SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS

Naive Gaussian Elimination Method

2006 *Jamie Trahan , Autar Kaw, Kevin Martin*
University of South Florida
United States of America
kaw@eng.usf.edu
http://numericalmethods.eng.usf.edu

Introduction

One of the most popular numerical techniques for solving simultaneous linear equations Naive Gaussian Elimination method. The approach is designed to solve a set of *n* equations with *n* unknowns, [A] [X] = [C], where $[A]_{nxn}$ is a square coefficient matrix, $[X]_{nx1}$ is the solution vector, and $[C]_{nx1}$ is the right hand side array.

Naive Gauss consists of two steps:

1) **Forward Elimination**: *In this step, the unknown is eliminated in each equation starting with the first equation. This way, the equations are "reduced" to one equation and one unknown in each equation.*

2) **Back Substitution**: *In this step, starting from the last equation, each of the unknowns is found.*

To learn more about Naive Gauss Elimination as well as the pitfalls of the method, clicl **here.**

A simulation of Naive Gauss Method follows.

Section 1: Input

Below are the input parameters to begin the simulation. This is the only section that requires user input. The user can change the values that are highlighted and Mathcad v calculate the solution vector [X].

ORIGIN := 1

- $[A]_{nxn}$ coefficient matrix

$$A := \begin{pmatrix} 1 & 10 & 100 & 1000 \\ 1 & 15 & 225 & 3375 \\ 1 & 20 & 400 & 8000 \\ 1 & 22.5 & 506.25 & 11391 \end{pmatrix}$$

- $[RHS]_{nx1}$ right hand side array

$$RHS := \begin{pmatrix} 227.04 \\ 362.78 \\ 517.35 \\ 602.97 \end{pmatrix}$$

- Number of equations

n := rows(A)

n = 4

Section 2: Naive Gaussian Elimination method

This section divides Naive Gaussian Elimination into two steps:
1) Forward Elimination
2) Back Substitution

To conduct Naive Gaussian Elimination, Mathcad will augment the [A] and [RHS] mat into one matrix, [C], that will facilitate the process of forward elimination.

$$C := \text{augment}(A, RHS)$$

$$C = \begin{pmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 1 & 15 & 225 & 3375 & 362.78 \\ 1 & 20 & 400 & 8000 & 517.35 \\ 1 & 22.5 & 506.25 & 11391 & 602.97 \end{pmatrix}$$

2.1: Forward Elimination

Forward elimination of unknowns consists of (n-1) steps. In each step k, the coefficien of the k^{th} unknown will be zeroed from every subsequent equation that follows the k^{th} row. For example, in step 2 (i.e. k=2), the coefficient of x_2 will be zeroed from rows 3..n. With each step that is conducted, a new matrix is generated until the coefficient matrix is transformed to an upper triangular matrix. The following procedure calculate the upper triangular matrix while demonstrating the intermediate coefficient matrices t are produced for each step k.

The following procedure conducts (n-1), or k, steps of forward elimination.

$$\text{forward\_elimination}(step) := \left| \begin{array}{l} C \leftarrow C \\ \text{for } k \in 1 .. n - 1 \\ \quad \text{for } i \in (k + 1) .. n \\ \quad\quad \left| \begin{array}{l} \text{multiplier} \leftarrow \dfrac{C_{i,k}}{C_{k,k}} \\ \text{for } j \in k .. n + 1 \\ \quad C_{i,j} \leftarrow C_{i,j} - \text{multiplier} \cdot C_{k,j} \\ U_k \leftarrow C \end{array} \right. \\ U_{step} \end{array} \right.$$

Conducting (n-1) steps of forward elimination.

Defining row to be transformed.

Calculating multiplier value.

Defining column elements.

Generating rows of upper triangular matrix [U].

Assigning [U] matrix to the k^{th} step.

Returning [U] of k^{th} step.

Defining the number of steps:

$$step := 1 .. n - 1$$

The following array stores the coefficient matrix that was generated for each step of forward elimination, the last matrix being an upper triangular coefficient matrix augme with the newly transformed right hand side array :

$$\text{forward\_elimination}(step) = \begin{bmatrix} \begin{pmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 0 & 5 & 125 & 2375 & 135.74 \\ 0 & 10 & 300 & 7000 & 290.31 \\ 0 & 12.5 & 406.25 & 10391 & 375.93 \end{pmatrix} \\ \begin{pmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 0 & 5 & 125 & 2375 & 135.74 \\ 0 & 0 & 50 & 2250 & 18.83 \\ 0 & 0 & 93.75 & 4453.5 & 36.58 \end{pmatrix} \\ \begin{pmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 0 & 5 & 125 & 2375 & 135.74 \\ 0 & 0 & 50 & 2250 & 18.83 \\ 0 & 0 & 0 & 234.75 & 1.2737 \end{pmatrix} \end{bmatrix}$$

The upper triangular matrix and new right hand side array can now be extracted from the augmented matrix of the final step of forward elimination.

$$\text{upper\_triangular} := \text{submatrix}(\text{forward\_elimination}(n-1), 1, n, 1, n)$$

$$\text{upper\_triangular} = \begin{pmatrix} 1 & 10 & 100 & 1000 \\ 0 & 5 & 125 & 2375 \\ 0 & 0 & 50 & 2250 \\ 0 & 0 & 0 & 234.75 \end{pmatrix}$$

$$\text{RHSnew} := \text{submatrix}(\text{forward\_elimination}(n-1), 1, n, n+1, n+1)$$

$$\text{RHSnew} = \begin{pmatrix} 227.04 \\ 135.74 \\ 18.83 \\ 1.2737 \end{pmatrix}$$

This is the end of the forward elimination steps. Notice that the final row in the upper triangular matrix has only one unknown to be solved for. The new upper triangular coeff matrix and right hand side array permit solving for the solution vector using backward substitution.

2.2 Back Substitution

Back substitution begins with solving the last equation as it has only one unknown.

$$x_n = \left| \frac{rhs_n}{a_n} \right| \qquad \text{Equation (2.1)}$$

The remaining equations can be solved for using the following formula:

$$x_i = \left| \frac{c_i - \sum_{j=i+1}^{n} \left[a_{(i,j)} \cdot x_j \right]}{a_{i,i}} \right| \qquad \text{Equation (2.2)}$$

The procedure below calculates the solution vector using back substitution.

back_substitution :=

a ← upper_triangular — Renaming the new coefficient matrix.

rhs ← RHSnew — Renaming new right hand side array.

$x_n \leftarrow \frac{rhs_n}{a_{n,n}}$ — Solving for the n^{th} equation as it has only one unknown.

for $i \in n-1, n-2 .. 1$ — Defining remaining rows whose unknowns will be solved for working backwards.

  sum ← 0

  for $j \in (i+1) .. n$ — Defining column elements.

    $sum \leftarrow sum + a_{i,j} \cdot x_j$ — Calculating summation term in Eq. (2.2).

  $x_i \leftarrow \frac{rhs_i - sum}{a_{i,i}}$ — Using Eq. (2.2) to solve for x_i

x — Returning [X].

The solution vector for the system of equations using Naive Gaussian Elimination metho

X := back_substitution

$$X = \begin{pmatrix} -4.228 \\ 21.2599 \\ 0.1324 \\ 0.0054 \end{pmatrix}$$

Section 2: Exact Solution

The exact solution to the system of linear equations can be found using Mathcad's buil tools.

$$\text{exactsoln} := \text{lsolve}(A, RHS)$$

$$\text{exactsoln} = \begin{pmatrix} -4.228 \\ 21.2599 \\ 0.1324 \\ 0.0054 \end{pmatrix}$$

References

Autar Kaw, *Holistic Numerical Methods Institute, http://numericalmethods.eng.usf.edu/mws,* See
How does Gaussian Elimination work?
Effects of Significant Digits on solution of equations.

Conclusion

Mathcad helped us apply our knowledge of Naive Gaussian Elimination method to solve system of *n* simultaneous linear equation.

Question 1: The velocity of a rocket is given at three different times:

$$i := 1 .. 3$$

$\text{time}_i :=$ $\quad$ $\text{velocity}_i :=$

time_i	velocity_i
5sec	$106.8 \frac{m}{s}$
8sec	$177.2 \frac{m}{s}$
12sec	$279.2 \frac{m}{s}$

The velocity data is approximated by a polynomial as

$$v(t) = a_1 \cdot t^2 + a_2 \cdot t + a_3 \quad , \qquad 5 \le t \le 12$$

The coefficients a_1, a_2, a_3 for the above expression were found to be given by

$$\begin{pmatrix} 25 & 5 & 1 \\ 64 & 8 & 1 \\ 144 & 12 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 106.8 \\ 177.2 \\ 279.2 \end{pmatrix}$$

The coefficients a_1, a_2, a_3 using Naive Gauss Elimination. Find the velocity at t=6,7.5,9,11 seconds.

Question 2: Choose a set of equations that has a unique solution but for which the Naive Gauss Elimination method fails.