Interpolazione polinomiale

Gabriella Puppo

Interpolazione polinomiale

- Matrice di Vandermonde
- Costruzione del polinomio di interpolazione
- Studio dell'errore
- Fenomeno di Runge
- Condizionamento

Matrice di Vandermonde

La matrice di Vandermonde è la matrice dei coefficienti di un problema di interpolazione, e dipende solamente dalla disposizione dei nodi di griglia, X_0, X_1, …, X_m.
E' la matrice (m+1) X (N+1) definita da:

$A(i,j) = x_i^{\,j}$

La matrice di Vandermonde quadrata può essere costruita con la function vander(v), dove v è un vettore che contiene i nodi di griglia. Il risulato è una matrice che ha le colonne disposte in modo opposto rispetto alle convenzioni solite

Esempio

Costruisce la matrice 3 X 3, relativa ai nodi:
$X_0 = 2$, $X_1=3$, $X_2=4$:

```
>> vander([2,3,4])
ans =
    4    2    1
    9    3    1
   16    4    1
```

Griglie di interpolazione

1) Griglia equispaziata, sull'intervallo [a,b] con m+1 nodi:
Esempio con a=1, b=5, m=10:

```
>> a=1; b=5; m=10;
>> h=(b-a)/m;
>> x=a:h:b
x =
  Columns 1 through 7
    1.0000    1.4000    1.8000    2.2000    2.6000    3.0000    3.4000
  Columns 8 through 11
    3.8000    4.2000    4.6000    5.0000
```

2) Griglia di Chebishev

```
function x=x_cheb(n,ab)
% X_CHEB(N,[A,B]): Calcola la griglia di Chebichev con N+1
%          punti, sull'intervallo [A,B]
% X_CHEB(N) Calcola la griglia di Chebichev su [-1,1]
if nargin==1
   a=-1; b=1;
else
   a=ab(1); b=ab(2);
end
for i=1:n+1
   x(i)= (a+b)/2 -(b-a)/2*cos(pi*(i-1)/n);
end
```

Esempio: Costruiamo la griglia di Chebishev sull’intervallo [1,5] con 11 punti:

```
>> x_cheb(10,[1,5])
ans =
  Columns 1 through 7
    1.0000    1.0979    1.3820    1.8244    2.3820    3.0000    3.6180
  Columns 8 through 11
    4.1756    4.6180    4.9021    5.0000
```

Studiamo ora la disposizione dei nodi per la griglia equispaziata e per la griglia di Chebishev, per m=10 e per m=20:

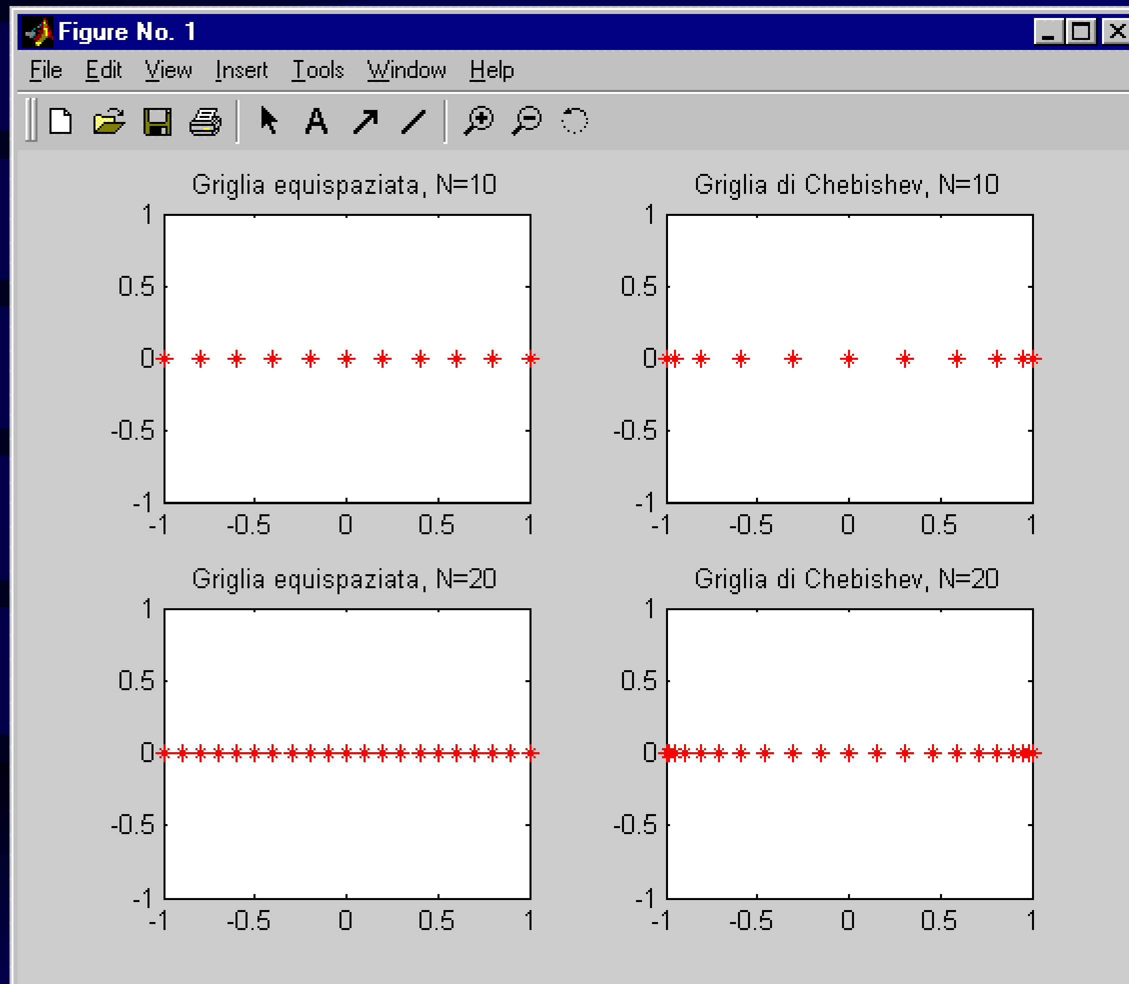


Figure No. 1
File Edit View Insert Tools Window Help
Griglia equispaziata, N=10
Griglia di Chebishev, N=10
Griglia equispaziata, N=20
Griglia di Chebishev, N=20

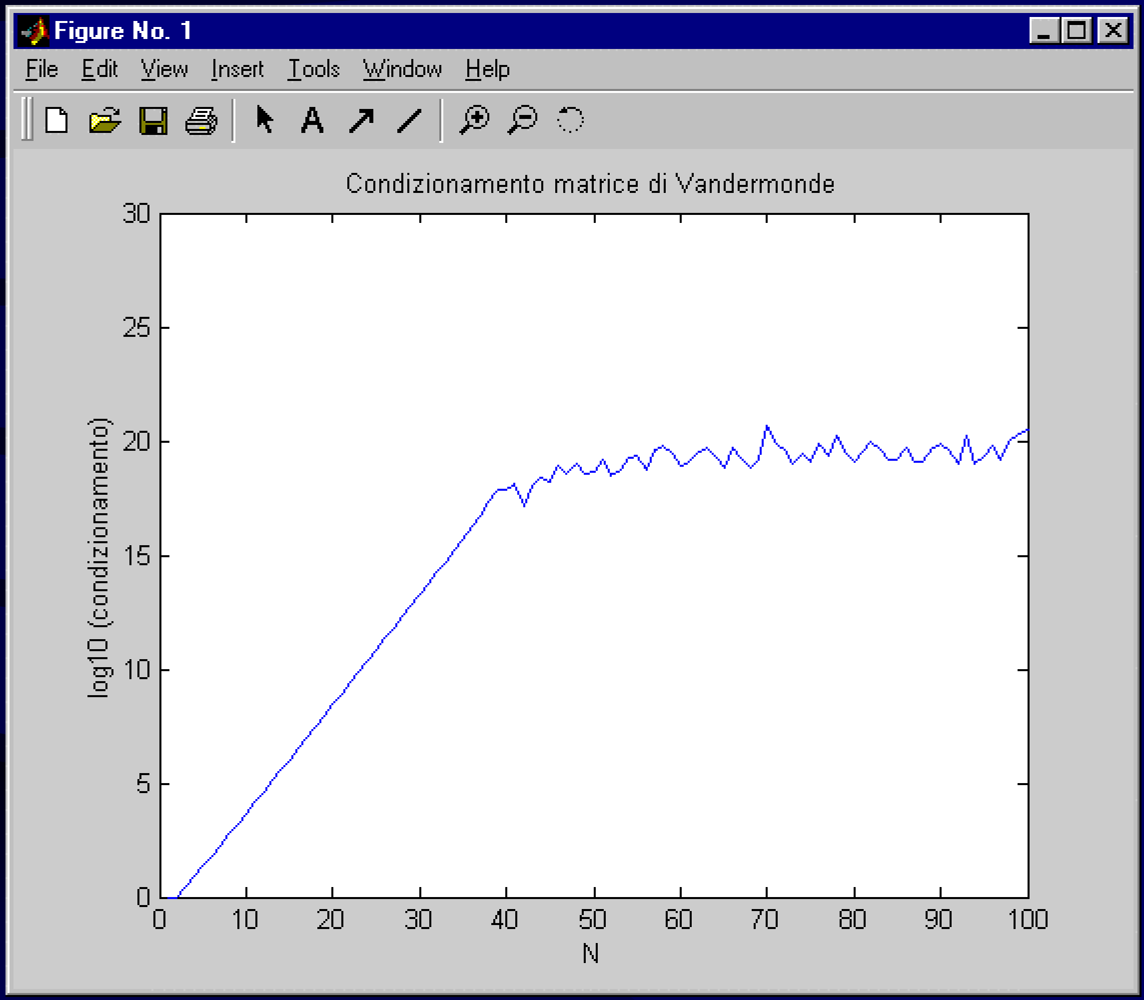
Studio del condizionamento delle matrici di Vandermonde

Costruisco le matrici di Vandermonde, corrispondenti ad una griglia equispaziata contenente N+1 punti, per N che va da 1 a 100 e ne stimo il numero di condizionamento con la function cond per ogni valore di N.
Poi stampo un grafico con i risultati

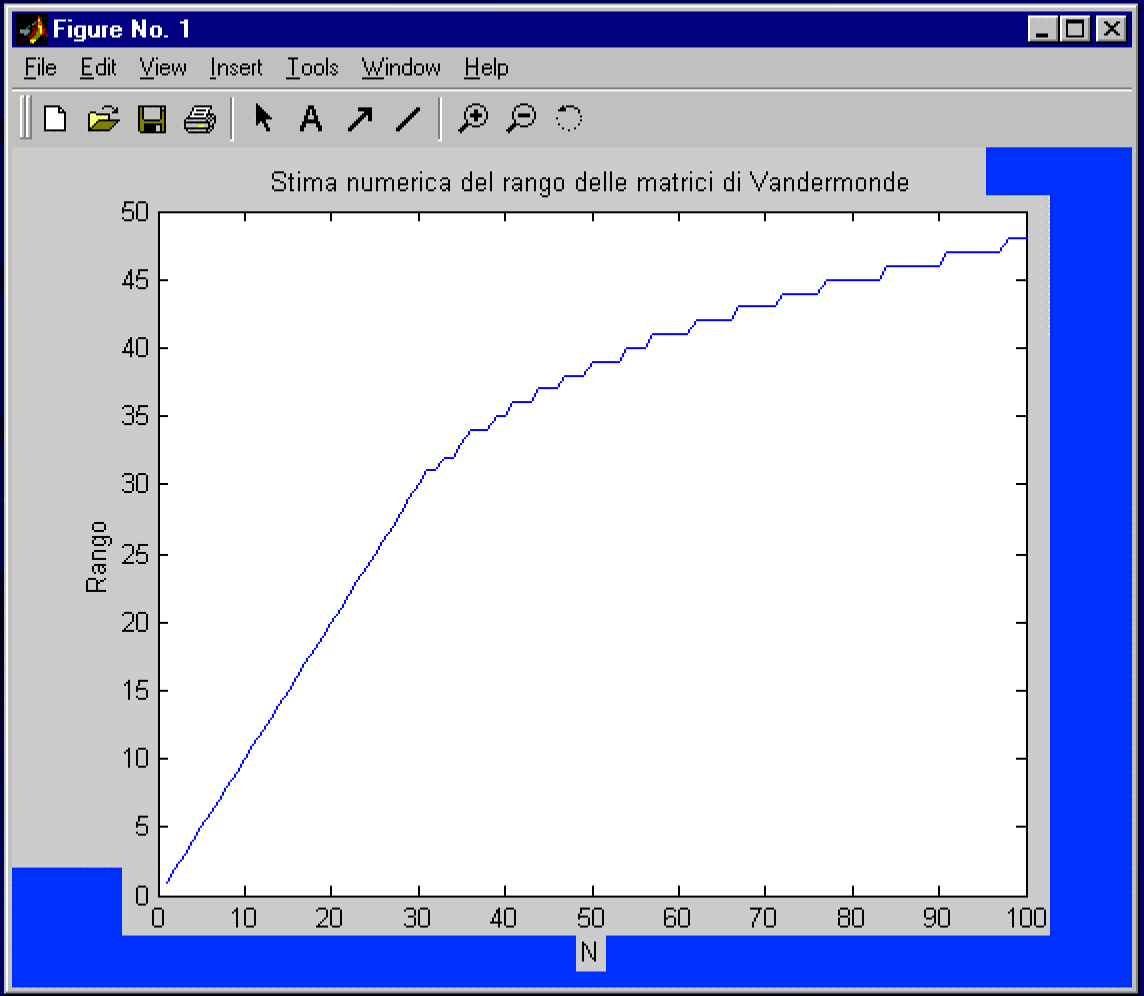
Listato dello script con_vander.m

```
% Studio della stima numerica del condizionamento della
% matrice di Vandermonde costruita su un vettore V di nodi
% uniformemente distribuiti su [-1,1]
%
c(1)=1;
% La piu' piccola matrice che costruisce e' 2X2
for n=1:100
    h=2/n;
    v=-1:h:1;
    a=vander(v);
    c(n+1)=cond(a);
end
plot(log10(c))
```

Come si vede, le matrici di Vandermonde sono mal condizionate



Esercizio: Stimare numericamente il rango delle matrici di Vandermonde, su una griglia equispaziata, per N che va da 1 a 100.



Costruzione del polinomio di interpolazione

Per costruire il polinomio di interpolazione devo:

- *Costruire la tabella delle differenze divise, e memorizzarne la diagonale superiore*
- *Usare le differenze divise per valutare il polinomio di interpolazione in corrispondenza dei punti richiesti*

Calcolo delle differenze divise

Costruisco una function che:

- *In input ha i vettori x ed f, contenenti i nodi di interpolazioni ed i valori da interpolare.*
- *In input ha il vettore contenente le differenze divise.*

Listato della function *dif_divise*

```
function dif=dif_divise(x,f)
% DIF=DIF_DIVISE(X,F) Calcola la tabella delle differenze
%         divise di Newton, per le coppie di punti (X(I),F(I)).
%         Immagazzina solo la diagonale superiore della tabella,
%         che viene fornita in output nel vettore DIF.
n=length(x);
dif=f;
for  i=2:n
    for j=n:-1:i
        dif(j)=(dif(j)-dif(j-1))/(x(j)-x(j-i+1));
    end
end
```

Valutazione del polinomio di interpolazione

Costruisco una function che:

- *In input ha il vettore dif delle differenze divise, il vettore x dei nodi di interpolazione, e un vettore xx contenente i punti sui quali valutare il polinomio*
- *In output ha il vettore p, contenente i valori del polinomio sui punti xx*
- *Per calcolare il polinomio, usa la regola di Horner*

Listato della function *pol_interp*

```
function p=pol_interp(x,dif,xx)
%
n=length(x); neval=length(xx);
for je=1:neval
    % Calcola il polinomio usando la regola di Horner
    p(je)=dif(n);
    for j=n:-1:2
        p(je)=p(je)*(xx(je) -x(j-1)) +dif(j-1);
    end
end
```

Esempio

Calcolare il polinomio di interpolazione di grado N=5 della funzione:

$f(x) = \exp(x) * \sin(2x)$,

usando una griglia equispaziata sull'intervallo [0,2].

Visualizzare i risultati, riportando sullo stesso grafico:

- *la funzione f(x)*
- *il polinomio di interpolazione p(x)*

Posso ottenere i risultati richiesti usando i comandi:

```
>> f=inline('exp(x).*sin(2*x)');
>> n=5;
>> x=0:2/5:2;
>> fx=f(x);
>> dif=dif_divise(x,fx);
>> xx=0:0.01:2;
>> p=pol_interp(x,dif,xx);
>> plot(xx,f(xx))
>> hold
Current plot held
>> plot(xx,p,'g')
```

Ottengo:

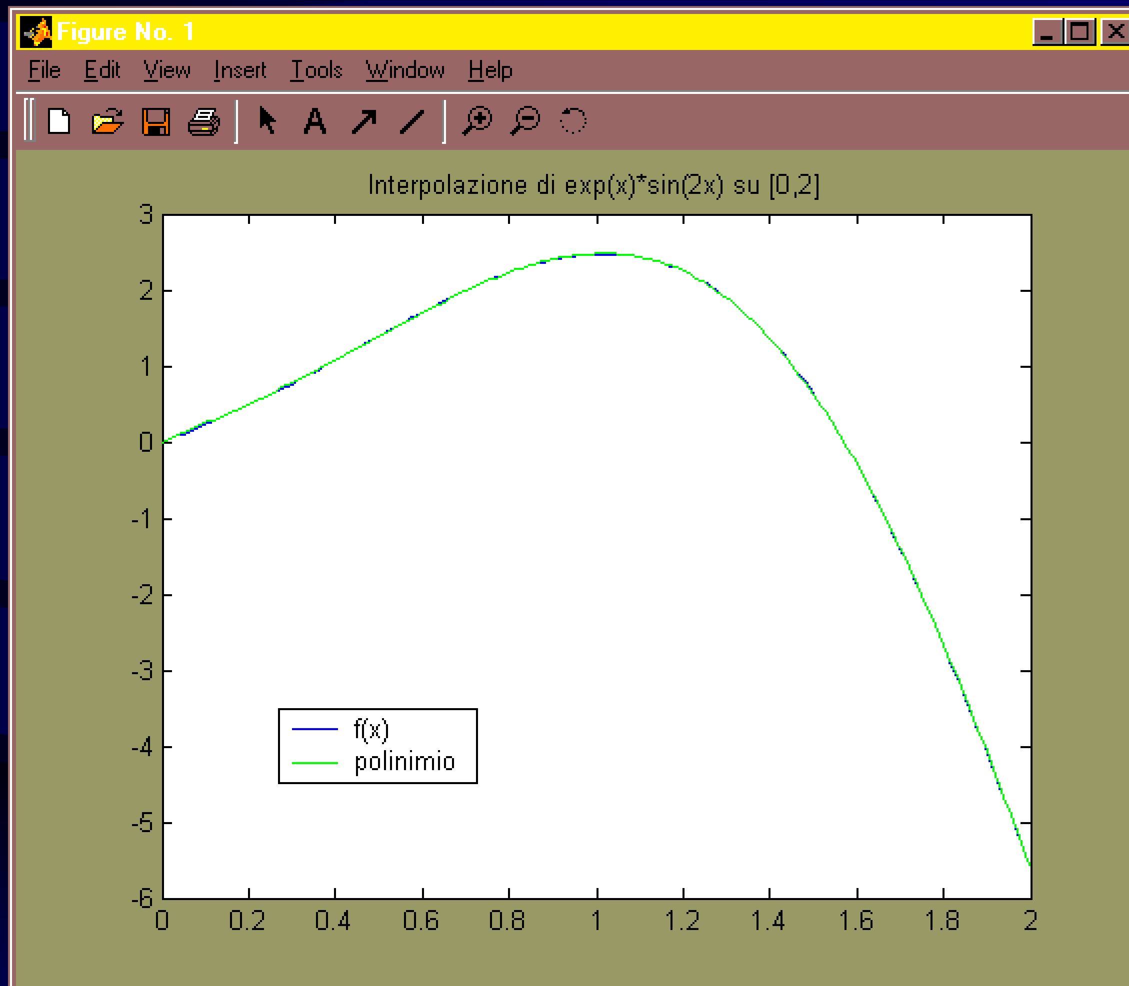


Figure No. 1
File Edit View Insert Tools Window Help
Interpolazione di exp(x)*sin(2x) su [0,2]
3
2
1
0
-1
-2
-3
-4
-5
-6
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
f(x)
polinimio

Studio dell'errore

Per stimare l'errore di interpolazione fra una funzione f(x) ed il suo polinomio di interpolazione P(x), di grado N, costruito sulla griglia X, devo:

- Calcolare il polinomio di interpolazione P(x)
- Preparare una griglia XX più fitta di quella di interpolazione
- Calcolare sia f(x) che P(x) sulla griglia XX
- Calcolare la norma infinito del vettore f(XX) - P(XX).

Function errore.m

La function errore ha due modalità di funzionamento:

- se voglio solo il calcolo dell'errore, chiamo: errore(f,x), dove f deve essere una stringa
- se voglio sia il calcolo dell'errore che il grafico di f e del polinomio (con in più i punti di interpolazione), chiamo: errore(f,x,1)

```
function err= errore(f,x,dummy)
% ERR=ERRORE(F,X) Calcola l'errore nella norma infinito
%       fra F ed il polinomio P di interpolazione di F,
%       costruito sulla griglia X
%   Se ci sono tre argomenti in input, stampa anche
%   il grafico di F e di P
n=length(x)-1;
fx=feval(f,x);
dif=dif_divise(x,fx);
% Costruisce la griglia su cui calcola l'errore
a=x(1); b=x(n+1); h=(b-a)/200;
xx=a:h:b;
fxx=feval(f,xx);
pxx=pol_interp(x,dif,xx);
err = norm(fxx-pxx,inf);
if nargin==3
   % Stampa il grafico di F(XX), P(XX) e dei punti di  interpolazione
   plot(xx,fxx), hold on;
   plot(xx,pxx,'g')
   plot(x,fx,'r*')
end
```

Esempio

- Calcolare l'errore di interpolazione per la funzione f(x)=exp(x)*cos(4x), su [-3,3], usando una griglia equispaziata con N=5, 10, 20 e 40 nodi.
- Stessa cosa per la funzione f(x) = abs(x-1)
- Rifare i conti per la griglia di Chebishev

Conclusioni

L'errore dipende:

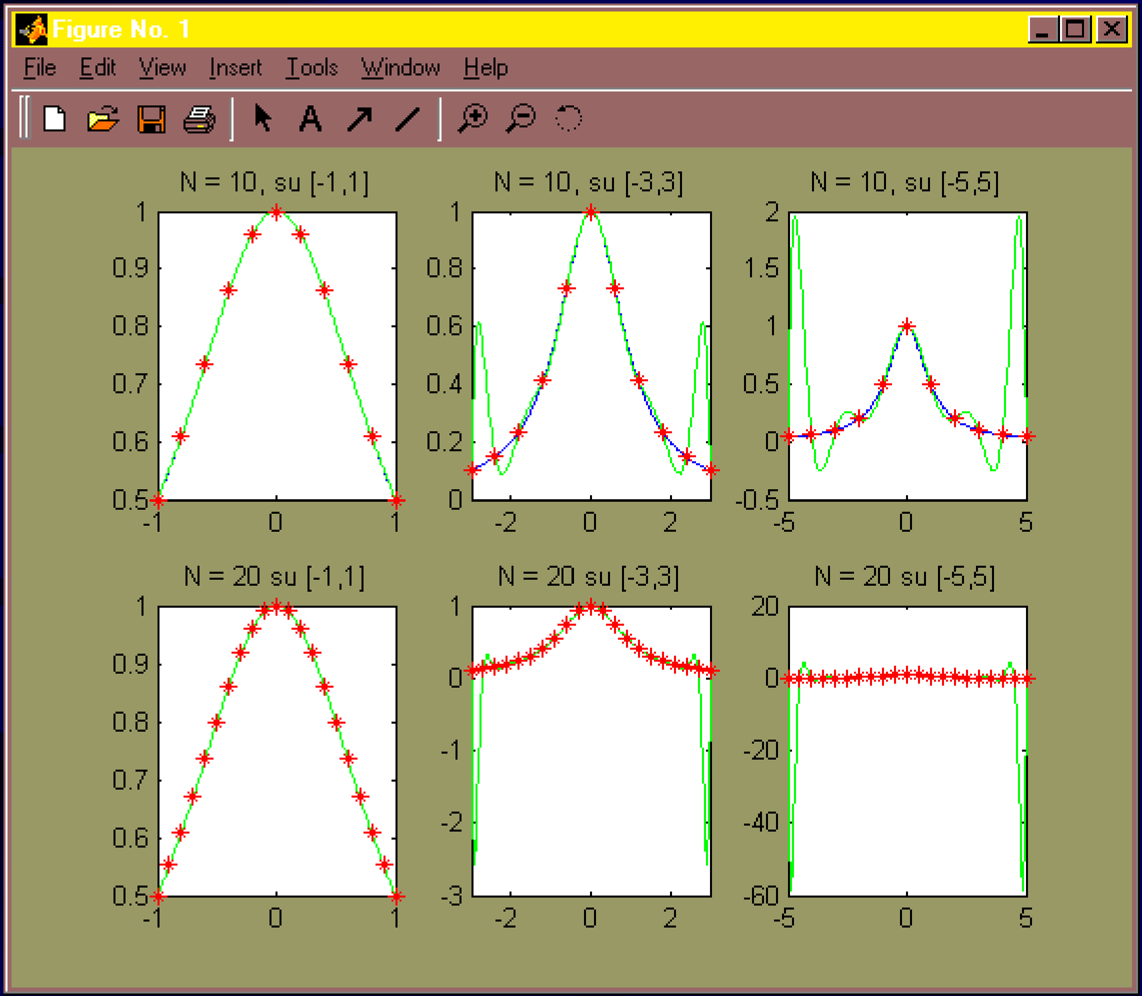
- dalla regolarità della funzione
- dalla disposizione dei nodi della griglia

Fenomeno di Runge

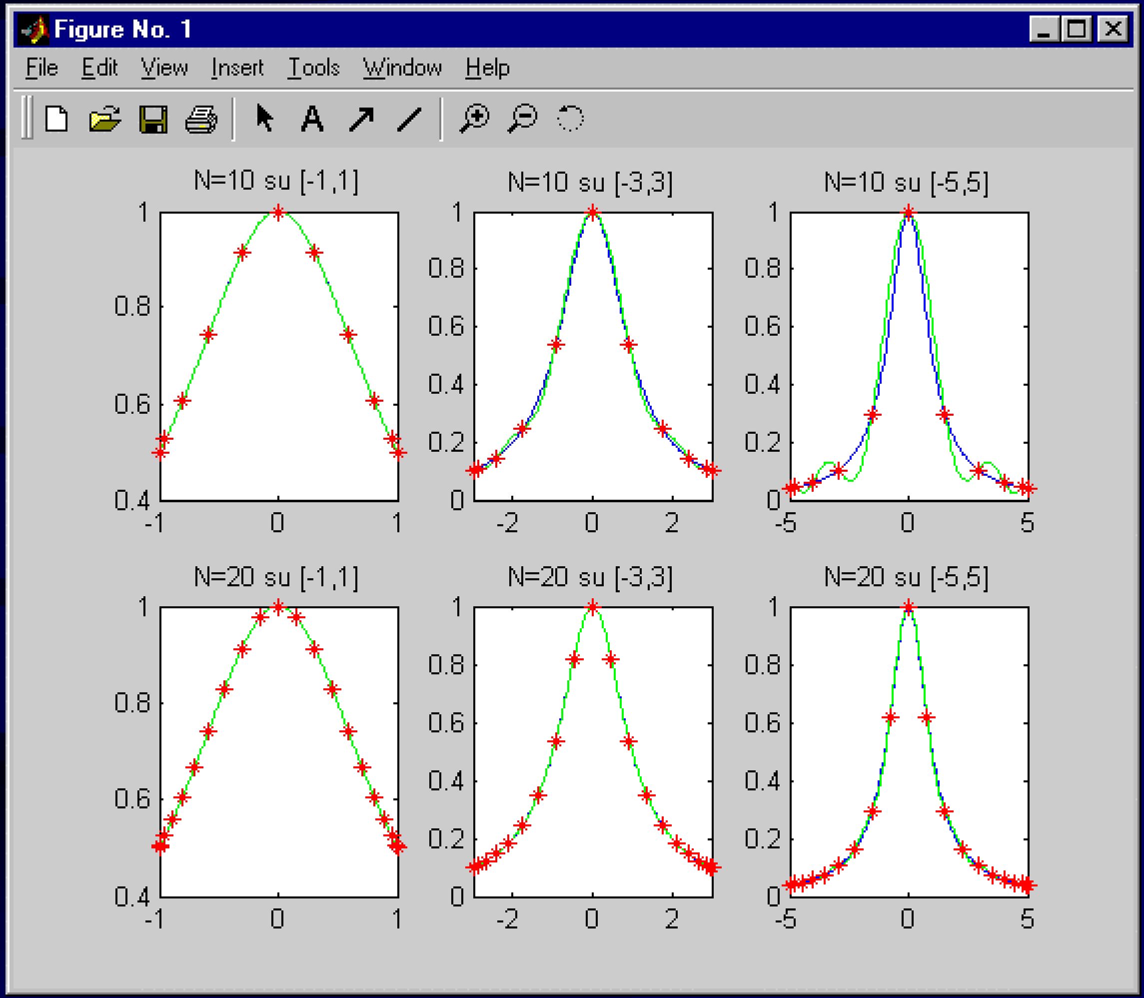
Interpolo la funzione
$f(x) = 1 / (1 + x^2)$
su intervalli del tipo $[-a,a]$, con $a=1$, $a=2$, $a=3$, $a=5$,
usando polinomi di grado $N = 10$ e $N = 20$.

Considero:
1) griglie equispaziate;
2) griglie di Chebishev.

Griglia equispaziata



Griglia di Chebishev



Condizionamento

Per studiare il condizionamento di un problema di interpolazione:

- *Definisco una funzione f(x) e una griglia X*
- *Costruisco il polinomio P(x) che interpola i dati (X, f(X))*
- *Disegno il grafico di f(x) e di P(x)*
- *Perturbo i dati f(X), per esempio*
  $F(X)=f(X)+10^{K} * r$ $\quad\quad$ *K=-3, -4, ecc.*
  (dove r è un vettore di numeri casuali)
- *Calcolo il polinomio di interpolazione dei dati perturbati e ne disegno il grafico*