

Лабораторная работа №1

Бочкарев Влад ФИТ-2

Вариант 2

Система из варианта:

$$\left(\begin{array}{ccc|c} 1.7 & 2.8 & 1.9 & 0.7 \\ 2.1 & 3.4 & 1.8 & 1.1 \\ 4.2 & -1.7 & 1.3 & 2.8 \end{array}\right)$$

Правильное решение найденное через калькулятор:

$$\begin{pmatrix} 0.820456 \\ 0.0471661 \\ -0.435179 \end{pmatrix}$$

## Задание 1.

Метода Гаусса с выбором элементов был реализован на языке c#.

Кратко метод решения:

```
public Vector<TFloat> Solve(Matrix<TFloat> m, Vector<TFloat> coefficients)
{
    Utils.ThrowIfNotLinearSolvable(m, coefficients);
    m = m.Clone();
    coefficients = coefficients.Clone();

    int[] pivots = Enumerable.Range(0, coefficients.Count).ToArray();
    int row = 0;
    int column = 0;
    for(int i = 0; i < Math.Min(m.ColumnCount, m.RowCount); i++){
```

```

        SwapPivot(m,coefficients,pivots, row+i,column+i);
        Elimination(m,coefficients,row+i,column+i);
    }
    return m.UForwardSubstitution(coefficients);
}

```

1. Проверяем условия квадратности матрицы, минимального размера матрицы(больше чем 2 на 2). Соответствие размера вектора.
2. Сохраняем перестановки строк в массив индексов `pivots`
3. Поочередно выполняем перестановку доминирующего элемента на главную диагональ и делаем элюминацию гаусса
4. В конце выполняем прямую подстановку вектора коэффициентов для полученной треугольной матрицы

При отсутствия решения, в последнем этапе прямой подстановке в результате получится вектор с `NaN` решением -> происходит это при делении на ноль.

Данная программа не завершается аварийно.

0. Пример с данной вариантом матрицей:

```

DenseMatrix 3x4-Double
1.7    2.8    1.9    0.7
2.1    3.4    1.8    1.1
4.2   -1.7    1.3    2.8

Gaussian solver
DenseVector 3-Double
0.820456
0.0471661
-0.435179

```

1. Пример с бесконечным кол-вом решений:

```
DenseMatrix 3x3-Double
```

```
0  1  2
```

```
3  4  5
```

```
6  7  8
```

```
DenseVector 3-Double
```

```
0
```

```
1
```

```
2
```

```
DenseVector 3-Double
```

```
NaN
```

```
NaN
```

```
NaN
```

2. Пример с одним решением:

```
DenseMatrix 3x3-Double
```

```
1 1 2
```

```
3 4 5
```

```
6 7 8
```

```
DenseVector 3-Double
```

```
0
```

```
1
```

```
2
```

```
DenseVector 3-Double
```

```
0
```

```
0.666667
```

```
-0.333333
```

3. Пример где отсутствия решений:

```
DenseMatrix 3x3-Double
```

```
0  1  0  
3  4  0  
6  7  0
```

```
DenseVector 3-Double
```

```
0  
1  
2
```

```
DenseVector 3-Double
```

```
NaN  
NaN  
NaN
```

## Задание 2.

Сначала приведём матрицу к форме с преобладанием диагональных элементов

$$\begin{pmatrix} 1.7 & 2.8 & 1.9 & | & 0.7 \\ 2.1 & 3.4 & 1.8 & | & 1.1 \\ 4.2 & -1.7 & 1.3 & | & 2.8 \end{pmatrix}$$
$$\begin{pmatrix} 5.9 & 1.1 & 3.2 & | & 3.5 \\ -2.1 & 5.1 & 0.5 & | & -1.7 \\ 4.2 & -0.7 & 10.6 & | & -1.2 \end{pmatrix} \begin{matrix} (I + III) \\ (II - III) \\ (III + 10 \cdot (2.1 \cdot I - 1.7 \cdot II)) \end{matrix}$$

Метод Гаусса-Зеделя

Кратко метод решения:

```

public Vector<TFloat> Solve(Matrix<TFloat> m, Vector<TFloat> coefficients)
{
    Utils.ThrowIfNotLinearSolvable(m, coefficients);
    if(!DiagonalDominance(m))
        throw new DiagonalMatrixException("Matrix m must be diagonal dominant");
    Iterations = 0;
    var solution = coefficients.Map(x=>x);
    var prevSolution = coefficients.Map(x=>x);
    do{
        prevSolution = solution;
        solution = m.ForwardSubstitution(coefficients, prevSolution);
        Iterations++;
    }
    while(!Converges(solution, prevSolution));
    return solution;
}

```

1. Нужно привести исходную систему к системе с диагональным преобладанием.  
Сделать это программно очень сложно, поэтому ограничим входные данные только матрицами с преобладанием диагональных элементов.
2. Далее нужно решить систему  $M \cdot x_{next} + b = x_{prev}$   
Т.к нам известно решение, мы можем так же использовать метод подстановки как мы делали с треугольной матрицей, единственное различие в том, что теперь мы делаем подстановку по известным значениям с прошлого решения, динамически обновляя их, и сразу же используя новые значения(см реализацию метода `ForwardSubstitution`)
3. Мы выполняем шаг (2) до тех пор, пока значения предыдущего решения и текущего не начнут сходиться по условию меньше заданного эпсилон.

Пример решения для данной в варианте матрицы

```
DenseMatrix 3x4-Double
```

```
1.7    2.8    1.9    0.7
```

```
2.1    3.4    1.8    1.1
```

```
4.2   -1.7    1.3    2.8
```

```
DenseMatrix 3x4-Double
```

```
5.9    1.1    3.2    3.5
```

```
-2.1    5.1    0.5   -1.7
```

```
4.2   -0.7   10.6   -1.2
```

```
Gaussian zedel solver
```

```
DenseVector 3-Double
```

```
0.820456
```

```
0.0471661
```

```
-0.435179
```