Naïve Gaussian Elimination

Jamie Trahan, Autar Kaw, Kevin Martin

University of South Florida

United States of America

kaw@eng.usf.edu

Introduction

One of the most popular numerical techniques for solving simultaneous linear equations is Naïve 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.
Naïve 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 Naïve Gauss Elimination as well as the pitfall's of the method, click here.

A simulation of Naive Gauss Method follows.

Section 1: Input Data

Below are the input parameters to begin the simulation. This is the only section that requires user input. Once the values are entered, *Mathematica* will calculate the solution vector [X].

- Number of equations, *n:*

```
n = 4
4
```

- *nxn* coefficient matrix, [A]:

```
A = Table[{{1, 10, 100, 1000}, {1, 15, 225, 3375},
   {1, 20, 400, 8000}, {1, 22.5, 506.25, 11391}}]; A // MatrixForm
```

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

- *n*x1 right hand side array, [RHS]:

```
RHS = Table[{227.04, 362.78, 517.35, 602.97}]; RHS // MatrixForm
```

$$\begin{pmatrix} 227.04 \\ 362.78 \\ 517.35 \\ 602.97 \end{pmatrix}$$

Section 2: Naïve Gaussian Elimination Method

The following sections divide Naïve Gauss elimination into two steps:
1) Forward Elimination
2) Back Substitution

To conduct Naïve Gauss Elimination, *Mathematica* will join the [A] and [RHS] matrices into one augmented matrix, [C], that will facilitate the process of forward elimination.

```
B = Transpose[Append[Transpose[A], RHS]]; B // MatrixForm
```

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

■ 2.1 Forward Elimination

Forward elimination of unknowns consists of (n-1) steps. In each step k, the coefficient 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 calculates the upper triangular matrix produced for each step k.

```
Print["Start", B // MatrixForm]; Print[" "]
For[k = 1, k ≤ n - 1, k++,
  For[i = k + 1, i <= n, i++,
  multiplier = N[B[[i, k]] / B[[k, k]]];
    For[j = k, j ≤ n + 1, j++, B[[i, j]] = B[[i, j]] - multiplier * B[[k, j]]]];
 Print["Step=", k, B // MatrixForm]; Print[" "]]
```

Start

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

Step=1

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

Step=2

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

Step=3

1	10	100	1000	227.04
0.	5.	125.	2375.	135.74
0.	0.	50.	2250.	18.83
0.	0.	0.	234.75	1.27375

The new upper triangular coefficient matrix, [A1], can be extracted from the final augmented matrix [B]:

```
A1 = Take[B, {1, n}, {1, n}]; MatrixForm[A1]
```

1	10	100	1000
0.	5.	125.	2375.
0.	0.	50.	2250.
0.	0.	0.	234.75

Notice that the final row, *n*, has only one unknown to be solved for.

The new right hand side array, [RHS1], is:

```
RHS1 = Take[B, {1, n}, {n + 1, n + 1}]; MatrixForm[RHS1]
```

227.04
135.74
18.83
1.27375

This is the end of the forward elimination steps. The new upper triangular coefficient 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.

$$x_n = \frac{\text{rhs}_n}{a_n}$$

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

$$x_i = \frac{(c_i - \sum_{j=i+1}^{n} a_{i,j}\, x_j)}{a_{i,i}}$$

The procedure below calculates the solution vector using back substitution.

Defining the [X] vector:

```
X = Array[x, {n, 1}];
```

Solving for the n^{th}equation as it has only one unknown:

```
X[[n]] = RHS1[[n]] / A1[[n, n]]
```

```
{0.00542599}
```

Solving for the remaining (n-1) unknowns working backwards from the $(n-1)^{\text{th}}$equation to the first equation:

```
Do[summ = Sum[A1[[i, j]] * X[[j]], {j, i+1, n}];
 X[[i]] = (RHS1[[i]] - summ) / A1[[i, i]], {i, n - 1, 1, -1}]
```

The solution vector [X] is

```
X // MatrixForm
```

$$\begin{pmatrix} -4.22796 \\ 21.2599 \\ 0.132431 \\ 0.00542599 \end{pmatrix}$$

Section 3: Exact Solution

Using *Mathematica's* built-in tools, the exact solution is given below.

```
exactsoln = LinearSolve[A, RHS]; MatrixForm[exactsoln]
```

$$\begin{pmatrix} -4.22796 \\ 21.2599 \\ 0.132431 \\ 0.00542599 \end{pmatrix}$$

Conclusion

Mathematica helped us apply our knowledge of Naïve Gaussian Elimination method to solve a system of *n* simultaneous linear equations.

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

time	velocity
5 sec	106.8 (m / s)
8 sec	177.2 (m / s)
12 sec	279.2 (m / s)

The velocity data is approximated by a polynomial as
$v(t) = a_1 t^2 + a_2 t + a_3, 5 \leq t \leq 12$

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

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

Find the values of a_1, a_2, a_3 using Naïve Gaussian 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 Naïve Gauss Elimination method fails.

References

[1] 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.