X^* = 2.0$$

i	0	1	2	3	4
x_i	0.0	1.0	2.0	3.0	4.0
y_i	0.0	1.0	1.4142	1.7321	2.0

Резултат

```
First derivative: 0.390125
Second derivative: -0.0963
```

3.5 Часть 5

Задание

Вычислить определенный интеграл $F = \int_{x_0}^{x_1} y dx$, методами прямоугольников, трапеций, Симпсон с шагами h_1, h_2. Оценить погрешность вычислений, используя метод Рунге-Ромберга

$$y = \frac{1}{(2x+7)(3x+4)}, \quad X_0 = -1, X_k = 1, h_1 = 0.5, h_2 = 0.25$$

Резултат

```
Exact value is 0.104471
Result of integration using rectangle method with step 0.5: 0.0990635
Result of integration using rectangle method with step 0.25: 0.102869
Result of integration using trapezoid method with step 0.5: 0.116522
Result of integration using trapezoid method with step 0.25: 0.107793
Result of integration using Simpson's method with step 0.5: 0.10748
Result of integration using Simpson's method with step 0.25: 0.104883
Error for rectangle method counted using Runge-Romberg method: -0.00126856
Error for trapezoid method counted using Runge-Romberg method: 0.0029098
Error for Simpson's method counted using Runge-Romberg method: 0.000173094
```

4 Лабораторная работа №4

4.1 Часть 1

Задание

Реализовать методы Эйлера, Рунге-Кутты и Адамса 4-го порядка в виде программ, задавая в качестве входных данных шаг сетки h. С использованием разработанного программного обеспечения решить задачу Коши для ОДУ 2-го порядка на указанном отрезке. Оценить погрешность численного решения с использованием метода Рунге–Ромберга и путем сравнения с точным решением.

$$y'' - (1 + 2tg^2 x)y = 0, \quad y(0) = 1, \quad y'(0) = 2, \quad x \in [0, 1], h = 0.1$$

Точное решение

$$y = \frac{1}{cosx} + sinx + \frac{x}{cosx}$$

Резултат

```
Exact solution:
{x: 0.0000000, y: 1.0000000}
{x: 0.1000000, y: 1.2053564}
{x: 0.2000000, y: 1.4230759}
{x: 0.3000000, y: 1.6562973}
{x: 0.4000000, y: 1.9094045}
{x: 0.5000000, y: 2.1886664}
{x: 0.6000000, y: 2.5032478}
{x: 0.7000000, y: 2.8668984}
{x: 0.8000000, y: 3.3009397}
{x: 0.9000000, y: 3.8399059}
{x: 1.0000000, y: 4.5431024}
Solution using Euler's method:
{x: 0.0000000, y: 1.0000000}
{x: 0.1000000, y: 1.2000000}
{x: 0.2000000, y: 1.4120000}
{x: 0.3000000, y: 1.6384043}
{x: 0.4000000, y: 1.8825391}
{x: 0.5000000, y: 2.1491021}
{x: 0.6000000, y: 2.4448393}
{x: 0.7000000, y: 2.7796180}
{x: 0.8000000, y: 3.1682125}
{x: 0.9000000, y: 3.6334428}
{x: 1.0000000, y: 4.2120479}
At point x = 1 absolute error for Euler's method is: 0.3310545
At point x = 1 error for Euler's method counted
using Runge-Romberg method: 0.1846072
Solution using Runge-Kutta fourth order method:
{x: 0.0000000, y: 1.0000000}
{x: 0.1000000, y: 1.2053559}
{x: 0.2000000, y: 1.4230747}
{x: 0.3000000, y: 1.6562950}
{x: 0.4000000, y: 1.9094007}
{x: 0.5000000, y: 2.1886602}
{x: 0.6000000, y: 2.5032373}
{x: 0.7000000, y: 2.8668803}
{x: 0.8000000, y: 3.3009063}
{x: 0.9000000, y: 3.8398397}
{x: 1.0000000, y: 4.5429557}
At point x = 1 absolute error for Runge-Kutta fourth order method is: 0.0001467
At point x = 1 error for Runge-Kutta fourth order method counted
using Runge-Romberg method: 0.0251470
Solution using Adams' fourth order method:
{x: 0.0000000, y: 1.0000000}
```

```
{x: 0.1000000, y: 1.2053559}
{x: 0.2000000, y: 1.4230747}
{x: 0.3000000, y: 1.6562950}
{x: 0.4000000, y: 1.9092277}
{x: 0.5000000, y: 2.1880631}
{x: 0.6000000, y: 2.5019333}
{x: 0.7000000, y: 2.8642337}
{x: 0.8000000, y: 3.2956990}
{x: 0.9000000, y: 3.8294922}
{x: 1.0000000, y: 4.5215632}
At point x = 1 absolute error for Adams' fourth order method is: 0.0215392
At point x = 1 error for Adams' fourth order method counted
using Runge-Romberg method: 0.0238297
```

4.2 Часть 2

Задание

Реализовать методы простой итерациии Ньютона решения систем нелинейных уравнений в виде программы, задавая в качестве входных данных точность вычислений. С использованием разработанного программного обеспечения решить систему нелинейных уравнений(при наличии нескольких решений найти то из них, в котором значения неизвестных являются положительными); начальное приближение определить графически. Проанализировать зависимость погрешности вычислений от количества итераций.

$$y'' - 2(1 + (tgx)^2)y = 0, \quad y(0) = 0, \quad y(\frac{\pi}{6}) = -\frac{\sqrt{3}}{3}$$

Точное решение

$$y = -tgx$$

Резултат

```
Exact solution:
{x: 0, y: -0}
{x: 0.1, y: -0.100335}
{x: 0.2, y: -0.20271}
{x: 0.3, y: -0.309336}
{x: 0.4, y: -0.422793}
{x: 0.5, y: -0.546302}
Solution using Shooting method:
{x: 0, y: 0}
{x: 0.1, y: -0.106037}
{x: 0.2, y: -0.214231}
{x: 0.3, y: -0.326917}
{x: 0.4, y: -0.446822}
{x: 0.5, y: -0.57735}
Absolute error for shooting method at point x = 0.3 is: 0.017581
Error using Runge-Romberg method for shooting
method at point x = 0.3 is: -0.0570635
```

```
Solution using Finite difference method:
{x: 0, y: 0}
{x: 0.1, y: -0.0273469}
{x: 0.2, y: -0.104415}
{x: 0.3, y: -0.224991}
{x: 0.4, y: -0.384026}
{x: 0.5, y: -0.57735}
Absolute error for finite difference method at point x = 0.3 is: 0.0843457
Error using Runge-Romberg method for finite difference
method at point x = 0.3 is: -0.0654845
```