Pivoting Strategies

```
> restart;
> with(LinearAlgebra):
```

To illustrate pivoting strategies, we use Problem 10b on Page 369.

We find an exact solution by setting Digits high and using LinearSolve .

```
> Digits:=25;
```

$$Digits := 25$$

```
> A:=Matrix(3,3,[[3.3330,15920,10.333],[2.2220,16.710,9.6120],
  [-1.5611,5.1792,-1.6855]]);
```

$$A := \begin{bmatrix} 3.3330 & 15920 & 10.333 \\ 2.2220 & 16.710 & 9.6120 \\ -1.5611 & 5.1792 & -1.6855 \end{bmatrix}$$

```
> b:=Vector([7953,0.965,2.714]);
```

$$b := \begin{bmatrix} 7953 \\ 0.965 \\ 2.714 \end{bmatrix}$$

```
> soln1:=LinearSolve(A,b);
```

$$soln1 := \begin{bmatrix} 1.000000000000000000000000 \\ 0.5000000000000000000000001 \\ -1.000000000000000000000242 \end{bmatrix}$$

It seems clear that the exact solution is

```
> soln1:=Vector([1,.5,-1]);
```

$$soln1 := \begin{bmatrix} 1 \\ 0.5 \\ -1 \end{bmatrix}$$

We test it by matrix multiplication using MatrixVectorMultiply .

```
> MatrixVectorMultiply(A,soln1);
```

$$\begin{bmatrix} 7953.0000 \\ 0.9650 \\ 2.71400 \end{bmatrix}$$

We were correct. We can generate the equations from the matrices by using GenerateEquations . We then solve the system using solve. We first reset **Digits** to 10.

```
> Digits:=10;
```

$$Digits := 10$$

```
> eq:=GenerateEquations(A,[x[1],x[2],x[3]],b);
```

$eq := [3.3330\, x_1 + 15920\, x_2 + 10.333\, x_3 = 7953,\ 2.2220\, x_1 + 16.710\, x_2 + 9.6120\, x_3 = 0.965,$

$-1.5611\, x_1 + 5.1792\, x_2 - 1.6855\, x_3 = 2.714]$

Befor using the solve command, we must convert **eq** from a list to a set, which we name **eq2**.

```
> eq2:=convert(eq,set);
```

$$eq2 := \{-1.5611\, x_1 + 5.1792\, x_2 - 1.6855\, x_3 = 2.714,\ 2.2220\, x_1 + 16.710\, x_2 + 9.6120\, x_3 = 0.965,\ 3.3330\, x_1 + 15920\, x_2 + 10.333\, x_3 = 7953\}$$

```
> soln2:=solve(eq2,{x[1],x[2],x[3]});
```

$$soln2 := \{x_1 = 1.,\ x_2 = 0.5000000000,\ x_3 = -1.\}$$

We can solve using Gaussian elimination with back substitution. We first create the augmented matrix.

```
> M:=GenerateMatrix(eq,[x[1],x[2],x[3]],augmented=true);
```

$$M := \begin{bmatrix} 3.3330 & 15920 & 10.333 & 7953 \\ 2.2220 & 16.710 & 9.6120 & 0.965 \\ -1.5611 & 5.1792 & -1.6855 & 2.714 \end{bmatrix}$$

We will use the Pivot command here. We pivot on the (1,1) entry and change rows 2 to 3 appropriately.

```
> M1:=Pivot(M,1,1,2..3);
```

$$M1 := \begin{bmatrix} 3.3330 & 15920 & 10.333 & 7953 \\ 0. & -10596.62333 & 2.723333333 & -5301.035000 \\ 0. & 7461.738456 & 3.154237864 & 3727.714990 \end{bmatrix}$$

We now pivot on the (2,2) entry and change row 3 appropriately.

```
> M2:=Pivot(M1,2,2,3);
```

$$M2 := \begin{bmatrix} 3.3330 & 15920 & 10.333 & 7953 \\ 0. & -10596.62333 & 2.723333333 & -5301.035000 \\ 0. & 0. & 5.071905448 & -5.071907 \end{bmatrix}$$

Since we are now in **reduced form**, We use back substitution.

```
> soln3:=BackwardSubstitute(M2);
```

$$soln3 := \begin{bmatrix} 1.00000057288604416 \\ 0.500000000078672291 \\ -1.00000030599939538 \end{bmatrix}$$

Some inaccury here. If you check back trough our steps, we did use **partial pivoting** where no row swaps were necessary.

The procedure GaussianElimination seems to do partial pivoting, as seen below.

```
> M1:=GaussianElimination(M);
```

$M1 :=$

$[[3.33300000000000018,\ 15920.,\ 10.3330000000000002,\ 7953.],$
$[0.,\ -10596.6233333333330,\ 2.72333333333333361,\ -5301.03499999999985],$
$[0.,\ 0.,\ 5.07190544735661941,\ -5.07190544735613003]]$

Next we do **scaled partial pivoting**. We first find the maximum absolute value of the coefficients of each row, called **scale factors** s_i.

```
> M;
```

$$\begin{bmatrix} 3.3330 & 15920 & 10.333 & 7953 \\ 2.2220 & 16.710 & 9.6120 & 0.965 \\ -1.5611 & 5.1792 & -1.6855 & 2.714 \end{bmatrix}$$

```
> s[1]:=15920;s[2]:=16.71;s[3]:=5.1792;
```

$$s_1 := 15920$$

$$s_2 := 16.71$$

$$s_3 := 5.1792$$

Then we find quotients by dividing the pivot column entries by the scale factors, then taking absolute values.

```
> q[1]:=abs(M[1,1])/s[1];q[2]:=abs(M[2,1])/s[2];q[3]:=abs(M[3,1])/s
  [3];
```

$$q_1 := 0.0002093592965$$

$$q_2 := 0.1329742669$$

$$q_3 := 0.3014172073$$

The largest quotient is in row 3, so we swap rows 1 and 3 and then pivot. We also swap the scale factors for these rows.

```
> M1:=RowOperation(M,[1,3]);
```

$$M1 := \begin{bmatrix} -1.5611 & 5.1792 & -1.6855 & 2.714 \\ 2.2220 & 16.710 & 9.6120 & 0.965 \\ 3.3330 & 15920 & 10.333 & 7953 \end{bmatrix}$$

```
> ss:=s[1];s[1]:=s[3];s[3]:=ss;
```

$$ss := 15920$$

$$s_1 := 5.1792$$

$$s_3 := 15920$$

```
> M2:=Pivot(M1,1,1,2..3);
```

$$M2 := \begin{bmatrix} -1.5611 & 5.1792 & -1.6855 & 2.714 \\ 1.\ 10^{-9} & 24.08184190 & 7.212934598 & 4.827986355 \\ -1.\ 10^{-9} & 15931.05776 & 6.734401895 & 7958.794480 \end{bmatrix}$$

We correct the roundoff problems by making the (2,1) and (3,1) entries 0.

```
> M2[2,1]:=0;M2[3,1]:=0;M2;
```

$$M2_{2,\,1} := 0$$

$$M2_{3,\,1} := 0$$

$$\begin{bmatrix} -1.5611 & 5.1792 & -1.6855 & 2.714 \\ 0 & 24.08184190 & 7.212934598 & 4.827986355 \\ 0 & 15931.05776 & 6.734401895 & 7958.794480 \end{bmatrix}$$

Now we do the same thing over starting in the (2,2) position and working down and to the right. We do not recompute the scale factors, however.

```
> q[2]:=abs(M2[2,2])/s[2];q[3]:=abs(M2[3,2])/s[3];
```

$$q_2 := 1.441163489$$

$$q_3 := 1.000694583$$

There is no need to swap rows here before pivoting.

```
> M3:=Pivot(M2,2,2,3);
```

$$M3 := \begin{bmatrix} -1.5611 & 5.1792 & -1.6855 & 2.714 \\ 0 & 24.08184190 & 7.212934598 & 4.827986355 \\ 0. & 0. & -4764.897194 & 4764.897194 \end{bmatrix}$$

We now use back substitution to find the solution.

```
> soln4:=BackwardSubstitute(M3);
```

$$soln4 := \begin{bmatrix} 1.00000000041329828 \\ 0.500000000124575239 \\ -1. \end{bmatrix}$$

This is a bit better. Now let's see what **complete pivoting** does.

```
> M;
```

$$\begin{bmatrix} 3.3330 & 15920 & 10.333 & 7953 \\ 2.2220 & 16.710 & 9.6120 & 0.965 \\ -1.5611 & 5.1792 & -1.6855 & 2.714 \end{bmatrix}$$

We first need to get the 15920 to the (1,1) position. We use ColumnOperation for this, which works line **RowOperation**. This also changes the solution vector to $[x_2, x_1, x_3]$. Then we pivot.

```
> M1:=ColumnOperation(M,[1,2]);
```

$$M1 := \begin{bmatrix} 15920 & 3.3330 & 10.333 & 7953 \\ 16.710 & 2.2220 & 9.6120 & 0.965 \\ 5.1792 & -1.5611 & -1.6855 & 2.714 \end{bmatrix}$$

```
> M2:=Pivot(M1,1,1,2..3);
```

$$M2 := \begin{bmatrix} 15920 & 3.3330 & 10.333 & 7953 \\ -1.\ 10^{-8} & 2.218501606 & 9.601154244 & -7.382652642 \\ -1.\ 10^{-9} & -1.562184314 & -1.688861600 & 0.126677286 \end{bmatrix}$$

We change the (2,1) and (3,1) entries to 0.

```
> M2[2,1]:=0;M2[3,1]:=0;M2;
```

$$M2_{2,1} := 0$$

$$M2_{3,1} := 0$$

$$\begin{bmatrix} 15920 & 3.3330 & 10.333 & 7953 \\ 0 & 2.218501606 & 9.601154244 & -7.382652642 \\ 0 & -1.562184314 & -1.688861600 & 0.126677286 \end{bmatrix}$$

We now need to get the 9.601154244 to the (2,2) position. We use **ColumnOperation** again for this again. This also changes the solution vector to $[x_2, x_3, x_1]$. Then we pivot.

```
> M3:=ColumnOperation(M2,[2,3]);
```

$$M3 := \begin{bmatrix} 15920 & 10.333 & 3.3330 & 7953 \\ 0 & 9.601154244 & 2.218501606 & -7.382652642 \\ 0 & -1.688861600 & -1.562184314 & 0.126677286 \end{bmatrix}$$

```
> M4:=Pivot(M3,2,2,3);
```

$$M4 := \begin{bmatrix} 15920 & 10.333 & 3.3330 & 7953 \\ 0 & 9.601154244 & 2.218501606 & -7.382652642 \\ 0. & 0. & -1.171945591 & -1.171945592 \end{bmatrix}$$

We can now back substitute.

```
> soln5:=BackwardSubstitute(M4);
```

$$soln5 := \begin{bmatrix} 0.500000000000219824 \\ -1.00000000061378103 \\ 1.00000000085328189 \end{bmatrix}$$

This does the best job overall, giving $x_1 =$ 1.00000000085328188, $x_2 = 0.500000000000219824$, $x_3 = -1.00000000061378102$.