1a) Please see the code in the file named gauss-seidel.m. I inserted the following parameters:

```
A=[3 -1 1; 3 6 -2; 3 3 7];
b=[5;1;3];
x0=[0;0;0];
tol=0.000001;
maxiter=100;
[x,iter] = gauss_seidel(A,b,x0,tol,maxiter)
```

With this stopping criterion:

$||x^{(k)} - x^{(k-1)}||_{inf} < 10^{-6}$

And got the following output where x is the solution vector and iter is the number of iterations it took the algorithm to converge:

```
>> HW4_script

x =

   1.469135995626869
  -0.555555703628854
   0.037037017715137


iter =

    10
```

b) For this stopping criterion: $||\mathrm{A}x^{(k)} - \mathrm{b}||_{inf} < 10^{-6}$

```
>> HW4_script

x =

   1.469135995626869
  -0.555555703628854
   0.037037017715137


iter =

    10
```

For this stopping criterion: $\frac{||x^{(k)}-x^{(k-1)}||_{inf}}{||x^{(k)}||_{inf}} < 10^{-6}$

```
>> HW4_script

x =

   1.469135995626869
  -0.555555703628854
   0.037037017715137
```

```
iter =

    10
```

All three different stopping criterion for the gauss-seidel method give the same result where the solution vector, x, is equal to [1.4691; -0.5556; 0.0370] and the number of iterations it takes the algorithm to converge is 10.

c) `>> HW4_script`

For 10^-2:

```
x =

   1.471151423532376
  -0.556815318720080
   0.036713097937588


iter =

     4
```

For 10^-6:

```
x =

   1.469135995626869
  -0.555555703628854
   0.037037017715137


iter =

    10
```

For 10^-10:

```
x =

   1.469135802485223
  -0.555555555571215
   0.037037037036854


iter =

    16
```

There is a difference in both the solution and the number of iterations it takes for the algorithm to converge when the tolerance is changed. As the tolerance level gets higher, the solution seems to become more accurate. In addition, the number of iterations is constantly increasing.

d)
```
>> HW4_script

x =

   1.469136033982938
  -0.555555814778738
   0.037037048912486


iter =

    11


x =

   1.469136204151538
  -0.555555960556187
   0.037037038459135


iter =

    11


x =

   1.469135879552570
  -0.555555557279271
   0.037037004740015


iter =

    13
```

The initial guess vectors that I chose were [4;1;2], [5;1;3], and [10;12;13]. Each of these vectors gave different close answers to the actual solution. For example, the second vector had the most accurate solution for x3 while the third vector had the most accurate solution for x1 and x2. In addition, the number of iterations it took the algorithm to converge or the first and second vector was equal, 11, while the third vector took two more iterations, 13, to converge.

2a)

2a) $T_J = D^{-1}(L+U)$

$$D = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{= D^{-1} = I} \quad L = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & -2 & 0 \end{bmatrix} \quad U = \begin{bmatrix} 0 & -2 & 2 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$(L+U) = \begin{bmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{bmatrix}$$

$$D^{-1}(L+U) = I(L+U) = \begin{bmatrix} 0 & -2 & 2 \\ -1 & 0 & -1 \\ -2 & -2 & 0 \end{bmatrix}$$

Find eigenvalues where $\det(A - \lambda I) = 0$

$$A - \lambda I = \begin{bmatrix} -\lambda & -2 & 2 \\ -1 & -\lambda & -1 \\ -2 & -2 & -\lambda \end{bmatrix}$$

$$\det = -2\begin{bmatrix} -2 & 2 \\ -\lambda & -1 \end{bmatrix} - (-2)\begin{bmatrix} -\lambda & 2 \\ -1 & -1 \end{bmatrix} + -\lambda\begin{bmatrix} -\lambda & -2 \\ -1 & -\lambda \end{bmatrix} = 0$$

$$\Rightarrow -2(2+2\lambda) + 2(\lambda+2) - \lambda(\lambda^2 - 2) \Rightarrow \cancel{-4} \cancel{-4\lambda} \cancel{+2\lambda} \cancel{+4} - \lambda^3 \cancel{+2\lambda}$$

$$\Rightarrow -\lambda^3 = 0 \quad \text{so} \quad \lambda = 0$$

Therefore, $\rho(T_J) = 0$ ✓

b) Running jacobi.m where x is the solution vector and iter is the number of iterations it took the algorithm to converge:

```
>> HW4_script

x =

     1
     2
    -1
```

```
iter =

     4
```

c)

2c) $T_{GS} = (D-L)^{-1} U$ we know D, L, U from 2a)

$D - L = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}$ $(D-L)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$

$(D-L)^{-1} \cdot U = \begin{bmatrix} 0 & -2 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 2 \end{bmatrix}$

Find eigenvalues: $\det(A - \lambda I) = 0$

$A - \lambda I = \begin{bmatrix} -\lambda & -2 & 2 \\ 0 & 2-\lambda & -3 \\ 0 & 0 & 2-\lambda \end{bmatrix}$

$\det = \cancel{0 \cdot \begin{bmatrix} -2 & 2 \\ 2-\lambda & -3 \end{bmatrix}} - \cancel{0 \cdot \begin{bmatrix} -\lambda & 2 \\ 0 & -3 \end{bmatrix}} + (2-\lambda)\begin{bmatrix} -\lambda & -2 \\ 0 & 2-\lambda \end{bmatrix} = 0$

$\Rightarrow (2-\lambda)(-2\lambda + \lambda^2) \Rightarrow -4\lambda + 2\lambda^2 + 2\lambda^2 - \lambda^3$

$\Rightarrow -\lambda^3 + 4\lambda^2 - 4\lambda \Rightarrow -\lambda(\lambda^2 - 4\lambda + 4)$

$\Rightarrow -\lambda(\lambda - 2)(\lambda - 2) = 0$ so $\lambda = 0, 2, 2$

Therefore, $\rho(T_{GS}) = 2$ ✓

d) Running gauss_seidel.m where x is the solution vector and iter is the number of iterations it took the algorithm to converge:

```
>> HW4_script

x =

   1.0e+09 *

   1.308622849000000
  -1.325400062000000
   0.033554431000000
```

```
iter =

    25
```

3.

3. Using $||x^{(k)} - x||$ and $x^{(k)} = Tx^{(k-1)} + c$, $k = 1, 2, \ldots$
we can do the following; $||x^{(k)} - x|| \Rightarrow ||Tx^{(k-1)} + c - x||$
We also know that $x = Tx + c$ so we can do this:
$||Tx^{(k-1)} + c - x|| \Rightarrow ||Tx^{(k-1)} \cancel{+c} - Tx \cancel{-c}||$ then,
$||Tx^{(k-1)} - Tx|| \Rightarrow ||T(x^{(k-1)} - x)||$ now we can do:
$||T(Tx^{(k-2)} \cancel{+c} - Tx \cancel{-c})|| \Rightarrow ||T^2(x^{(k-2)} - x)||$, , ,
If we keep going, this will give:
$\Rightarrow ||T^k(x^{(0)} - x)||$
So therefore, $||x^{(k)} - x|| \le ||T||^k ||x^{(0)} - x||$ ✓