Systems of linear equations

dr Szymon Murawski

November 12, 2019

1 Systems of linear equations

1.1 Introduction

Systems of linear equations are sets of equations with linear coefficients. Numerical linear algebra deals with algorithms that can solve this types of systems, as they can describe problems in may fields: engineering, signal processing, economy, bioinformatics and so on.

Even though most of the phenomena in real world is described by non-linear equations, approximation by linear equations can give surprisingly good results, or at least an insight into the problem.

Typically systems of linear equations have coefficients expressed as real numbers, but in fact they can be anything: integer numbers, exotic algebraic structures or even other linear equations! Algorithms that solve systems of linear equations are very general, so if only there is a way of performing certain operations on coefficients (like substraction, multiplication and division) they can be applied to this advanced systems.

1.2 Definition

A system of linear equations has the following form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_2 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots \quad \cdots \quad \cdots \quad \cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned}$$

Coefficients $a_{11}, a_{12}, \ldots, a_{nn}$ are known, as well as constant terms $b_1, b_2, \ldots, b_n$. The unknowns are $x_1, x_2, \ldots, x_n$.

A solution of system of linear equations is set of variables $x_1, x_2, \ldots, x_n$ such that it satisfies all the equations in the set.

For example, the following system of linear equations:

$$\begin{aligned} 2x + 3y &= 6 \\ 4x + 9y &= 15 \end{aligned}$$

has the solution

$$x = 3/2$$
$$y = 1$$

Systems of linear equations can be expressed as a matrix in the following way:

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}}_{coefficients} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{variables} \approx \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{constants}$$

Which can be simplified into this form of **augmented matrix**:

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right]$$

1.3 Algorithms

Over the years many different methods and algorithms were proposed. Some of the methods:

- Graphical method - every equation is drawn on n-dimensional plane, n being the number of variables in equation. The point in which all equations meet is the solution. This method is unpractical for $n > 3$
- Using Cramer's rule - each variable is calculated as the fraction of two matrix determinants. This method is unsuitable for large systems, as calculating determinant is very costly
- Elimination method - step by step we eliminate variables, by substitution. In the end we obtain a single equation with a single variable, and by backtracking and some more substitution we obtain the final solution
- Gaussian elimination - it's formalized elimination method. We represent the system in the form of augmented matrix and reduce it to upper triangular state, from which by backtracking we can solve the system
- Gauss-Jordan elimination method - similar to the above, but we transform the matrix into reduced raw echelon form, which removes the need of backtracking

2 Gaussian elimination algorithm

Core of Gaussian elimination method is transforming the augmented matrix that represents system of linear equations, into upper triangular form. To do this we perform **elementary row operations** on the matrix, that consist of:

- Swapping two rows
- Multiplying a row by a nonzero number
- Adding (or substracting) a multiple of one row from another

Using these three operations we can transform any matrix into upper triangular form, which then leads us to the solution. It is worth noting, that performing any of those operations does not change the result set.

2.1 The algorithm

The algorithm consists of the following steps:

1. Write the system of linear equations as the augmented matrix
2. Using elementary row operations transform the matrix, so that the first coefficient is non-zero in the first row, but zero in all the others
3. Repeat the above steps for the rest of coefficients (for example second coefficient is non-zero in first and second row, but zero in others and so on)
4. In the end we obtained upper triangular matrix
5. Last row consists of only one variable, so it's trivial to solve it
6. Use the solution from previous step as substitution for variable in second to last row. It now has only one variable, so we can solve it easily
7. Repeat the above steps for the rest of the rows

Steps 2-3 are called **forward elimination**, while steps 5-7 are called **back substitution**.

Step 2 can be performed by choosing a_{mn} as a pivot element and then from every row m', $m' > m$ subtracting $a_{m'n}/a_{mn}$ times the row itself.

2.2 Example

Let's solve the following system:

$$\begin{aligned} 6x_1 - 2x_2 + 2x_3 + 4x_4 &= 16 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 &= 26 \\ 3x_1 - 13_x2 + 9x_3 + 3x_4 &= -19 \\ -6x_2 + 4x_2 + x_3 - 18x_4 &= -34 \end{aligned}$$

Step 0 - form augmented matrix

$$\left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 12 & -8 & 6 & 10 & 26 \\ 3 & -13 & 9 & 3 & -19 \\ -6 & 4 & 1 & -18 & -34 \end{array}\right]$$

Step 1 - forward elimination

Step 1.1 Eliminate x_1, pivot $p = 6$

$$\left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -4 & 2 & 2 & -6 \\ 0 & -12 & 8 & 1 & -27 \\ 0 & 2 & 3 & -14 & -18 \end{array}\right]$$

How we've obtained the above result? Let's focus on second row. First we calculated the factor f, that made the first coefficient in second row equal to zero: $f = a_{21}/a_{11} = 12/6 = 2$. Next for every coefficient a_{2n} in the second row we did the following: $a_{2n} = a_{2n} - f * a_{1n}$. This made the first coefficient zero (as $12 - 2 * 6 = 0$) and every other coefficient and constant value in the row changed value.

Step 1.1 Eliminate x_2, pivot $p = -4$

$$\left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -4 & 2 & 2 & -6 \\ 0 & 0 & 2 & -5 & -9 \\ 0 & 0 & 4 & -13 & -21 \end{array}\right]$$

Since we are removing coefficients from second row, first row is unchanged!

Step 1.1 Eliminate x_3, pivot $p = 2$

$$\left[\begin{array}{cccc|c} 6 & -2 & 2 & 4 & 16 \\ 0 & -4 & 2 & 2 & -6 \\ 0 & 0 & 2 & -5 & -9 \\ 0 & 0 & 0 & -3 & -3 \end{array}\right]$$

Step 2 - backward substitution

Step 2.1 Find x_4

$$x_4 = \frac{b_4}{a_{44}} = \frac{-3}{-3} = 1$$

Step 2.2 Find x_3

$$x_3 = \frac{b_3 - a_{34}x_4}{a_{33}} = \frac{-9 - (-5) * 1}{2} = -2$$

Step 2.3 Find x_2

$$x_2 = \frac{b_2 - a_{23}x_3 - a_{24}x_4}{a_{22}} = \frac{-6 - 2 * (-2) - 2 * 1}{-4} = 1$$

Step 2.4 Find x_1

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3 - a_{14}x4}{a_{11}} = \frac{16 - (-2) * 1 - 2 * (-2) - 4 * 1}{6} = 3$$

2.3 Pseudocodes

Forward elimination

Assuming we have N equations and N variables in a matrix A;

```
1 for (k=0; k<N; k++){
2   for (int i = k + 1; i < n; i++){
3     double factor = A[i, k] / A[k,k];
4     for (int j = k; j < N; j++){
5       A[i,j] -= factor * A[k,j];
6     }
7  }
8 }
```

Backwards substitution

Assume we store the result in array V

```
1 V = new double[N - 1];
2 for (int i = N - 1; i >= 0; i--){
3   V[i] = A[i, N - 1]; //result is equal to right hand side of the
        equation at the moment
4     for (int j = i + 1; j < N - 1; j++){
5     //left hand side of the equation is subtracted
6       V[i] -= A[i, j] * V[j];
7     }
8   //finally we divide by the coefficient
9   V[i] /= M[i, i];
10 }
```

2.4 Pivoting

Consider the following set of equations:

$$\left[\begin{array}{cc|c} 0 & 2 & 4 \\ 1 & -3 & 3 \end{array}\right]$$

If we would to continue solving this set as described before, we would end with division by zero, thanks to pivot $p = 0$.

To properly handle this type of problems, we need to add an additional step to forward elimination: **pivoting**. At every step we look for the largest coefficient in the column, and swap rows, so the largest coefficient ends up on the diagonal. After pivoting above example would end up like this:

$$\left[\begin{array}{cc|c} 1 & -3 & 3 \\ 0 & 2 & 4 \end{array}\right]$$