



Università degli Studi di Napoli - Federico II

Facoltà di
Scienze Matematiche, Fisiche e Naturali

Corso di Laurea in Scienze Informatiche

Calcolo Numerico A.A. 2004 / 2005

Prof.ssa Eleonora Messina

Esercitazione Fitting

Realizzato da

Giovanni Di Cecca & Virginia Bellino

50 / 887

408 / 466



<http://www.dicecca.net>

2 *Esercitazione Fitting*

© 2005

Giovanni Di Cecca & Virginia Bellino – <http://www.dicecca.net>

Indice

✔ Testo dell'esercitazione

✔ Codice MATLAB

Esercizio 1

- Funzione di Runge interpolato con polinomio di grado 10
 - Plot dei dati
- Funzione di Runge interpolato con polinomio di grado 20
 - Plot dei dati
- Profilo superiore di un aereo su punti definiti
 - Plot dei dati

Esercizio 2

- Funzione di Runge interpolato con i nodi di Chebychev in $n = 10$
 - Plot dei dati
- Funzione di Runge interpolato con i nodi di Chebychev in $n = 20$
 - Plot dei dati
- Profilo superiore di un aereo su punti definiti interpolato con un Spline cubica
 - Plot dei dati

✔ Osservazioni

Testo dell'esercitazione

CORSO DI LAUREA IN INFORMATICA

a.a. 2004/2005

Calcolo Numerico

Esercitazione in Matlab

1. Usando la routine di Matlab “polyfit” costruire il polinomio di Lagrange di grado n ; commentare le prestazioni nei seguenti casi:

i. Sia $f(x)=1/(1+25x^2)$, dove $x \in [-1,1]$. Usando le routine di Matlab “polyval” e “plot”, plottare $f(x)$ ed i polinomi interpolanti di grado 10 e 20 su nodi equispaziati in $[-1,1]$.

ii. E’ data una tabella di 20 punti che individuano il profilo superiore di un aereo

$x = 1 \ 2 \ 4 \ 11 \ 25 \ 45 \ 47 \ 51 \ 58 \ 63 \ 74 \ 105 \ 125 \ 145 \ 170 \ 180 \ 190 \ 204 \ 215 \ 228$

$y = 2 \ 3 \ 4 \ 6 \ 9 \ 12 \ 13 \ 15 \ 20 \ 23 \ 25 \ 24 \ 23 \ 21 \ 19 \ 20 \ 24 \ 33 \ 39 \ 40$

plottare i dati ed il polinomio interpolante relativo ai dati.

2. Sviluppare i seguenti temi confrontando le prestazioni ottenute con quelle dei punti corrispondenti dell’esercizio 1.

i. Ripetere l’esercizio 1.i. utilizzando come nodi i punti di Chebychev in $[-1,1]$.

$$x_k = \cos\left(\frac{(2k+1)\pi}{2(n+1)}\right), \quad 0 \leq k \leq n$$

dove $n=10$ ed $n=20$ rispettivamente. Commentare.

ii. Relativamente alla tabella dell’esercizio 1.ii., plottare i dati e la spline cubica (usare la routine routine “spline” di Matlab) associata ai dati. Commentare.

Codice

MATLAB

```
%          Esercitazione Fitting
%
%          Programma elaborato da
%
%          Giovanni DI CECCA & Virginia BELLINO
%          50 / 887          408 / 466
%
%          http://www.dicecca.net
%
% Punto 1.i

clear % Pulisci memoria

clc % Pulisci schermo

% -----
% Costruzione della Funzione di Runge
% -----

% Polinomio di grado 10
pol=10

% Inserisci i nodi di interpolazione
x=linspace(1,-1,pol+1);

% Funzione di Runge
fun=inline(' (1+25*x.^2).^(-1) ');

% Vettore contenente i valori della funzione in x
y=feval(fun,x);

% -----
% Costruzione della Polinomio interpolante
% della Funzione di Runge
% -----

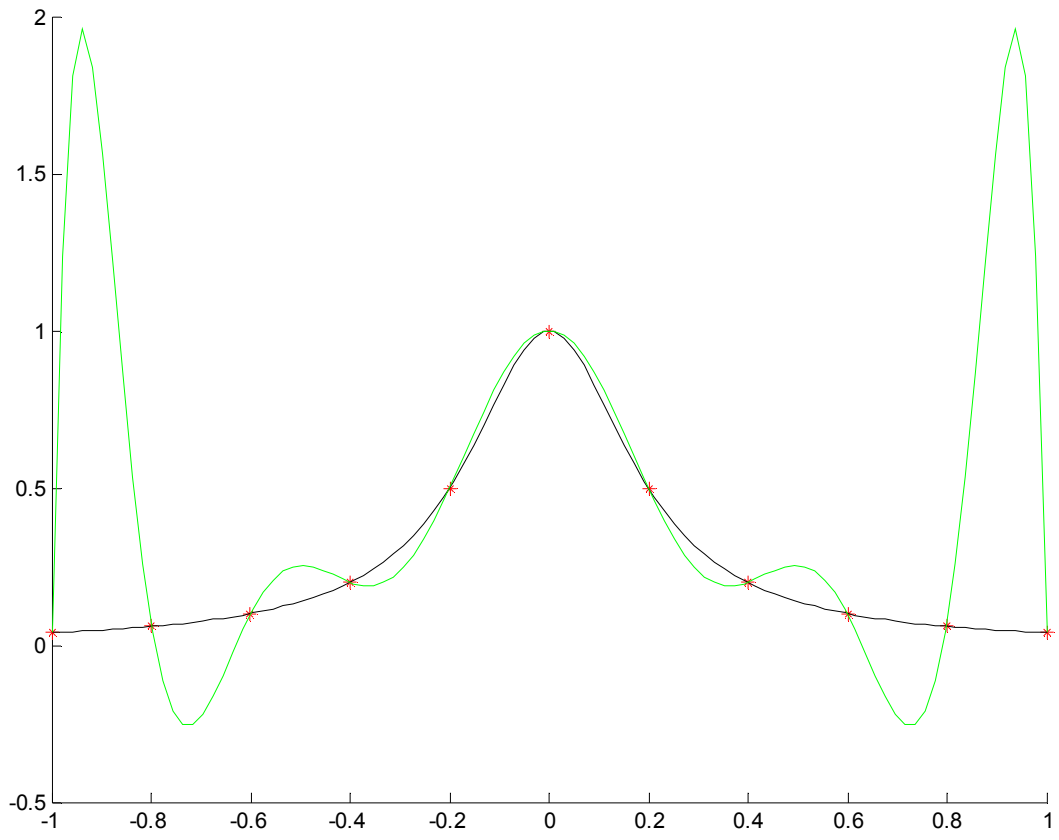
% Punti equispaziati in -1, 1
x1=linspace(-1,1);

% Vettore contenente i valori della funzione nei punti x1
y1=feval(fun,x1);

% Vettore dei coefficienti del polinomio interpolatore
p=polyfit(x,y,pol);

% Valuta il polinomio interpolante nei punti x1
f=polyval(p,x1);
```

```
% -----  
% Plot dei risultati  
% -----  
  
hold on  
  
zoom on  
  
% Plot dei punti della funzione di Runge  
plot(x,y,'r*')  
  
% Plot della funzione di Runge  
plot(x1,y1,'k')  
  
% Plot dei punti polinomio interpolante nei  
% punti equispaziati x1 della funzione di Runge  
plot(x1,f,'g')  
  
hold off
```

10 *Esercitazione Fitting*

```

%           Esercitazione Fitting
%
%           Programma elaborato da
%
%           Giovanni DI CECCA & Virginia BELLINO
%           50 / 887           408 / 466
%
%           http://www.dicecca.net
%
% Punto 1.i

clear % Pulisci memoria

clc % Pulisci schermo

% -----
% Costruzione della Funzione di Runge
% -----

% Polinomio di grado 10
pol=20

% Inserisci i nodi di interpolazione
x=linspace(1,-1,pol+1);

% Funzione di Runge
fun=inline(' (1+25*x.^2).^(-1) ');

% Vettore contenente i valori della funzione in x
y=feval(fun,x);

% -----
% Costruzione della Polinomio interpolante
% della Funzione di Runge
% -----

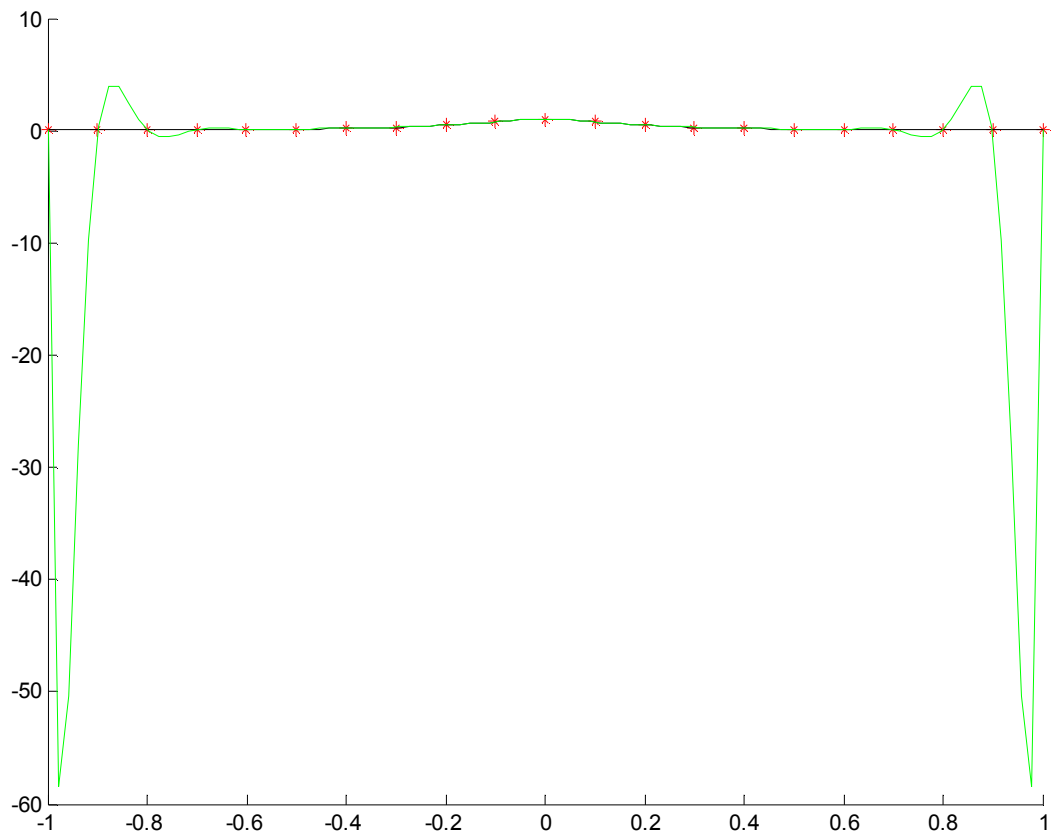
% Punti equispaziati in -1, 1
x1=linspace(-1,1);

% Vettore contenente i valori della funzione nei punti x1
y1=feval(fun,x1);

% Vettore dei coefficienti del polinomio interpolatore
p=polyfit(x,y,pol);

% Valuta il polinomio interpolante nei punti x1
f=polyval(p,x1);
```

```
% -----  
% Plot dei risultati  
% -----  
  
hold on  
  
zoom on  
  
% Plot dei punti della funzione di Runge  
plot(x,y,'r*')  
  
% Plot della funzione di Runge  
plot(x1,y1,'k')  
  
% Plot dei punti polinomio interpolante nei  
% punti equispaziati x1 della funzione di Runge  
plot(x1,f,'g')  
  
hold off
```



14 *Esercitazione Fitting*

```
% Esercitazione Fitting
%
% Programma elaborato da
%
% Giovanni DI CECCA & Virginia BELLINO
% 50 / 887 408 / 466
%
% http://www.dicecca.net
%
% Punto 1.ii

clear % Pulisci memoria

clc % Pulisci schermo

% -----
% Dati ingresso
% -----

x=[1,2,4,11,25,45,47,51,58,63,74,105,125,145,170,180,190,204,215,228]

y=[2,3,4,6,9,12,13,15,20,23,25,24,23,21,19,20,24,33,39,40]

% -----
% Costruzione del polinomio interpolante
% -----

% Valutazione dei punti con un polinomio di grado 19, in quanto
% si stanno utilizzando 20 punti x^0 compreso
p=polyfit(x,y,19)

% Punti equispaziati calcolati in base agli estremi della x
ls=linspace(1,228)

% Valutazione del polinomio interpolante
f=polyval(p,ls)

% -----
% Plot dei risultati
% -----

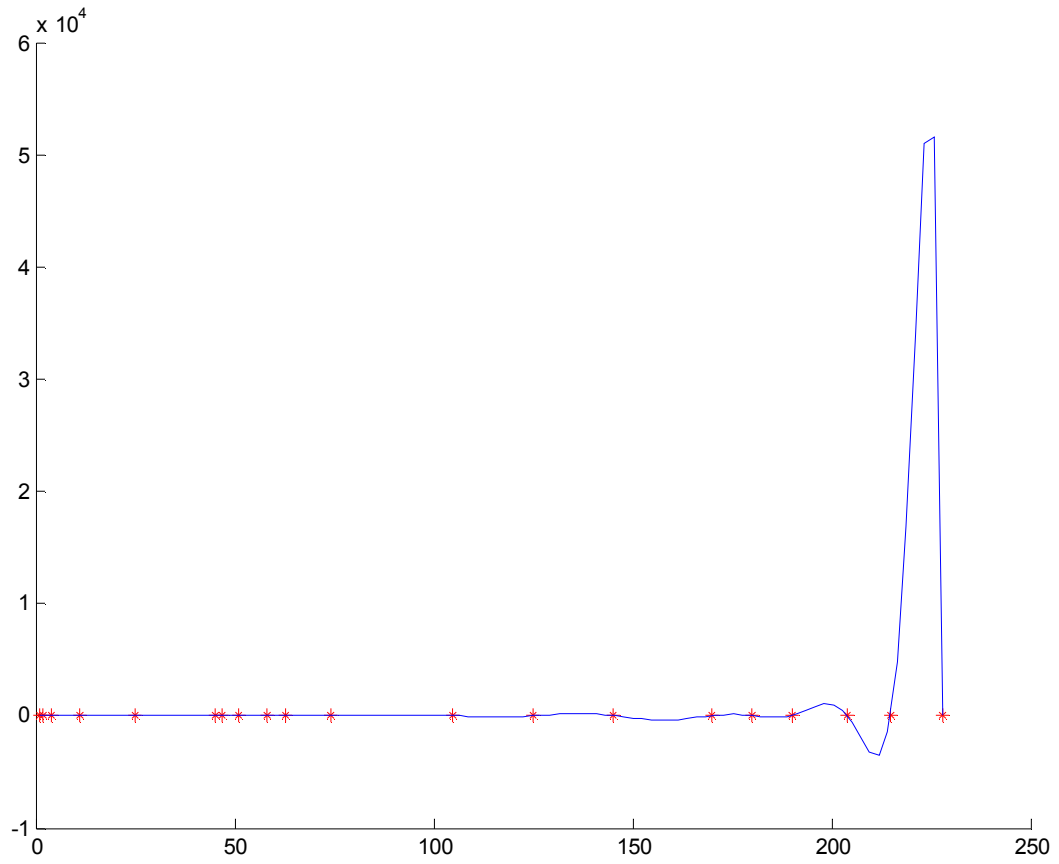
hold on

% Plot dei punti x et y
plot(x,y,'r*')

% Plot del polinomio interpolato
```

```
plot(ls,f,'b')
```

```
hold off
```


```

%           Esercitazione Fitting
%
%           Programma elaborato da
%
%           Giovanni DI CECCA & Virginia BELLINO
%           50 / 887           408 / 466
%
%           http://www.dicecca.net
%
% Punto 2.i

clear % Pulisci memoria

clc % Pulisci schermo

pol=10 % Polinomio di grado 10

% -----
% Costruzione della Funzione di Runge
% Costruzione dei nodi di Chebychev
% -----

% Inserisci i nodi interpolazione
x=linspace(-1,1);

% Nodi di Chebychev su cui costruiamo il polinomio
cheb=cos((pi*(2*[0:pol]))/(2*pol));

% Funzione di Runge
fun=inline(' (1+25*x.^2).^(-1) ');

% Vettore contenente le valutazioni della Funzione di Runge
% nei nodi x
y=feval(fun,x);

% Vettore contenente le valutazioni della funzione di Runge
% sui nodi di Chebychev
y_cheb=feval(fun,cheb);

% -----
% Costruzione del polinomio interpolante
% -----

% Vettore dei coefficienti del polinomio interpolatore
% dei nodi di Chebychev
```

```
p=polyfit(cheb,y_cheb,pol);

% Valutazione dei coefficienti del polinomio interpolatore
% nei punti x di interpolazione
f=polyval(p,x);

% -----
% Plot dei risultati
% -----

hold on

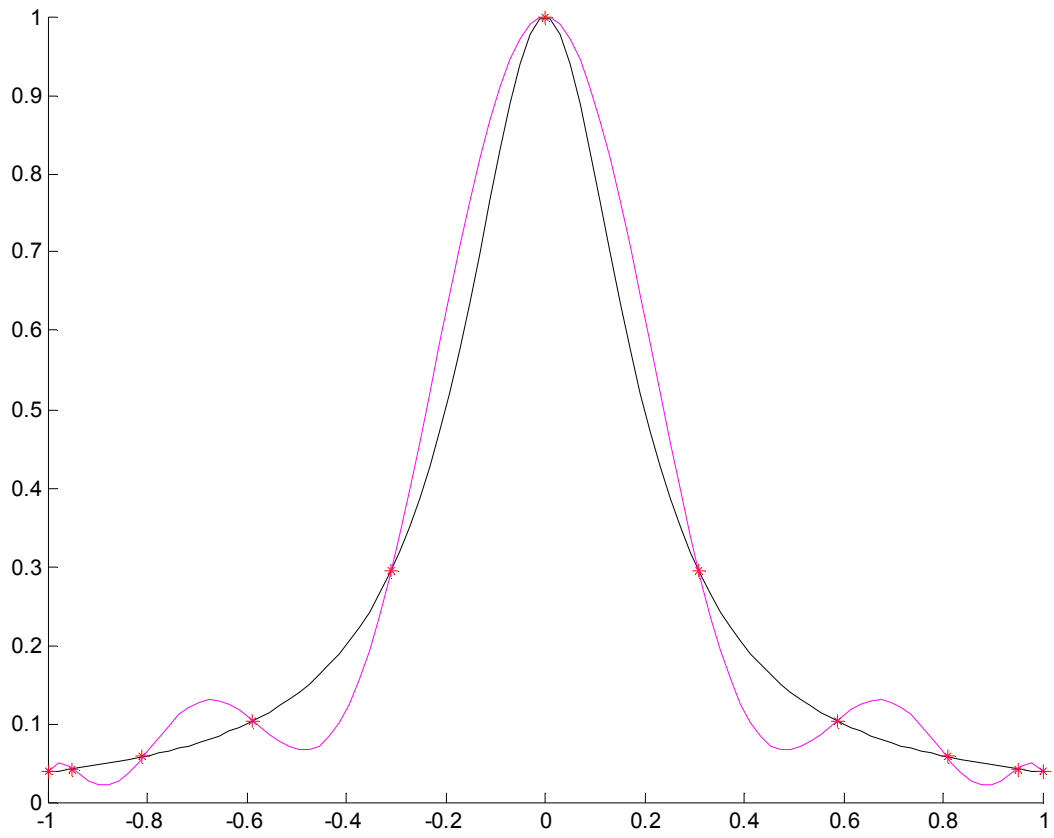
zoom on

% Plot dei punti della Funzione di Runge
plot (x,y,'k')

% Plot dei punti di interpolazione nei
% nodi di Chebychev
plot (cheb,y_cheb,'r*')

% Plot del polinomio interpolante nei nodi di
% interpolazione x et f
plot (x,f,'m')

hold off
```



```
%          Esercitazione Fitting
%
%          Programma elaborato da
%
%          Giovanni DI CECCA & Virginia BELLINO
%          50 / 887          408 / 466
%
%          http://www.dicecca.net
%
% Punto 2.i

clear % Pulisci memoria

clc % Pulisci schermo

pol=20 % Polinomio di grado 20

% -----
% Costruzione della Funzione di Runge
% Costruzione dei nodi di Chebychev
% -----

% Inserisci i nodi intepolazione
x=linspace(-1,1);

% Nodi di Chebychev su cui costruiamo il polinomio
cheb=cos((pi*(2*[0:pol]))/(2*pol));

% Funzione di Runge
fun=inline(' (1+25*x.^2).^(-1) ');

% Vettore contenente le valutazioni della Funzione di Runge
% nei nodi x
y=feval(fun,x);

% Vettore contenente le valutazioni della funzione di Runge
% sui nodi di Chebychev
y_cheb=feval(fun,cheb);

% -----
% Costruzione del polinomio interpolante
% -----

% Vettore dei coefficienti del polinomio interpolatore
% dei nodi di Chebychev
```

```
p=polyfit(cheb,y_cheb,pol);

% Valutazione dei coefficienti del polinomio interpolatore
% nei punti x di interpolazione
f=polyval(p,x);

% -----
% Plot dei risultati
% -----

hold on

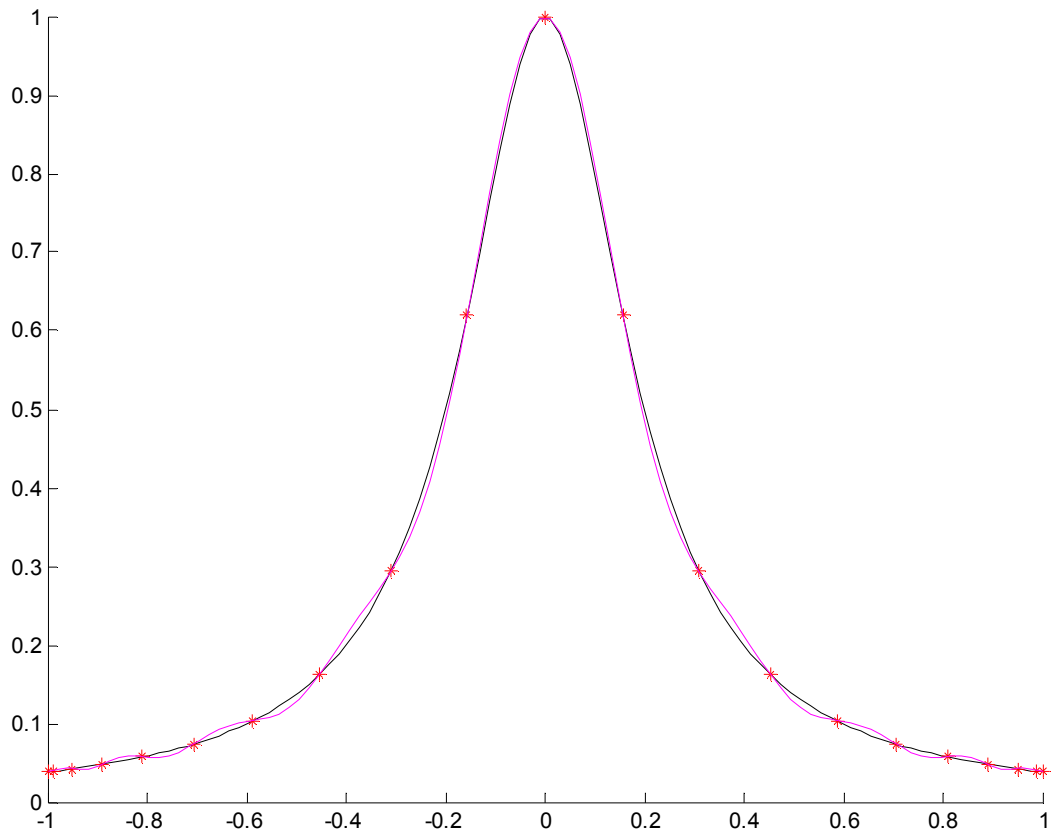
zoom on

% Plot dei punti della Funzione di Runge
plot (x,y,'k')

% Plot dei punti di interpolazione nei
% nodi di Chebychev
plot (cheb,y_cheb,'r*')

% Plot del polinomio interpolante nei nodi di
% interpolazione x et f
plot (x,f,'m')

hold off
```


```
%          Esercitazione Fitting
%
%          Programma elaborato da
%
%          Giovanni DI CECCA & Virginia BELLINO
%          50 / 887          408 / 466
%
%          http://www.dicecca.net
%
% Punto 2.ii
%
% spline

% Pulisci memoria
clear

% Pulisci video
clc

% -----
% Dati ingresso
% -----

x=[1,2,4,11,25,45,47,51,58,63,74,105,125,145,170,180,190,204,215,228]

y=[2,3,4,6,9,12,13,15,20,23,25,24,23,21,19,20,24,33,39,40]

% -----
% Costruzione della Spline cubica
% -----

% Punti equispaziati calcolati in base agli estremi della x
ls=linspace(1,228)

% Funzione di spline per disegnare la figura
sf=spline(x,y,ls);

% -----
% Plot dei risultati
% -----

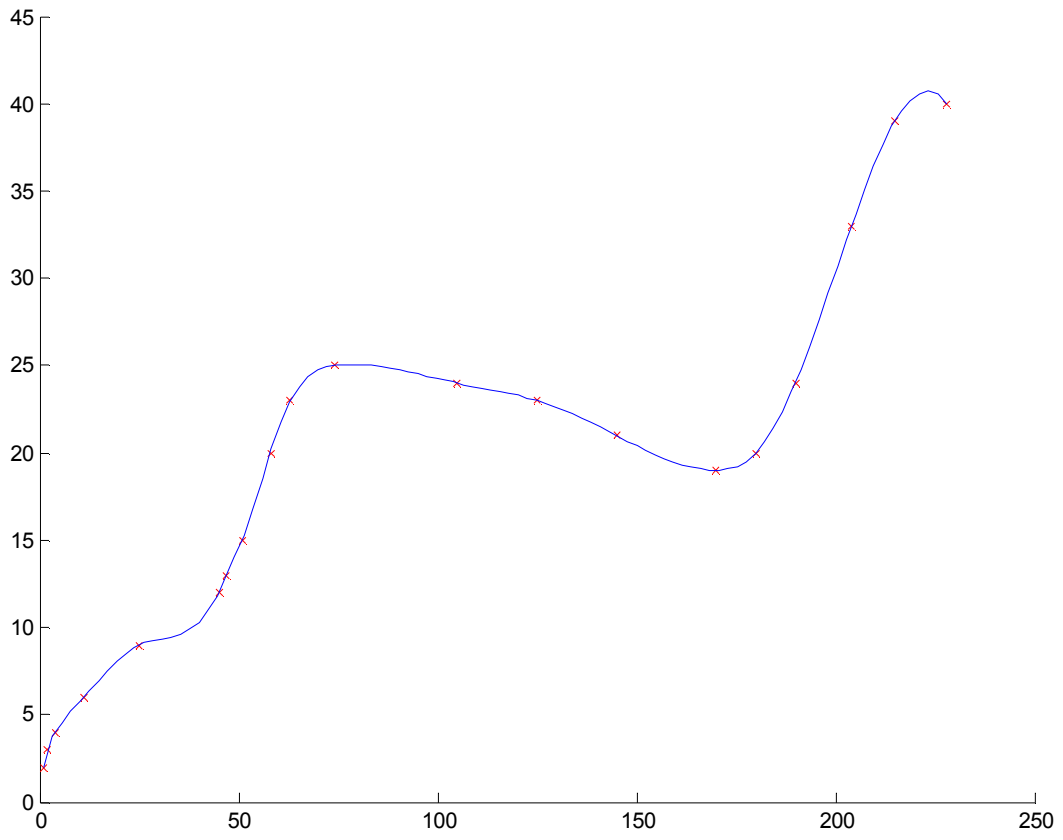
hold on

% Plot dei punti
plot(x,y,'rx')

% Plot della spline
```

```
plot(ls,sf,'b')
```

```
hold off
```



Osservazioni

- **Funzione di Runge**

Sia $f(x)$ una funzione definita in un generico intervallo $[a,b]$ e sia $p(x)$ un polinomio interpolante costruito su $n + 1$ nodi equispaziati appartenenti al medesimo intervallo.

In generale, si potrebbe pensare che, se f è sufficientemente regolare in $[a,b]$, al crescere del grado n il polinomio $p(x)$ approssimi sempre meglio la funzione $f(x)$.

In realtà, ciò non sempre è vero, anzi, può accadere che la successione dei polinomi interpolanti non converga affatto ad $f(x)$, e questo è proprio quanto accade per la funzione di Runge esaminata nella esercitazione.

Infatti, si può osservare che, aumentando il grado del polinomio interpolante, la successione descritta da $p(x)$ non converge affatto come si può notare soprattutto agli estremi, ove sono presenti dei picchi. Questo accade poiché l'errore commesso usando il polinomio di interpolazione dipende in parte anche dalla scelta del tipo di nodi.

Nel caso specifico, si dimostra che la scelta di nodi equispaziati non è ottimale.

La situazione migliora notevolmente se scegliamo come nodi di interpolazione le **ascisse di Chebychev**. Si tratta infatti di nodi che minimizzano l'errore, e ciò appare evidente osservando i plot relativi ai files *cheby10* e *cheby20*.

- **Profilo superiore di un aereo**

In questo caso, partendo da un numero moderatamente grosso di nodi, è stato approssimato l'andamento del profilo superiore di un aereo, dapprima utilizzando un polinomio interpolante (*file aereo.m*), poi ricorrendo ad una spline cubica interpolante (*file splinefly.m*).

Osservando i grafici, si può notare che l'interpolazione polinomiale non è la scelta più ottimale, poiché genera una approssimazione poco attendibile del fenomeno dovuta al fatto che il numero di nodi preso in considerazione è moderatamente grosso.

Utilizzando invece una spline cubica interpolante la situazione migliora notevolmente, poiché questa funzione, racchiudendo in sé le caratteristiche migliori ereditate dalla interpolazione a tratti unite ad una maggiore regolarità, consente di mitigare le oscillazioni, generando così una approssimazione del fenomeno più aderente alla realtà.