COMP 350 Solutions to Assignment 3

1. (a) Forward elimination

- $k = 1$
  Interchange row 1 and row 4 :

$$\begin{bmatrix} 4 & -5 & -6 & 7 \\ -2 & 3 & -4 & 5 \\ 3 & 4 & 5 & -6 \\ 1 & 2 & 3 & -4 \end{bmatrix} \begin{bmatrix} x_4 \\ x_2 \\ x_3 \\ x_1 \end{bmatrix} = \begin{bmatrix} -4 \\ -14 \\ 10 \\ 6 \end{bmatrix}$$

  Perform row $2 - (-1/2) *$ row 1, row $3 - (3/4) *$ row 1 and row $4 - (1/4) *$ row 1:

$$\begin{bmatrix} 4 & -5 & -6 & 7 \\ 0 & \frac{1}{2} & -7 & \frac{17}{2} \\ 0 & \frac{31}{4} & \frac{19}{2} & -\frac{45}{4} \\ 0 & \frac{13}{4} & \frac{9}{2} & -\frac{23}{4} \end{bmatrix} \begin{bmatrix} x_4 \\ x_2 \\ x_3 \\ x_1 \end{bmatrix} = \begin{bmatrix} -4 \\ -16 \\ 13 \\ 7 \end{bmatrix}$$

- $k = 2$
  Interchange row 2 and row 3 :

$$\begin{bmatrix} 4 & -5 & -6 & 7 \\ 0 & \frac{31}{4} & \frac{19}{2} & -\frac{45}{4} \\ 0 & \frac{1}{2} & -7 & \frac{17}{2} \\ 0 & \frac{13}{4} & \frac{9}{2} & -\frac{23}{4} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -4 \\ 13 \\ -16 \\ 7 \end{bmatrix}$$

  Perform row $3 - (2/31) *$ row 2 and row $4 - (13/31) *$ row 2:

$$\begin{bmatrix} 4 & -5 & -6 & 7 \\ 0 & \frac{31}{4} & \frac{19}{2} & -\frac{45}{4} \\ 0 & 0 & -\frac{236}{31} & \frac{286}{31} \\ 0 & 0 & \frac{16}{31} & -\frac{32}{31} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -4 \\ 13 \\ -\frac{522}{31} \\ \frac{48}{31} \end{bmatrix}$$

- $k = 3$
  There is no row interchange. Perform row $4 - (-16/236) *$ row 3:

$$\begin{bmatrix} 4 & -5 & -6 & 7 \\ 0 & \frac{31}{4} & \frac{19}{2} & -\frac{45}{4} \\ 0 & 0 & -\frac{236}{31} & \frac{286}{31} \\ 0 & 0 & 0 & -\frac{24}{59} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -4 \\ 13 \\ -\frac{522}{31} \\ \frac{24}{59} \end{bmatrix}$$

(b) Back substitution.

$$x_4 = \frac{\frac{24}{59}}{-\frac{24}{59}} = -1$$

$$x_3 = \frac{-\frac{522}{31} - \frac{286}{31} * (-1)}{-\frac{236}{31}} = 1$$

$$x_2 = \frac{13 - \frac{19}{2} * 1 - (-\frac{45}{4}) * (-1)}{\frac{31}{4}} = -1$$

$$x_1 = \frac{-4 - (-5) * (-1) - (-6) * 1 - 7 * (-1)}{4} = 1$$

2.

$$\begin{aligned}
\text{Total \# of FLOPS} &= \sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=j}^{n} 2 \\
&= \sum_{i=1}^{n} \sum_{j=i}^{n} 2(n-j+1) \\
&= \sum_{i=1}^{n} (n-i+1)(n-i+1+n-n+1) \\
&= \sum_{i=1}^{n} (n-i+1)^2 - \sum_{i=1}^{n} (n-i+1) \\
&= \frac{n(n+1)(2n+1)}{6} + \frac{n(1+n)}{2} \\
&= \frac{1}{3}n^3 + n^2 + \frac{2}{3}n
\end{aligned}$$

3. (a) Matlab codes

```
function x = genp(a,d,b)
% genp.m GENP for a special linear system system
% input: a, b and d are (2n+1)-vectors
% output: x is the solution for the system Ax=b; A is formed by a and d

n = (length(b)-1)/2;
for k = 1:n
   mult = a(k)/d(k);
   d(2*n-k+2) = d(2*n-k+2) - mult*a(2*n-k+2);
   b(2*n-k+2) = b(2*n-k+2) - mult*b(k);
end
x = zeros(2*n+1,1);
for k = 2*n+1:-1:n+1
   x(k) = b(k)/d(k);
end
for k = n:-1:1
   x(k) = (b(k)-a(2*n-k+2)*x(2*n-k+2))/d(k);
end
```

Flops : $(1+2+2) * n + 1 * (2n + 1 - (n+1) + 1) + 3 * n \approx 9n$
(Notice that the index operations were not taken into account.)
Storage: four one-dimensional arrays of size $2n+1$ to store a, d, b and x.

```
function x = gepp(a,d,b)
% gepp.m GEPP for a special linear system system
% input: a, b and d are (2n+1)-vectors
% output: x is the solution for the system Ax=b; A is formed by a and d

n = (length(b)-1)/2;
for k = 1:n
   if abs(d(k)) < abs(a(k))
      temp = d(k);
      d(k) = a(k);
      a(k) = d(k);
      temp = a(2*n-k+2);
      a(2*n-k+2) = d(2*n-k+2);
      d(2*n-k+2) = temp;
      b([k,2*n-k+2]) = b([2*n-k+2,k]);
   end
   mult = a(k)/d(k);
   d(2*n-k+2) = d(2*n-k+2) - mult*a(2*n-k+2);
   b(2*n-k+2) = b(2*n-k+2) - mult*b(k);
end
x = zeros(2*n+1,1);
for k = 2*n+1:-1:n+1
   x(k) = b(k)/d(k);
end
for k = n:-1:1
   x(k) = (b(k)-a(2*n-k+2)*x(2*n-k+2))/d(k);
end
```

In additional to $9n$ flops, gepp.m needs n comparisons. Like genp.m, it needs four one-dimensional arrays of size $2n + 1$.

(b)(c)
```
% test.m is to test genp(a,d,b) and gepp(a,d,b)
%       a(n+1) = d(n+1)
%       b(k) = d(k) + a(2*n+2-k),   k=1:2*n+1
%

a = randn(2*n+1,1);
d = randn(2*n+1,1);
a(n+1) = d(n+1);

for k=1:2*n+1
   b(k) = d(k)+a(2n+2-k);
end;
b(n+1)= d(n+1);

x = ones(n,1);
xnp = genp(a,d,b);
xpp = gepp(a,d,b);
[a, d, b, xnp, xpp]

rnp =norm(x-xnp)/norm(x)
rpp =norm(x-xpp)/norm(x)
```

```
%change d(1) to see the difference between genp and gepp

d(1) = 1.0e-13;
b(1) = d(1) + a(2*n+1);

xnp = genp(a,d,b);
xpp = gepp(a,d,b);
[a, d, b, xnp, xpp]
rnp = norm(x-xnp)/norm(x)
rpp = norm(x-xpp)/norm(x)
```

Output:

```
>> format short e
>> n = 4;
>> test

Xinye: Put the test results here.
```

The relative errors for GEPP and GENP are very similar for the first example, this means that the pivoting and non-pivoting methods do not make any significant difference for this case.

The relative error for GEPP is much smaller than that GENP for the 2nd example. This indicates that partial pivoting is crucial for accuracy of the solution.