MATH4370 Project: Numerical Analysis of Direct and Iterative Methods for Solving Systems of Linear Algebraic Equations

Yaroslav Mikhaylik

December 12, 2022

Abstract

This paper presents a detailed study of the numerical analysis of systems of linear equations. We focus on the issue of round-off error, which occurs due to the limited precision of decimal floating point numbers and can lead to inaccuracies in the solutions. To address this problem, we investigate both direct and iterative methods for solving these systems, examining their strengths and weaknesses through specific examples. We pay particular attention to classical iterative methods, such as the Jacobi and Gauss-Seidel methods, and discuss their limitations and potential pitfalls. All calculations are done using Python code and pseudo-random invertible matrices constructed from elementary matrices. The final results provide a comprehensive analysis of different approaches for solving systems of linear equations and offer insights into the effectiveness of these methods.

1 Decimal Machine Numbers

1.1 Introduction

Floating point numbers are a common form of real numbers in computational machine operations. In general, the main parameters of this form of representation are the numerical base β and the precision p. Let's define a general machine number in the following way:

$$\pm d_0.d_1d_2...d_{p-1} \times \beta^e \tag{1}$$

More precisely, equation (1) represents a number [5]:

$$\pm(d_0 + d_1\beta^{-1} + d_2\beta^{-2} + ... + d_{p-1}\beta^{p-1}) \times \beta^e, 0 \le d_i \le \beta \tag{2}$$

The part $\cdot.d_1d_2...d_{p-1}$ is called the *mantissa* of the number and consists of p digits, and an integer exponent e. However, most often machine numbers are represented by *normalized decimal floating point form* [2], where $\beta = 10$ and $d_1 \neq 0$:

$$\pm 0.d_1d_2...d_n \times 10^e, 1 \le d_1 \le 9, 0 \le d_i \le 9, i \ge 2 \tag{3}$$

1.2 Precision Problem

In the process of performing many consecutive numerical calculations it is common to encounter a discrepancy between the result obtained using exact arithmetic and the result obtained using finite precision arithmetic. This discrepancy, also known as round-off error, can lead to a loss of calculation accuracy and, as a result, to completely different results being obtained [7].

In general, round-off error occurs when numbers are represented using a limited number of digits, which can cause a loss of precision during calculations. This is because many numbers, such as those with a decimal point, cannot be represented exactly using a finite number of digits. As a result, these numbers must be rounded to the nearest representable value, which can introduce errors into the calculation. While these errors may be small for individual calculations, they can accumulate and lead to significant errors in the final result.

As an example, suppose we have an exact value of $\pi = 3.14159265359\ldots$ and somehow we need to provide it to a computer. For this we have to limit the precision of decimal machine numbers and deal with the imprecision that happens due to the loss of the infinitely many digits. As discussed above, we can limit our normalized decimal floating point numbers to use $n = 8$ digits for the mantissa, and to deal with the leftover digits we could employ a *rounding* technique [2]. So our conversion from exact reals is defined as:

$$fl(y) = \begin{cases} 0.d_1d_2...d_{n-1}(d_n + 1), \text{if } d_{n+1} > 5, \\ 0.d_1d_2...d_{n-1}d_n, \text{if } d_{n+1} \le 5 \end{cases} \tag{4}$$

where $fl(y)$ will be the normalized decimal floating-point form of y [8]. So with $n = 8$ we have:

$$fl(\pi) = 3.14159265$$

while with $n = 4$ we have:

$$fl(\pi) = 3.1416$$

It should be noted that there are different techniques for dealing with conversions, including digit chopping and other methods of rounding (i.e.: ceiling or flooring the last mantissa digit) [8], but we will be using the method described in equation (4) going forward.

2 Direct Methods for Solving SLAEs

2.1 Gaussian Elimination

The Gaussian Elimination method is a row reduction algorithm for solving linear systems of equations and is a good problem to analyze for potential round-off error issues. The algorithm consists of the following steps with a total complexity of $O(n^3)$, performed on an augmented matrix $[A|b]$:

1. Construct a multiplier $m_{j,i} = \dfrac{a_{j,i}}{a_{i,i}}$,
2. If a division by zero is encountered, swap the rows to the first one with a non-zero a entry,
3. Replace all rows A_j below A_i with $A_j - m_{j,i} \times A_i$,
4. Go to the next column (excluding augmented column $\mathbf{b}$) and repeat.

Upon terminating the algorithm after going through each column we end up with an upper triangular matrix. We then can solve for the unknowns via *back substitution* [13].

We will consider the following general system of n equations:

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i, 1 \leq i \leq n \tag{5}$$

which can be rewritten in matrix notation as:

$$A\mathbf{x} = \mathbf{b} \tag{6}$$

In general, the solution $\hat{x}$ computed in decimal floating point arithmetic is not exactly equal to x, but it still may be acceptable if it satisfies any of the following criteria:

1. It satisfies the precision requirements of the problem. This usually involves some measure of how far $\hat{x}$ is from x, i.e.: relative error is below a certain epsilon-threshold.
2. It is the exact solution for a system which differs by less than the uncertainty in the data, so that is theoretically possible that $\hat{x}$ exactly solves the original problem, i.e.: the system is well-conditioned.

In order to demonstrate round-off errors in a clear way, in all further calculations our systems of linear equations are generated with an accuracy of up to $n = 8$ mantissa digits in a normalized decimal floating point form, while the solving processes will be carried out with an accuracy of up to $n = 4$ mantissa digits also in a normalized form, with the same $fl(y)$ definition as in equation (4). This way we will be dealing with less precision in the solving process, which will demonstrate the round-off error quite well.

Using the `sys_random()` function with `size = 4` (the dimension of the matrix) and `seed = 24` (starting number, the point at which the sequence of pseudo-random numbers begins) parameters, we obtain the following system (Attachment 6.1):

$$\begin{bmatrix} 25 & 0 & -1440 & 0 \\ 0 & -648 & 3240 & 0 \\ 150 & -252 & -7416 & 1200 \\ 30 & 0 & -1728 & 240 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -1025325 \\ 2472120 \\ -4983690 \\ -1183830 \end{bmatrix} \tag{7}$$

The exact solution and solution using `sys_solve_gaussian_elimination()` function for this system are:

$$X_{exact} = \begin{bmatrix} 459 \\ -215 \\ -720 \\ 194 \end{bmatrix}, \quad X_{numeric} = \begin{bmatrix} -360 \\ -287.0 \\ 705.6 \\ 191.7 \end{bmatrix} \tag{8}$$

Here we can see that with low precision of calculations the final result differs from the exact solution based on $n = 8$ mantissa digits. As mentioned before, the final answer in general will not be exactly equal to expected, since every mathematical operation involves an implicit conversion to decimal floating point numbers.

Now we introduce the *relative error* δ to estimate the difference between the exact solution and the numerical approximation, which uses an infinity norm [6]:

$$\delta = \frac{||\mathbf{X}_{Exact} - \mathbf{X}_{Numeric}||_\infty}{||\mathbf{X}_{Exact}||_\infty} \tag{9}$$

So, the relative error for Gaussian Elimination in system (7) is $\delta_{GE} = 1.138$. We perform these calculations using `error(exact, approx)` function, where *exact* is the exact solution, and *approx* is the numerical one.

Let's look at another example generated with `sys_random(4, 9)`:

$$\begin{bmatrix} 2592 & 324 & 0 & 23328 \\ 0 & 27 & 0 & 0 \\ 21 & 0 & -3 & 216 \\ 20736 & 3780 & 0 & 186652 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -10593504 \\ 6912 \\ -97998 \\ -84456560 \end{bmatrix} \tag{10}$$

After the first three applications of Gaussian Elimination method we have the following system (Attachment 6.2):

$$\left[\begin{array}{cccc|c} 2592 & 324 & 0 & 2.333E+4 & -1.059E+7 \\ 0 & 27 & 0 & 0 & 6912 \\ 0 & 0 & -3.000 & 27.00 & 1.153E+4 \\ 0 & 0 & 0 & 0.00 & -2.390E+4 \end{array}\right] \tag{11}$$

As we can see, the next application is impossible due to the potential division by zero, which we cannot avoid as there are no more rows to swap. Even though the pseudo-random matrix generator always produces invertible matrices, there is a chance that due to round-off errors we may end up in a situation where we cannot proceed with Gaussian Elimination because of the division by zero, so the system becomes unsolvable.

2.2 Partial Pivoting

It turns out that in some cases round-off errors can be significantly reduced if we make a guided choice of when and which rows of the augmented matrix to swap. A rule we use to choose the rows is called a *pivoting strategy.* In particular, the algorithm that we will discuss in this section is Gaussian Elimination with *Partial Pivoting* [3].

Consider the following example:

$$\left[\begin{array}{cc|c} 5 & 4 & 1 \\ -7 & 9 & 2 \end{array}\right] \tag{12}$$

Using classic Gaussian Elimination we are not paying special attention to the coefficients before x_n, but now we are going to allow for the possibility of swapping the rows to eliminate x_1 from the rows below. A widely used method is to use the row with the leading coefficient which is largest in absolute value. In this case, we would swap rows 1 and 2. Such an approach with the choice of the largest coefficient before x_n allows to significantly reduce the likelihood of adding or subtracting very small or very large numbers, and therefore reduce the round-off error [3, 12].

Consider the system (7), which we will solve once more, but now by using the Partial Pivoting method. The solution using `sys_solve_partial_pivoting()` for this system is (Attachment 6.3):

$$X_{numeric} = \begin{bmatrix} -133.3 \\ -267.1 \\ 709.7 \\ 192 \end{bmatrix} \tag{13}$$

Comparing this result with $X_{numeric}$ from `sys_solve_gaussian_elimination()` from (8), we see that this method gives the result that closer to the exact solution. The relative error also confirms this conclusion:

$$\delta_{PP} = 0.8226 \tag{14}$$

Another illustrative example that we can generate using `sys_random(5, 96)`:

$$\begin{bmatrix} -405 & 0 & 2288 & 330 & 66 \\ -16 & -16 & 0 & 0 & 0 \\ 0 & 0 & -1568 & 0 & 0 \\ -567 & 0 & -988 & -25 & -4 \\ 0 & 0 & 268 & 30 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 2175989 \\ 9536 \\ -1346912 \\ -720793 \\ 240484 \end{bmatrix} \tag{15}$$

Then the exact solution X_{exact}, Gaussian Elimination X_{GE}, and Partial Pivoting X_{PP} are as follows (Attachment 6.4):

$$X_{exact} = \begin{bmatrix} -241 \\ -355 \\ 859 \\ 380 \\ -188 \end{bmatrix}, \quad X_{GE} = \begin{bmatrix} -237.0 \\ -356.2 \\ 859.1 \\ 117.5 \\ 1129 \end{bmatrix} \quad X_{PP} = \begin{bmatrix} -242.7 \\ -353.3 \\ 859.0 \\ 515.0 \\ -865.5 \end{bmatrix} \tag{16}$$

With the following relative errors:

$$\begin{aligned} \delta_{GE} &= 1.533 \\ \delta_{PP} &= 0.7887 \end{aligned} \tag{17}$$

In this case, the limited precision of the decimal floating point numbers used to represent these values still causes great relative errors, but we can see that in general the error can be reduced by using different numerical methods for solving systems of equations.

2.3 Scaled Partial Pivoting

Despite the control of round-off error, Partial Pivoting can lead to a significant loss of precision when replacing row A_j with $A_j - m_{i,j} \times A_i$, which is significantly larger in magnitude. To resolve this problem, we require a modification of Partial Pivoting. First of all, we initialize two vectors:

1. Permutation vector $P = (1, 2, ..., n)$, where n — dimension of the matrix A
2. Scale factors vector $S = (\max_{j=1}^{n} |a_{i,j}|)_{i=1}^{n}$

Then we apply forward elimination: when eliminating elements in column j, we interchange row j with row p, where:

$$\frac{|a_{p,j}|}{s_p} = \max_{j \le i \le n} \frac{|a_{i,j}|}{s_i} \tag{18}$$

It is important to notice that the order of s_i in vector S also depends on the permutation vector, which changes with row swaps. For example, if rows A_1 and A_3 are swapped, then components of P p_1 and p_3 are swapped, as well as the components of S s_1 and s_3.

When all forward elimination steps are executed, and a matrix is reduced, we perform back substitution using reverse order of permutation vector P. These steps combined together represent a technique called *Scaled Partial Pivoting* and it is more effective than Partial Pivoting in reducing the round-off error [9, 4].

Coming back to the system (7), we compare all three numerical approximations with each other using their relative errors (Attachment 6.5):

$$\begin{aligned} \delta_{GE} &= 1.138 \\ \delta_{PP} &= 0.8226 \\ \delta_{SPP} &= 0.5615 \end{aligned} \tag{19}$$

We observe the same results when comparing relative errors for each approach in the system (15) (Attachment 6.6):

$$\begin{aligned} \delta_{GE} &= 1.533 \\ \delta_{PP} &= 0.7887 \\ \delta_{SPP} &= 0.009779 \end{aligned} \tag{20}$$

3 Iterative Methods for Solving SLAEs

3.1 Jacobi

With direct methods outlined above it possible to obtain solutions that are approximate to the exact solution only for linear systems of small size. In contrast, iterative methods produce an approximate solution to the linear system after a finite number of steps [14]. Iterative methods can be used with any matrix, but they are typically applied to large sparse matrices for which direct solves are slow [1].

Without a loss of generality, consider the following linear system:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases} \tag{21}$$

Which can be written as:

$$\begin{aligned} x_1 &= \tfrac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3), \\ x_2 &= \tfrac{1}{a_{22}}(b_2 - a_{21}x_1 - a_{23}x_3), \\ x_3 &= \tfrac{1}{a_{33}}(b_3 - a_{31}x_1 - a_{32}x_2) \end{aligned} \tag{22}$$

So, if the initial approximation is known:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} \tag{23}$$

then the new approximation can be calculated as:

$$\begin{aligned} x_1^{(1)} &= \tfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}), \\ x_2^{(1)} &= \tfrac{1}{a_{22}}(b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)}), \\ x_3^{(1)} &= \tfrac{1}{a_{33}}(b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(0)}) \end{aligned} \tag{24}$$

Next we can use $x^{(1)}$ to calculate a new approximation $x^{(2)}$ and so on, for a finite number of iterations. In general we have the formula for *Jacobi method* [10]:

$$\begin{aligned} x_1^{(n+1)} &= \tfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(n)} - a_{13}x_3^{(n)}), \\ x_2^{(n+1)} &= \tfrac{1}{a_{22}}(b_2 - a_{21}x_1^{(n)} - a_{23}x_3^{(n)}), \\ x_3^{(n+1)} &= \tfrac{1}{a_{33}}(b_3 - a_{31}x_1^{(n)} - a_{32}x_2^{(n)}) \end{aligned} \tag{25}$$

where $n = 0, 1, 2, \ldots$.

In matrix form:

$$\mathbf{x}^{(k)} = T_J \mathbf{x}^{(k-1)} + C_J \tag{26}$$

where

$$T_J = D^{-1}(-A + D) \;\; and \;\; C_J = D^{-1}\mathbf{b} \tag{27}$$

and D is the diagonal part of initial matrix A.

By applying iterative steps we expect our approximations to converge to the exact solution, so we employ a metric with an epsilon-threshold to check when our iterative approximations become "close to each other". In other words, we continue iterating until the following is true for some small-enough epsilon:

$$error(x^{(k)}, x^{(k-1)}) < \epsilon$$

It should be noted, that given a limited decimal floating point precision it is not guaranteed that some epsilon-threshold values can ever be reached, so we can run into issues where iterative methods stop converging with big relative error.

Jacobi method mainly relies on the requirement that for each $i, d_{ii} \equiv a_{ii} \neq 0$ [6, 14]. If this relationship does not hold for even a single value of i, then the equations of the system must be reordered before the Jacobi method can be applied, as otherwise D^{-1} does not exist.

We can compare the performance of Jacobi method to the direct method by generating a new system with `sys_random(4, 87)` (Attachment 6.7):

$$\begin{bmatrix} -432 & 0 & 0 & 9 \\ -2592 & 2 & 720 & 54 \\ 0 & 0 & 1620 & 0 \\ 0 & 0 & 240 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 309969 \\ 1420198 \\ -991440 \\ -147673 \end{bmatrix} \tag{28}$$

Then by using `sys_solve_iterative_jacobi()` with $epsilon = 0.001$ we get the following T_J and C_J matrices:

$$T_J = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1296 & 0.0 & -360.0 & -27.0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 240 & 0 \end{bmatrix}, \quad C_J = \begin{bmatrix} -717.6 \\ 7.101E+5 \\ -612.0 \\ 1.477E+5 \end{bmatrix} \tag{29}$$

Finally, comparing solutions X_{SPP}, X_J and their relative errors:

$$X_{SPP} = \begin{bmatrix} -698.8 \\ 3E+2 \\ -612.0 \\ 9E+2 \end{bmatrix} \quad \delta_{SPP} = 0.2673, \qquad X_J = \begin{bmatrix} -700.9 \\ 4E+2 \\ -612.0 \\ 8E+2 \end{bmatrix} \quad \delta_J = 0.1412 \tag{30}$$

we can observe that the iterative Jacobi method gives better results than the best direct method which is Scaled Partial Pivoting.

3.2 Gauss-Siedel

The Gauss-Siedel method expands on top of the Jacobi method. Without a loss of generality, we start with the equation (25):

$$\begin{aligned} x_1^{(n+1)} &= \tfrac{1}{a_{11}}(b_1 - a_{12}x_2^{(n)} - a_{13}x_3^{(n)}), \\ x_2^{(n+1)} &= \tfrac{1}{a_{22}}(b_2 - a_{21}x_1^{(n)} - a_{23}x_3^{(n)}), \\ x_3^{(n+1)} &= \tfrac{1}{a_{33}}(b_3 - a_{31}x_1^{(n)} - a_{32}x_2^{(n)}) \end{aligned}$$

where we can use a better approximation $x_1^{(n+1)}$ to calculate $x_2^{(n+1)}$ and we can use these values to calculate $x_3^{(n+1)}$. This modification is known as *Gauss-Siedel method* [11], and for $n = 0, 1, 2, \dots$ it has the form:

$$\begin{array}{ll} x_1^{(n+1)} &= \frac{1}{a_{11}}(b_1 - a_{12}x_2^{(n)} - a_{13}x_3^{(n)}), \\ x_2^{(n+1)} &= \frac{1}{a_{22}}(b_2 - a_{21}x_1^{(n+1)} - a_{23}x_3^{(n)}), \\ x_3^{(n+1)} &= \frac{1}{a_{33}}(b_3 - a_{31}x_1^{(n+1)} - a_{32}x_2^{(n+1)}) \end{array} \tag{31}$$

The matrix form of Gauss-Siedel method is the same as in equation (26), but there is a difference in the matrices T and C:

$$T_{GS} = L^{-1}(D - U) \;\; and \;\; C_{GS} = L^{-1}\mathbf{b} \tag{32}$$

We are re-solving the system (28), by using `sys_solve_iterative_gauss_seidel()` with $epsilon = 0.001$ to calculate T_{GS} and C_{GS} matrices (Attachment 6.8):

$$\begin{array}{l} T_{GS} = \begin{bmatrix} 0 & 0 & 0 & 0.02084 \\ 0 & 0 & -360.0 & 0.0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\ C_{GS} = \begin{bmatrix} -717.6 \\ -2.198E+5 \\ -612.0 \\ 8E+2 \end{bmatrix} \end{array} \tag{33}$$

And finally we compare exact, Jacobi and Gauss-Seidel solutions:

$$X_{exact} = \begin{bmatrix} -701 \\ 512 \\ -612 \\ 793 \end{bmatrix}, \quad X_J = \begin{bmatrix} -700.9 \\ 400 \\ -612.0 \\ 800 \end{bmatrix} X_{GS} = \begin{bmatrix} -700.9 \\ 500 \\ -612.0 \\ 800.0 \end{bmatrix} \tag{34}$$

Moreover, the relative error of Gauss-Seidel method is significantly less than Jacobi method:

$$\delta_{GS} = 0.01513 \tag{35}$$

In addition to low relative error, with the iterative approach we can save a lot of time for calculating the numerical answer [14, 1] for systems of larger dimension, assuming that the iterative steps converge.

4 Results

In our study, we focused on the problem of decimal machine numbers and round-off error due to their limited precision. We provided basic representations of these numbers according to existing standards and discussed a generally accepted rounding approach.

To understand the impact of round-off error, we considered systems of linear equations that are sensitive to this type of error and examined methods for reducing it. We studied Gaussian Elimination, Partial Pivoting, and Scaled Partial Pivoting as direct methods, and Jacobi and Gauss-Siedel methods as iterative ones. Our results show that iterative methods are preferable in terms of speed and relative error of the solution vector. Additionally, we found that the Gauss-Siedel method is particularly effective in reducing rounding error. Overall, our findings provide insight into the challenges and potential solutions for dealing with round-off error in the numerical analysis of systems of linear equations.

5 Acknowledgements

I extend my appreciation to professor Stephen Kirkland at University of Manitoba for instructing Linear Algebra and Matrix Analysis class and providing me with an opportunity to explore the topic of decimal machine number precision and their problems by myself.

6 Attachments

6.1 Example 1

```
System of 4 equations:
4x4 matrix:
25            0             -1440         0
0             -648          3240          0
150           -252          -7416         1200
30            0             -1728         240

Exact solution x:
1x4 matrix:
459           -215          720           194

Vector b:
```

```
1x4 matrix:
-1025325      2472120       -4983690      -1183830

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             -252          1224          1200          1.166E+6
0             0.0           0             240.0         4.6E+4

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             0             -36           1200          2.046E+5
0             0             0.0           240.0         4.60E+4

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             0             -36           1200          2.046E+5
0             0             0             240.0         4.600E+4

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             0             -36           1200          2.046E+5
0             0             0             240.0         4.600E+4

4x1 matrix:
-360
-287.0
705.6
191.7
```

6.2 Example 2

```
System of 4 equations:
4x4 matrix:
2592          324           0             23328
0             27            0             0
21            0             -3            216
20736         3780          0             186652

Exact solution x:
1x4 matrix:
-51           256           -235          -452

Vector b:
1x4 matrix:
-10593504     6912          -97998        -84456560

Next gaussian step:
4x5 matrix:
2592          324           0             2.333E+4      -1.059E+7
0             27            0             0             6912
0             -2.625        -3.000        27.0          -1.220E+4
0             1187          0.000         0E+2          2.8E+5

Next gaussian step:
4x5 matrix:
2592          324           0             2.333E+4      -1.059E+7
0             27            0             0             6912
0             0             -3.000        27.00         -1.153E+4
0             0             0.000         0.00          -2.39E+4

Next gaussian step:
4x5 matrix:
2592          324           0             2.333E+4      -1.059E+7
0             27            0             0             6912
0             0             -3.000        27.00         -1.153E+4
0             0             0             0.00          -2.390E+4
```

```
Exception: Potential division by 0 due to precision loss. Terminating.
```

6.3 Example 3

```
System of 4 equations:
4x4 matrix:
25            0             -1440         0
0             -648          3240          0
150           -252          -7416         1200
30            0             -1728         240

Exact solution x:
1x4 matrix:
459           -215          720           194

Vector b:
1x4 matrix:
-1025325      2472120       -4983690      -1183830

Next gaussian step:
4x5 matrix:
30            0             -1728         240           -1.184E+6
0             -648          3240          0             2.472E+6
0             -252          1224          0             9.36E+5
0             0.0000        0             -200.0        -3.84E+4

Next gaussian step:
4x5 matrix:
30            0             -1728         240           -1.184E+6
0             -252          1224          0             9.36E+5
0             0             93            0.000         6.6E+4
0             0             0.0000        -200.0        -3.840E+4

Next gaussian step:
4x5 matrix:
```

```
30            0             -1728         240           -1.184E+6
0             -252          1224          0             9.36E+5
0             0             93            0.000         6.6E+4
0             0             0             -200.0        -3.840E+4

Next gaussian step:
4x5 matrix:
30            0             -1728         240           -1.184E+6
0             -252          1224          0             9.36E+5
0             0             93            0.000         6.6E+4
0             0             0             -200.0        -3.840E+4

4x1 matrix:
-133.3
-267.1
709.7
192
```

6.4 Example 4

```
System of 5 equations:
5x5 matrix:
-405          0             2288          330           66
-16           -16           0             0             0
0             0             -1568         0             0
-567          0             -988          -25           -4
0             0             268           30            6

Exact solution x:
1x5 matrix:
-241          -355          859           380           -188

Vector b:
1x5 matrix:
2175989       9536          -1346912      -720793       240484
```

```
Next gaussian step:
5x6 matrix:
-405          0             2288          330           66            2.176E+6
0             -16.00        -90.40        -13.04        -2.608        -7.643E+4
0             0             -1568         0             0             -1.347E+6
0             0.0           -4191         -487.0        -96.4         -3.767E+6
0             0             268           30            6             2.405E+5

Next gaussian step:
5x6 matrix:
-405          0             2288          330           66            2.176E+6
0             -16.00        -90.40        -13.04        -2.608        -7.643E+4
0             0             -1568         0             0.0           -1.347E+6
0             0             -4191         -487.0        -96.40        -3.767E+6
0             0             268           30            6.0           2.405E+5

Next gaussian step:
5x6 matrix:
-405          0             2288          330           66            2.176E+6
0             -16.00        -90.40        -13.04        -2.608        -7.643E+4
0             0             -1568         0             0.0           -1.347E+6
0             0             0             -487.0        -96.40        -1.66E+5
0             0             0             30.00         6.000         1.03E+4

Next gaussian step:
5x6 matrix:
-405          0             2288          330           66            2.176E+6
0             -16.00        -90.40        -13.04        -2.608        -7.643E+4
0             0             -1568         0             0.0           -1.347E+6
0             0             0             -487.0        -96.40        -1.66E+5
0             0             0             0             0.062         7E+1

Next gaussian step:
5x6 matrix:
-405          0             2288          330           66            2.176E+6
0             -16.00        -90.40        -13.04        -2.608        -7.643E+4
```

```
0             0             -1568         0             0.0           -1.347E+6
0             0             0             -487.0        -96.40        -1.66E+5
0             0             0             0             0.062         7E+1

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             405.0         2288          330.0         66.00         1.935E+6
0             0             -1568         0             0             -1.347E+6
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30            6             2.405E+5

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             -1568         0.0           0.0           -1.347E+6
0             0             2994          347.9         68.86         2.691E+6
0             0             268           30.0          6.0           2.405E+5

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30.0          6.0           2.405E+5
0             0             0             12.8          1.84          5E+3
0             0             0             175.5         35.11         6.0E+4

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30.0          6.0           2.405E+5
0             0             0             175.5         35.11         6.0E+4
0             0             0             0             -0.721        624
```

```
Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30.0          6.0           2.405E+5
0             0             0             175.5         35.11         6.0E+4
0             0             0             0             -0.721        624

5x1 matrix:
-237.0
-356.2
859.1
117.5
1129

5x1 matrix:
-242.7
-353.3
859.0
515.0
-865.5
```

6.5 Example 5

```
System of 4 equations:
4x4 matrix:
25            0             -1440         0
0             -648          3240          0
150           -252          -7416         1200
30            0             -1728         240

Exact solution x:
1x4 matrix:
459           -215          720           194

Vector b:
```

```
1x4 matrix:
-1025325      2472120       -4983690      -1183830

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             -252          1224          1200          1.166E+6
0             0.0           0             240.0         4.6E+4

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             0             -36           1200          2.046E+5
0             0             0.0           240.0         4.60E+4

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             0             -36           1200          2.046E+5
0             0             0             240.0         4.600E+4

Next gaussian step:
4x5 matrix:
25            0             -1440         0             -1.025E+6
0             -648          3240          0             2.472E+6
0             0             -36           1200          2.046E+5
0             0             0             240.0         4.600E+4

Next gaussian step:
4x5 matrix:
30            0             -1728         240           -1.184E+6
0             -648          3240          0             2.472E+6
0             -252          1224          0             9.36E+5
```

```
0              0.0000         0              -200.0         -3.84E+4

Next gaussian step:
4x5 matrix:
30             0              -1728          240            -1.184E+6
0              -252           1224           0              9.36E+5
0              0              93             0.000          6.6E+4
0              0              0.0000         -200.0         -3.840E+4

Next gaussian step:
4x5 matrix:
30             0              -1728          240            -1.184E+6
0              -252           1224           0              9.36E+5
0              0              93             0.000          6.6E+4
0              0              0              -200.0         -3.840E+4

Next gaussian step:
4x5 matrix:
30             0              -1728          240            -1.184E+6
0              -252           1224           0              9.36E+5
0              0              93             0.000          6.6E+4
0              0              0              -200.0         -3.840E+4

4x5 matrix:
150            -252           -7416          1200           -4.984E+6
0              -648           3240           0              2.472E+6
25             0              -1440          0              -1.025E+6
30             0              -1728          240            -1.184E+6

Next gaussian step:
4x5 matrix:
150            -252           -7416          1200           -4.984E+6
0              -648           3240           0              2.472E+6
0              42.01          -204           -200.0         -1.942E+5
0              50.4           -245           0.0            -1.872E+5
```

```
Next gaussian step:
4x5 matrix:
150           -252          -7416         1200          -4.984E+6
0             -648          3240          0             2.472E+6
0             0             6.0           -200.0        -3.39E+4
0             0             7.0           0.00000       5.1E+3

Next gaussian step:
4x5 matrix:
150           -252          -7416         1200          -4.984E+6
0             -648          3240          0             2.472E+6
0             0             6.0           -200.0        -3.39E+4
0             0             0             233.4         4.466E+4

Next gaussian step:
4x5 matrix:
150           -252          -7416         1200          -4.984E+6
0             -648          3240          0             2.472E+6
0             0             6.0           -200.0        -3.39E+4
0             0             0             233.4         4.466E+4

4x1 matrix:
-360
-287.0
705.6
191.7

4x1 matrix:
-133.3
-267.1
709.7
192

4x1 matrix:
863.3
-180.6
```

```
726.7
191.3
```

6.6 Example 6

```
System of 5 equations:
5x5 matrix:
-405          0             2288          330           66
-16           -16           0             0             0
0             0             -1568         0             0
-567          0             -988          -25           -4
0             0             268           30            6

Exact solution x:
1x5 matrix:
-241          -355          859           380           -188

Vector b:
1x5 matrix:
2175989       9536          -1346912      -720793       240484

Next gaussian step:
5x6 matrix:
-405          0             2288          330           66            2.176E+6
0             -16.00        -90.40        -13.04        -2.608        -7.643E+4
0             0             -1568         0             0             -1.347E+6
0             0.0           -4191         -487.0        -96.4         -3.767E+6
0             0             268           30            6             2.405E+5

Next gaussian step:
5x6 matrix:
-405          0             2288          330           66            2.176E+6
0             -16.00        -90.40        -13.04        -2.608        -7.643E+4
0             0             -1568         0             0.0           -1.347E+6
0             0             -4191         -487.0        -96.40        -3.767E+6
0             0             268           30            6.0           2.405E+5
```

```
Next gaussian step:
5x6 matrix:
-405         0            2288         330          66           2.176E+6
0            -16.00       -90.40       -13.04       -2.608       -7.643E+4
0            0            -1568        0            0.0          -1.347E+6
0            0            0            -487.0       -96.40       -1.66E+5
0            0            0            30.00        6.000        1.03E+4

Next gaussian step:
5x6 matrix:
-405         0            2288         330          66           2.176E+6
0            -16.00       -90.40       -13.04       -2.608       -7.643E+4
0            0            -1568        0            0.0          -1.347E+6
0            0            0            -487.0       -96.40       -1.66E+5
0            0            0            0            0.062        7E+1

Next gaussian step:
5x6 matrix:
-405         0            2288         330          66           2.176E+6
0            -16.00       -90.40       -13.04       -2.608       -7.643E+4
0            0            -1568        0            0.0          -1.347E+6
0            0            0            -487.0       -96.40       -1.66E+5
0            0            0            0            0.062        7E+1

Next gaussian step:
5x6 matrix:
-16          -16          0            0            0            9536
0            405.0        2288         330.0        66.00        1.935E+6
0            0            -1568        0            0            -1.347E+6
0            567.0        -988.0       -25.00       -4.00        -1.059E+6
0            0            268          30           6            2.405E+5

Next gaussian step:
5x6 matrix:
-16          -16          0            0            0            9536
```

```
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             -1568         0.0           0.0           -1.347E+6
0             0             2994          347.9         68.86         2.691E+6
0             0             268           30.0          6.0           2.405E+5

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30.0          6.0           2.405E+5
0             0             0             12.8          1.84          5E+3
0             0             0             175.5         35.11         6.0E+4

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30.0          6.0           2.405E+5
0             0             0             175.5         35.11         6.0E+4
0             0             0             0             -0.721        624

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30.0          6.0           2.405E+5
0             0             0             175.5         35.11         6.0E+4
0             0             0             0             -0.721        624

5x6 matrix:
-16           -16           0             0             0             9536
-405          0             2288          330           66            2.176E+6
0             0             -1568         0             0             -1.347E+6
-567          0             -988          -25           -4            -7.208E+5
0             0             268           30            6             2.405E+5
```

```
Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             405.0         2288          330.0         66.00         1.935E+6
0             0             -1568         0             0             -1.347E+6
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             268           30            6             2.405E+5

5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             -1568         0             0             -1.347E+6
0             405.0         2288          330.0         66.00         1.935E+6
0             0             268           30            6             2.405E+5

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             -1568         0.0           0.0           -1.347E+6
0             0             2994          347.9         68.86         2.691E+6
0             0             268           30.0          6.0           2.405E+5

5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             2994          347.9         68.86         2.691E+6
0             0             -1568         0.0           0.0           -1.347E+6
0             0             268           30.0          6.0           2.405E+5

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             2994          347.9         68.86         2.691E+6
0             0             0             182.2         36.06         6.2E+4
```

```
0             0             0             -1.14         -0.164        -4E+2

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             2994          347.9         68.86         2.691E+6
0             0             0             182.2         36.06         6.2E+4
0             0             0             0             0.0616        -12.1

Next gaussian step:
5x6 matrix:
-16           -16           0             0             0             9536
0             567.0         -988.0        -25.00        -4.00         -1.059E+6
0             0             2994          347.9         68.86         2.691E+6
0             0             0             182.2         36.06         6.2E+4
0             0             0             0             0.0616        -12.1

5x1 matrix:
-237.0
-356.2
859.1
117.5
1129

5x1 matrix:
-242.7
-353.3
859.0
515.0
-865.5

5x1 matrix:
-239.9
-356.1
859.4
```

```
379.1
-196.4
```

6.7 Example 7

```
System of 4 equations:
4x4 matrix:
-432          0             0             9
-2592         2             720           54
0             0             1620          0
0             0             240           -1

Exact solution x:
1x4 matrix:
-701          512           -612          793

Vector b:
1x4 matrix:
309969        1420198       -991440       -147673

Next gaussian step:
4x5 matrix:
-432          0             0             9             3.100E+5
0             2             720           0             -4.40E+5
0             0             1620          0             -9.914E+5
0             0             240           -1            -1.477E+5

Next gaussian step:
4x5 matrix:
-432          0             0             9             3.100E+5
0             2             720           0             -4.40E+5
0             0             1620          0             -9.914E+5
0             0             240           -1            -1.477E+5

Next gaussian step:
4x5 matrix:
```

```
-432          0             0             9             3.100E+5
0             2             720           0             -4.40E+5
0             0             1620          0             -9.914E+5
0             0             0             -1.000        -9E+2

Next gaussian step:
4x5 matrix:
-432          0             0             9             3.100E+5
0             2             720           0             -4.40E+5
0             0             1620          0             -9.914E+5
0             0             0             -1.000        -9E+2

T is:
4x4 matrix:
0.000000      0.000000      0.000000      0.02084
1296          0.0           -360.0        -27.0
0E-7          0E-7          0E-7          0E-7
0             0             240           0

C is:
4x1 matrix:
-717.6
7.101E+5
-612.0
1.477E+5

4x1 matrix:
-698.8
3E+2
-612.0
9E+2

4x1 matrix:
-700.9
4E+2
-612.0
```

```
8E+2
```

6.8 Example 8

```
System of 4 equations:
4x4 matrix:
-432          0             0             9
-2592         2             720           54
0             0             1620          0
0             0             240           -1

Exact solution x:
1x4 matrix:
-701          512           -612          793

Vector b:
1x4 matrix:
309969        1420198       -991440       -147673

T is:
4x4 matrix:
0.000000      0.000000      0.000000      0.02084
0.000         0.000         -360.0        0.00
0E-13         0E-13         0E-13         0E-13
0E-13         0E-13         0E-13         0E-13

C is:
4x1 matrix:
-717.6
-2.198E+5
-612.0
8E+2

4x1 matrix:
-700.9
5E+2
```

```
-612.0
800.0

0.01513
```

6.9 Code

Full source code is available at: `https://github.com/HaselLoyance/MATH4370-project`.

7 Bibliography

[1] Owe Axelsson. “A class of iterative methods for finite element equations”. In: *Computer Methods in Applied Mechanics and Engineering* 9.2 (1976), pp. 123–137.

[2] M.F. Cowlishaw. “Decimal floating-point: algorism for computers”. In: *Proceedings 2003 16th IEEE Symposium on Computer Arithmetic.* 2003, pp. 104–111. DOI: `10.1109/ARITH.2003.1207666`.

[3] Alan George and Esmond Ng. “An implementation of Gaussian elimination with partial pivoting for sparse systems”. In: *SIAM journal on scientific and statistical computing* 6.2 (1985), pp. 390–409.

[4] Israel Gohberg, Thomas Kailath, and Vadim Olshevsky. “Fast Gaussian elimination with partial pivoting for matrices with displacement structure”. In: *Mathematics of computation* 64.212 (1995), pp. 1557–1576.

[5] David Goldberg. “What Every Computer Scientist Should Know about Floating-Point Arithmetic”. In: *ACM Comput. Surv.* 23.1 (Mar. 1991), pp. 5–48. ISSN: 0360-0300. DOI: `10.1145/103162.103163`. URL: `https://doi.org/10.1145/103162.103163`.

[6] Gene H Golub and Gérard Meurant. “Matrices, moments and quadrature II; how to compute the norm of the error in iterative methods”. In: *BIT Numerical Mathematics* 37.3 (1997), pp. 687–705.

[7] Nicholas J. Higham. “The Accuracy of Floating Point Summation”. In: *SIAM Journal on Scientific Computing* 14.4 (1993), pp. 783–799. DOI: `10.1137/0914050`. eprint: `https://doi.org/10.1137/0914050`. URL: `https://doi.org/10.1137/0914050`.

[8] Clive Maxfield. “An introduction to different rounding algorithms”. In: *Programmable Logic Design Line* (2006), pp. 1–15.

[9] Juan Manuel Peña. “Scaled pivots and scaled partial pivoting strategies”. In: *SIAM journal on numerical analysis* 41.3 (2003), pp. 1022–1031.

[10] Heinz Rutishauser. “The Jacobi method for real symmetric matrices”. In: *Numerische Mathematik* 9.1 (1966), pp. 1–10.

[11] Masataka Usui, Hiroshi Niki, and Toshiyuki Kohno. “Adaptive Gauss-Seidel method for linear systems”. In: *International Journal of Computer Mathematics* 51.1-2 (1994), pp. 119–125.

[12] Ashu Vij. “Pivoting Strategies and Their Applications for Solving Linear Systems”. In: ().

[13] J. H. Wilkinson. “Error Analysis of Direct Methods of Matrix Inversion”. In: *J. ACM* 8.3 (July 1961), pp. 281–330. ISSN: 0004-5411. DOI: `10.1145/321075.321076`. URL: `https://doi.org/10.1145/321075.321076`.

[14] David M Young. “A historical overview of iterative methods”. In: *Computer Physics Communications* 53.1-3 (1989), pp. 1–17.