Systems of linear equations

Our Matlab function for naive Gaussian elimination looks like this:

```
function x = naiv_gauss(A,b);
n = length(b);  x = zeros(n,1);

for k=1:n-1  % forward elimination
  for i=k+1:n
    xmult = A(i,k)/A(k,k);
    for j=k+1:n
      A(i,j) = A(i,j)-xmult*A(k,j);
    end
    b(i) = b(i)-xmult*b(k);
  end
end

% back substitution
x(n) = b(n)/A(n,n);
for i=n-1:-1:1
  sum = b(i);
  for j=i+1:n
    sum = sum-A(i,j)*x(j);
  end
  x(i) = sum/A(i,i);
end
```

Example:

$$\begin{pmatrix} 1^9 & 1^8 & 1^7 & \cdots & 1 & 1 \\ 2^9 & 2^8 & 2^7 & & 2 & 1 \\ 3^9 & 3^8 & 3^7 & & 3 & 1 \\ \vdots & & & & & \vdots \\ 10^9 & 10^8 & 10^7 & \cdots & 10 & 1 \end{pmatrix} x = \begin{pmatrix} 2 \\ 3 \\ 4 \\ \vdots \\ 11 \end{pmatrix}$$

Exact solution: $x = [0, 0, 0, \cdots, 0, 1, 1]^T$.

>> naiv_gauss(A,b)	>> A\b
ans =	ans =
0.00000000000088	-0.00000000000000
-0.00000000004331	0.00000000000000
0.00000000091111	-0.00000000000000
-0.00000001068735	0.00000000000000
0.00000007662105	-0.00000000000004
-0.00000034609131	0.00000000000017
0.00000097786263	-0.00000000000049
-0.00000165121338	0.00000000000086
1.00000149290615	0.99999999999921
0.99999945973353	1.00000000000029

Max error:

$1.6 \cdot 10^{-6}$ $\qquad$ $8.6 \cdot 10^{-13}$

LU factorization

```
>> % make a randon 4x4 matrix
>> A = rand(4)
A =
    0.3961    0.0850    0.6639    0.1191
    0.5327    0.0981    0.0208    0.3344
    0.7264    0.4951    0.3609    0.1855
    0.3239    0.8650    0.0558    0.6908

>> % make a rhs st the given x
>> % would be the solution
>> x = [1;1;1;1];  b = A*x;
```

```
>> % LU factorization
>> [L,U] = lu(A)
L =
    0.5453   -0.2872    1.0000         0
    0.7334   -0.4114   -0.6573    1.0000
    1.0000         0         0         0
    0.4460    1.0000         0         0
U =
    0.7264    0.4951    0.3609    0.1855
         0    0.6442   -0.1052    0.6081
         0         0    0.4369    0.1927
         0         0         0    0.5752

>> % back substitution and solving
>> v = L\b;
>> xs = U\v
xs =
    1.0000
    1.0000
    1.0000
    1.0000
```

Iterative mathods for systems of linear equations

Example: We wish to solve a system:

$$Ax = b$$

where A is a 6×6 matrix

```
A =

     4    -1    -1     0     0     0
    -1     4     0    -1     0     0
    -1     0     4    -1    -1     0
     0    -1    -1     4     0    -1
     0     0    -1     0     4    -1
     0     0     0    -1    -1     4
```

and the rhs vector is:

$$b = [1; 5; 0; 3; 1; 5].$$

The exact solution becomes:

$$x = [1; 2; 1; 2; 1; 2].$$

We solve the system wth iterative methods, with the initial value:

$$x^{(0)} = [0.25; 1.25; 0; 0.75; 0.25; 1.25].$$

Jacobi iterations:

k	x1	x2	x3	x4	x5	x6
1	0.2500	1.2500	0	0.7500	0.2500	1.2500
2	0.5625	1.5000	0.3125	1.3750	0.5625	1.5000
3	0.7031	1.7344	0.6250	1.5781	0.7031	1.7344
4	0.8398	1.8203	0.7461	1.7734	0.8398	1.8203
5	0.8916	1.9033	0.8633	1.8467	0.8916	1.9033
6	0.9417	1.9346	0.9075	1.9175	0.9417	1.9346
7	0.9605	1.9648	0.9502	1.9442	0.9605	1.9648
8	0.9787	1.9762	0.9663	1.9699	0.9787	1.9762
9	0.9856	1.9872	0.9819	1.9797	0.9856	1.9872
10	0.9923	1.9913	0.9877	1.9890	0.9923	1.9913
11	0.9948	1.9953	0.9934	1.9926	0.9948	1.9953
12	0.9972	1.9968	0.9955	1.9960	0.9972	1.9968
13	0.9981	1.9983	0.9976	1.9973	0.9981	1.9983
14	0.9990	1.9988	0.9984	1.9985	0.9990	1.9988
15	0.9993	1.9994	0.9991	1.9990	0.9993	1.9994
16	0.9996	1.9996	0.9994	1.9995	0.9996	1.9996
17	0.9997	1.9998	0.9997	1.9996	0.9997	1.9998
18	0.9999	1.9998	0.9998	1.9998	0.9999	1.9998
19	0.9999	1.9999	0.9999	1.9999	0.9999	1.9999
20	1.0000	1.9999	0.9999	1.9999	1.0000	1.9999

One needs more iterations to get the convergence.

Gauss-Seidal iterations:

k	x1	x2	x3	x4	x5	x6
1	0.2500	1.2500	0	0.7500	0.2500	1.2500
2	0.5625	1.5781	0.3906	1.5547	0.6602	1.8037
3	0.7422	1.8242	0.7393	1.8418	0.8857	1.9319
4	0.8909	1.9332	0.9046	1.9424	0.9591	1.9754
5	0.9594	1.9755	0.9652	1.9790	0.9852	1.9910
6	0.9852	1.9911	0.9873	1.9924	0.9946	1.9967
7	0.9946	1.9967	0.9954	1.9972	0.9980	1.9988
8	0.9980	1.9988	0.9983	1.9990	0.9993	1.9996
9	0.9993	1.9996	0.9994	1.9996	0.9997	1.9998
10	0.9997	1.9998	0.9998	1.9999	0.9999	1.9999
11	0.9999	1.9999	0.9999	2.0000	1.0000	2.0000
12	1.0000	2.0000	1.0000	2.0000	1.0000	2.0000
======	======	======	======	======	======	======
13	1.0000	2.0000	1.0000	2.0000	1.0000	2.0000
14	1.0000	2.0000	1.0000	2.0000	1.0000	2.0000
15	1.0000	2.0000	1.0000	2.0000	1.0000	2.0000

We see that after 12 iterations the method converges.

SOR iterations with $\omega = 1.12$:

k	x1	x2	x3	x4	x5	x6
1	0.2500	1.2500	0	0.7500	0.2500	1.2500
2	0.6000	1.6280	0.4480	1.6813	0.7254	1.9239
3	0.7893	1.8964	0.8411	1.9434	0.9671	1.9841
4	0.9518	1.9831	0.9805	1.9922	0.9940	1.9980
5	0.9956	1.9986	0.9972	1.9992	0.9994	1.9998
6	0.9994	1.9998	0.9998	1.9999	1.0000	2.0000
7	0.9999	2.0000	1.0000	2.0000	1.0000	2.0000
8	1.0000	2.0000	1.0000	2.0000	1.0000	2.0000
9	1.0000	2.0000	1.0000	2.0000	1.0000	2.0000
10	1.0000	2.0000	1.0000	2.0000	1.0000	2.0000

We see that after 8 iterations the method converges.

Error against number of iterations