Prima dell'analisi sulla risoluzione di sistemi lienari andiamo a vedere la sintassi di alcuni comandi in Matlab

Il ciclo "for": Il ciclo for in Matlab:

```
for i=min:delta:max         si cicla sulla variabile i che va da min a max con passo delta
         istruzione1    istruzioni del ciclo
         istruzione2
   ..............
end                     fine del ciclo
```

L'istruzione "if": la sintassi é:

```
if condizione1
else if   condizione2      istruzioni
else if   condizione3
   ..............
else
end                     fine dell'istruzione
```

la condizione puó essere seguita dal comando *break* che se la condizione é verificata blocca la routine.

Risoluzione di sistemi lineari in Matlab: Andiamo a scrivere un codice Matlab per la risoluzione di un sistema lineare $Ax = b$, con A matrice dei coefficienti, x vettore delle incognite e b vettore dei termini noti;

```
>> A = rand(5)
>> for i=1:5
        b(i)=0;
          for j=1:5
            b(i)=b(i)+A(i,j);   (somma degli elementi della i-sima riga di A)
          end
   end
>> b=b';
>> x=A\b;    (il comando \ risolve in Matlab il sistema Ax=b)
>> x
   x =
          1.0
          1.0
          1.0
          1.0
          1.0
```

Implementazione del Metodo di Jacobi
Dato un sistema in forma compatta $Ax = b$, sia

$$x_1 = -\frac{a_{12}}{a_{11}}x_1 - \cdots - \frac{a_{1n}}{a_{nn}}x_n + \frac{b_1}{a_{11}},$$

in generale

$$x_i = -\frac{a_{i1}}{a_{ii}}x_1 - \cdots - \frac{a_{i,i-1}}{a_{ii}}x_{i-1} - \frac{a_{i,i+1}}{a_{ii}}x_{i+1} - \cdots - \frac{a_{in}}{a_{nn}}x_n + \frac{b_i}{a_{ii}},$$

scrivibile come:

$$M = \begin{pmatrix} 0 & & -\frac{a_{ij}}{a_{ii}} \\ & \ddots & \\ -\frac{a_{ij}}{a_{ii}} & & 0 \end{pmatrix} \qquad d = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}$$

il sistema diventa dunque $x = Mx + d$. Sia $x^{(0)}$, scelta a caso, l'approssimzione iniziale della soluzione e definiamo lo schema iterativo:

$$x^{(1)} = Mx^{(0)} + d$$
$$x^{(2)} = Mx^{(1)} + d$$

....

$$x^{(k+1)} = Mx^{(k)} + d,$$

se A é diagonale dominante, cioé

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|$$

il metodo di Jacobi é convergente, ovvero la soluzione approssimata converge a quella esatta: $x^{(n)} \to x$.

Scriviamo adesso lo Script che implementa il metodo:

```
d=b./diag(A);    costruzione del vettore d
for i=1:n
  for j=1:n
    if i == j  M(i,j)=0;                        costruzione della matrice M
    else       M(i,j)=-A(i,j)/A(i,i);
  end
end
xk=zeros(n,1);      vettore inziale
for k=1:2000     ciclo di iterazioni del metodo
    xkp1=M*xk+d;
    errore = xkp1-xk         differenza tra due successive soluzioni
    eps=norm(errore,inf)   norma infinito (massimo) del vettore differenza
     if eps<0.0000001   controllo dell'errore
        break             esco dal ciclo e dalla routine
     end
   xk=xkp1       aggiornamento dei vettori soluzione
end
```

a questo punto posso salvare il file ad esempio come *jaboci.m*.

Osservazione: Prima di eseguire il programma devo aver definito A e b

É possibile mandare in run il programma dalla *Command Window*

```
>> clear
>> A=ones(12)
>> B=diag([11:22])
  B =
             11    0   .  .  0  0   0
             0    12                .
             0     0    .           .
             .      .        .      .
             .      .           21  .
             0     0  .  .  0   0  22
>> A=A+B
  A =
             11    1   .  .  1  1   1
             1    12                .
             1     1    .           .
             .      .        .      .
             .      .           21  .
             1     1  .  .  1   1  22
>> b=[22; 23; .... ;34];
>> b=b';
>> A\b;     risoluzione con la routine in Matlab
>> jacobi   risoluzione con il metodo di jacobi
   xkp1 =
                 1
                 .
                 .
                 .
                 1
```

Osservazione: É utile implementare una routine che costruisca la matrice A ed il vettore b, che potrá poi essere riusata ad ogni prova, o ad esempio, dal momento che il problema puó essere sperimentale, caricandoli attraverso dati contenuti su un file (Appendice).

Appendice: L'importazione dei dati da file.
I comandi per l'importazione sono: file $\rightarrow$ import data $\rightarrow$ nomefile.*.
Adesso, si sceglie il file contente i dati la cui estensione puó essere .txt, .dat, .doc etc.