An Analysis on Direct and Iterative Techniques to Solve Linear Systems

Ratislav Krylov Caleb Lewis

April 17, 2018

Abstract

The need to solve the problem $Ax = b$ comes up in many real-world scenarios. As a result, there have been many solutions of varying efficiencies that people have employed to solve it. In this paper, we look at 9 of these methods and compare their complexity and accuracy.

1 Introduction

A section briefly introduces the background and purpose of this paper.

2 Examples

In our analysis of each method, we will be using 4 sample matrices. These samples represent different conditions that will affect how well each method will perform.

$$A = \begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} 4 & 1 & -3 & 4 \end{bmatrix}, \quad x = \begin{bmatrix} -1 & 2 & 0 & 1 \end{bmatrix} \tag{1}$$

Note example 1 is not diagnolly dominant.

$$A = \begin{bmatrix} .2 & .1 & 1 & 1 & 0 \\ .1 & 4 & -1 & 1 & -1 \\ 1 & -1 & 60 & 0 & -1 \\ 1 & 1 & 0 & 8 & 4 \\ 0 & -1 & -2 & 4 & 700 \end{bmatrix}, \quad b = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix}, x = \begin{bmatrix} 7.859713071 & 0.4229264082 & -.07359223906 & -.540643 \end{bmatrix} \tag{2}$$

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 24 & 30 & -24 \end{bmatrix}, \quad x = \begin{bmatrix} 3 & 4 & -5 \end{bmatrix} \tag{3}$$

$$A = \begin{bmatrix} 3.3330 & 15920 & -10.333 \\ 2.2220 & 16.710 & 9.6120 \\ 1.5611 & 5.1791 & 1.6852 \end{bmatrix}, \quad b = \begin{bmatrix} 15913 & 28.544 & 8.4254 \end{bmatrix}, \quad x = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{4}$$

3 Methods

3.1 Gaussian Elimination

Gaussian Elimination is one of the oldest methods used to solve systems of linear equations. It utilizes elementary row operations to convert the augmented system into an upper triangular matrix and then solve for each unknown x through backwards substitution. In theory, Gaussian Elimination finds exact values for the system of linear equations, but due to physical limitations, its not really used in practice. There is always rounding in numerical computations and Gaussian Elimination is quite susceptible to them. Additionally its computational of $O(n^3)$ is quite high and makes it impossible to use for large matrices.

3.1.1 Partial Pivoting

One way

3.1.2 Scaled Partial Pivoting

stuff abuot scaled partial Pivoting

3.2 Jacobi Iterative

The first of the iterative methods that we consider is the Jacobi Iterative method. For a diagnolly dominant matrix A,

$$\text{let} \quad x^{(0)} \in \mathbb{R}^n, \quad D = diag(A), \quad R = A - D \tag{5}$$

$$x^{k+1} = D^{-1}(b - Rx^{(k)}) \tag{6}$$

We repeat the above equation for $k = 0, 1, ...$ until convergence. The stopping critia is:

$$\frac{\left\|x^{(k)} - x^{(k-1)}\right\|}{x^{(k)}} \leq \epsilon \tag{7}$$

For our experiments, we set $\epsilon = .001$ as we found it was enough to show the differences in accuracies and behaviors between this and other methods. The following is a table of the results after running the method on the examples for a various number of iterations:

Example 2	10 0.52058755	20 0.17910029	30 0.20600456	40 0.02831351
Example 3	10 0.46132609	20 0.04399548	30 .00419573634	40 .000400136598

Notice that examples 1 and 4 have been left out - those are not diagnolly dominant, therefore the Jacobi method is not able to converge.

3.3 Gauss-Seidel

Next is the Gauss-Seidel method. This method has similar limitations as the Jacobi method, but performs better in practice. First we define L to be the lower triangle part of A and:

$$let x^{(0)} \in \mathbb{R}^n, \quad U = A - L \tag{8}$$

the method is the iteration

$$x^{k+1} = L^{-1}(b - Ux^{(k)}) \tag{9}$$

for $k = 0, 1, ...$ until the convergence or stopping criteria. We will use the same criteria as we did in the Jacobi method.

Example 2	10 0.1327548	20 .00326896	30 .00292537	40 .003067423
Example 3	10 0.02302011	20 .00033498	30 .00000190	40 .00000002

Examples 1 and 4 have been left out due to the same reasons as the Jacobi method - this method does not converge on matrices that are not diagnolly dominant. Also notice that on example 2, the method does not get more accurate after iteration 20 - it seems to only oscilate without improving.

3.4 Successive Over-Relaxation

stuff abuot Successive Over-Relaxation

3.5 Iterative Refinement

stuff abuot Iterative Refinement

3.6 (Preconditioned) Conjugate Gradient Method

stuff abuot (Preconditioned) Conjugate Gradient Method

4 Numerical experiments

You will need to demonstrate the performance of the methods on several (ideally 3 to 5) example IVPs (of your own choice). You can choose some problems from textbook, but make sure that you explicitly state what the problem you chose for each test.

To show the performance, it is often better to use figures rather than tables (unless there are very few numbers to show). For example, you can show the result of RK4 using Figure If you have multiple results, you can plot each with a curve (in different color/line-style/marker type) in the plot. If they are too close, you can consider to plot $|w_i - y_i|$, the error of estimate w_i to true solution $y_i = y(t_i)$, instead of actual y_i and w_i. This way, you can see which methods have lower errors (higher accuracy).

5 Discussion

This is a major part for this project. It should constitute your findings and thoughts. Based on the tests you have, you want to comment on the performance of these methods and how would you suggest to use in practice. Have extensive discussions with your team members and give detailed reasonings for your claims. You can cite books, papers, or other

6 Summary

A quick summary to conclude the term paper using a paragraph or two.