Naïve Gaussian Elimination Method

©2006 Kevin Martin, Autar Kaw, Jamie Trahan

University of South Florida

United States of America

kaw@eng.usf.edu

NOTE: This worksheet demonstrates the use of Maple to illustrate Naïve Gaussian Elimination, a numerical technique used in solving a system of simultaneous linear equations.

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 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 Gaussian Elimination as well as the pitfalls of the method, click here.

A simulation of Naïve 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, Maple will calculate the solution vector [X].

Input Parameters:
n = number of equations
[A] = *n*x*n* coefficient matrix
[RHS] = *n*x*1* right hand side array
Digits = number of significant digits used in calculation

```
> restart;
  n:=4;
  A:=Matrix([[1,10,100,1000],[1,15,225,3375],[1,20,400,8000],[1,
  22.5,506.25,11391]]);
  RHS:=[227.04,362.78,517.35,602.97];
  Digits:=10;
```

$$n := 4$$

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

$$RHS := [227.04, 362.78, 517.35, 602.97]$$

$$Digits := 10 \qquad (2.1)$$

Section 2: Naïve Gauss Elimination

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

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

```
> with(linalg):
  C:=augment(A,RHS);
Warning, the protected names norm and trace have been redefined
and unprotected
```

$$C := \begin{bmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 1 & 15 & 225 & 3375 & 362.78 \\ 1 & 20 & 400 & 8000 & 517.35 \\ 1 & 22.5 & 506.25 & 11391 & 602.97 \end{bmatrix} \qquad (3.1)$$

Step 1: Forward Elimination

Forward elimination of unknowns consists of (n-1) steps. In each step k, the coefficient element 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 while demonstrating the intermediate coefficient matrices that are produced for each step k.

```
> with(linalg):
```

```
> #Defining the augmented matrix [C].
C:=augment(A,RHS):
print(`start`,C);
print(` `);
#Conducting k, or (n-1) steps of forward elimination.
for k from 1 by 1 to (n-1) do
    #Defining the proper row elements to transform [C] into [U].
    for i from k+1 by 1 to n do
        #Generating the value that is multiplied to each equation.
        multiplier:=C[i,k]/C[k,k]:
            for j from k by 1 to n+1 do
            #Subtracting the product of the multiplier and pivot equation from the ith row to
generate new rows of [U] matrix.
                C[i,j]:=C[i,j]-multiplier*C[k,j]:
            end do:
    end do:
    print(`step`=k,C):
    print(` `);
end do:
```

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

$$step = 1, \begin{bmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 0 & 5 & 125 & 2375 & 135.74 \\ 0 & 10 & 300 & 7000 & 290.31 \\ 0 & 12.5 & 406.25 & 10391 & 375.93 \end{bmatrix}$$

$$step = 2, \begin{bmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 0 & 5 & 125 & 2375 & 135.74 \\ 0 & 0 & 50 & 2250 & 18.83 \\ 0 & 0. & 93.7500000 & 4453.500000 & 36.5800000 \end{bmatrix}$$

$$step = 3, \begin{bmatrix} 1 & 10 & 100 & 1000 & 227.04 \\ 0 & 5 & 125 & 2375 & 135.74 \\ 0 & 0 & 50 & 2250 & 18.83 \\ 0 & 0. & 0. & 234.750000 & 1.27375000 \end{bmatrix} \quad (3.1.1)$$

The new upper triangular coefficient matrix, [A1], is now:

```
> A1:=submatrix(C,1..n,1..n);
```

$$A1 := \begin{bmatrix} 1 & 10 & 100 & 1000 \\ 0 & 5 & 125 & 2375 \\ 0 & 0 & 50 & 2250 \\ 0 & 0. & 0. & 234.750000 \end{bmatrix} \quad (3.2)$$

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

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

```
> RHS1:=col(C,n+1);
```

$$RHS1 := [227.04 \quad 135.74 \quad 18.83 \quad 1.27375000] \quad (3.3)$$

This is the end of forward elimination steps. The new upper triangular coefficient matrix and right hand side array permit solving for the solution vector using backward substitution.

```
> i:='i':
  j:='j':
```

Step 2: Back Substitution

Back substitution begins with solving the last equation as it has only one unknown. The remaining equations can be solved for using the following formula:

$$x_{'i'} = \frac{\left(c_{'i'} - Sum\left(a_{'i','j'} \cdot x_{'j'},\ j = (i+1)..\text{"}n\text{"}\right)\right)}{a_{'i','i'}}$$

$$x_i = \frac{c_i - \sum_{j=i+1}^{\text{"n"}} a_{i,j} x_j}{a_{i,i}} \quad (3.2.1)$$

```
> #Defining [X] as a vector.
  X:=Array(1..n):
  #Solving for the n^th equation as it has only one unknown.
  X[n]:=RHS1[n]/A1[n,n]:
  #Solving for the remaining (n-1) unknowns working backwards from the (n-1)^th equation to
  the first equation.
  for i from (n-1) by -1 to 1 do
      #Setting the series sum equal to zero.
      summ:=0;
      for j from i+1 by 1 to n do
          #
          summ:=summ+A1[i,j]*X[j]:
      end do:
      #Generating the summation term in equation (3.2.1), at which time the unknowns that
  have been solved for are used to calculate X_i.
      X[i]:=(RHS1[i]-summ)/A1[i,i]:
  end do:
```

The solution vector [X] is:

```
> X;
```

$$[-4.2279552 \quad 21.25989030 \quad 0.1324306710 \quad 0.005425985091] \tag{3.2.2}$$

Section 3: Exact solution

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

```
> with(linalg):
  exactsoln:=linsolve(A,RHS);
```

$$exactsoln := [-4.227959624 \quad 21.25989118 \quad 0.132430616 \quad 0.00542598620] \tag{4.1}$$

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.

Conclusion

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

Question 1: Change the number of significant digits used in calculations. See how this value affects the accuracy in the solution vector.

Question 2: The velocity of a rocket is given at three different times:

time	***velocity***
5 *sec*	$106.8 \left(\frac{m}{s}\right)$
8 *sec*	$177.2 \left(\frac{m}{s}\right)$
12 *sec*	$279.2 \left(\frac{m}{s}\right)$

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

The coefficients a_1, a_2, a_3 for the above expressions were found to be given by
$Matrix([[25, 5, 1], [64, 8, 1], [144, 12, 1]]) Vector(3, symbol = a) = < 106.8, 177.2, 279.2 >$

$$\begin{bmatrix} 25\, a_1 + 5\, a_2 + a_3 \\ 64\, a_1 + 8\, a_2 + a_3 \\ 144\, a_1 + 12\, a_2 + a_3 \end{bmatrix} = \begin{bmatrix} 106.8 \\ 177.2 \\ 279.2 \end{bmatrix} \quad (6.1)$$

Question 3: Choose a set of equations that has a unique solution but for which Naïve Gauss Elimination method fails.