

FORMULA DI NEWTON DEL POLINOMIO DI INTERPOLAZIONE

Siano $x_i, i=0, \dots, m$ $m+1$ punti distinti.

Siano $y_i = f(x_i), i=0, \dots, m$ $m+1$ valori distinti o meno.

Si dimostra che il polinomio che soddisfa le condizioni di interpolazione

$$p_m(x_i) = y_i, \quad i=0, \dots, m$$

può essere scritto nel seguente modo

$$\begin{aligned} p_m(x) = & f(x_0) \\ & + f[x_0, x_1] \cdot (x - x_0) \\ & + f[x_0, x_1, x_2] \cdot (x - x_0)(x - x_1) \\ & + \dots \\ & + f[x_0, x_1, x_2, \dots, x_{m-1}, x_m] \cdot (x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_{m-1}) \end{aligned}$$

dove $f[x_0, x_1], \dots, f[x_0, x_1, \dots, x_m]$ sono le differenze divise definite dalle formule ricorsive seguenti:

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}, \quad \text{differenze divise di ordine 1}$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}, \quad \text{diff. div. di ordine 2}$$

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}, \quad \text{diff. div. di ordine 3}$$

$$\vdots$$

$$f[x_0, x_1, \dots, x_{m-1}, x_m] = \frac{f[x_1, \dots, x_m] - f[x_0, \dots, x_{m-1}]}{x_m - x_0}, \quad \text{diff. div. di ordine } m$$

Scriviamo $p_1(x)$:

$$p_1(x) = f(x_0) + f[x_0, x_1](x - x_0)$$

Vediamo che soddisfa le condizioni $p_1(x_0) = f(x_0), p_1(x_1) = f(x_1)$

Vediamo che soddisfa le condizioni

$$p_1(x_0) = f(x_0), p_1(x_1) = f(x_1)$$

$$p_1(x_0) = f(x_0) + \cancel{f[x_0, x_1](x_0 - x_0)} = f(x_0)$$

$$p_1(x_1) = f(x_0) + f[x_0, x_1](x_1 - x_0) =$$

$$= \cancel{f(x_0)} + \frac{f(x_1) - \cancel{f(x_0)}}{x_1 - x_0} (x_1 - x_0) = f(x_1)$$

Analogamente, lo si dimostra per $p_2(x)$.

È consuetudine scrivere le differenze divise che servono nel calcolo del polinomio usando una tabella

x_0	$f(x_0)$		
x_1	$f(x_1)$	$f[x_0, x_1]$	
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$

$$p_2(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1)$$

x_0	$f(x_0)$			
x_1	$f(x_1)$	$f[x_0, x_1]$		
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
x_3	$f(x_3)$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$

$$p_3(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) +$$

$$+ f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) = p_3(x)$$

Però

- (•) il polinomio si legge sulla diagonale superiore;
- (•) per passare da $p_2(x)$ a $p_3(x)$ basta aggiungere solo la diagonale inferiore delle diff. divise. Questo non va (e caso né con Vandermonde né con Lagrange dove l'aggiunta

di un modo richiedere la ripetizione dell'intero calcolo.

ESEMPIO Costruiamo il polinomio di Newton per

$$\begin{array}{l|l}
 x_0 = -1 & 2 \\
 x_1 = 0 & 1 \\
 x_2 = 1 & 3
 \end{array}
 \quad
 \begin{array}{l}
 p[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1 - 2}{0 - (-1)} = -1 \\
 p[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{3 - 1}{1 - 0} = 2 \\
 p[x_0, x_1, x_2] = \frac{p[x_1, x_2] - p[x_0, x_1]}{x_2 - x_0} = \frac{2 - (-1)}{1 - (-1)} = \frac{3}{2}
 \end{array}$$

Il polinomio di interpolazione è

$$\begin{aligned}
 p_2(x) &= f(x_0) + p[x_0, x_1](x - x_0) + p[x_0, x_1, x_2](x - x_0)(x - x_1) \\
 &= 2 + (-1)(x - (-1)) + \frac{3}{2}(x - (-1))(x - 0) \\
 &= 2 - x - 1 + \frac{3}{2}x^2 + \frac{3}{2}x = \frac{3}{2}x^2 + \frac{1}{2}x + 1
 \end{aligned}$$

Verifichiamo che soddisfa le condizioni di interpolazione:

$$p_2(x_0) = \frac{3}{2} \cdot (-1)^2 + \frac{1}{2} \cdot (-1) + 1 = \frac{3}{2} - \frac{1}{2} + 1 = 2 = f(x_0) \quad \checkmark$$

$$p_2(x_1) = \frac{3}{2} \cdot 0^2 + \frac{1}{2} \cdot 0 + 1 = 1 = f(x_1) \quad \checkmark$$

$$p_2(x_2) = \frac{3}{2} \cdot 1^2 + \frac{1}{2} \cdot 1 + 1 = 3 = f(x_2) \quad \checkmark$$

OSS. Nel caso in cui abbiamo già la tabella associata ai nodi $x_0, \dots, x_m$ e voglio aggiungere un altro nodo x_{m+1} lo posso mettere SEMPRE alla fine della tabella indipendentemente dal valore di x_{m+1} rispetto a $x_0, \dots, x_m$.

Infatti si dimostra che

$$p[x_0, x_1, \dots, x_k] = p[x_{i_0}, x_{i_1}, \dots, x_{i_k}]$$

dove $i_0, i_1, \dots, i_k$ è una qualsiasi permutazione di $0, \dots, k$.

Ad esempio per $k=2$ è

$$p[x_2, x_1] = p[x_1, x_0] \quad \text{immediato!}$$

Per $k=3$ è

$$\begin{aligned} p[x_2, x_1, x_0] &= p[x_2, x_0, x_1] = p[x_1, x_2, x_0] = p[x_1, x_0, x_2] = \\ &= p[x_0, x_2, x_1] = p[x_0, x_1, x_2] \end{aligned}$$

COMPORTAMENTO DEI POLINOMI DI INTERPOLAZIONE AL CRESCERE DEL GRADO m .

Spesso abbiamo la necessità di sostituire una funzione $f(x)$, $x \in [a, b]$ con una funzione più semplice che ben rappresenta f in $[a, b]$. Tipicamente la funzione più semplice è il polinomio di interpolazione su un insieme di nodi x_i , $i=0, \dots, m$.

Ciò fatto, esegue ogni operazione richiesta su f sul polinomio di interpolazione. Ad esempio

$$f(\bar{x}) \approx p_m(\bar{x})$$

$$f'(\bar{x}) \approx p_m'(\bar{x})$$

$$\int_a^b f(x) dx \approx \int_a^b p_m(x) dx$$

Pero, devo tenere sotto controllo l'errore

$$E_m(x) = f(x) - p_m(x)$$

che deve "essere piccolo" $\forall x \in [a, b]$.

Valgono una serie di risultati.

PROP. Siano $x_i, i=0, \dots, m$ punti distinti di $[a, b]$ con $a = x_0, b = x_m$.

Supponiamo $f \in C^{m+1}([a, b])$.

Allora esiste $\xi \in (a, b)$ tale che

$$E_m(x) = \prod_{i=0}^m (x - x_i) \frac{f^{(m+1)}(\xi)}{(m+1)!}$$

dove ξ dipende da x .

Dato che ξ non è noto, di solito si dà una maggiorazione del valore assoluto dell'errore:

$$|E_m(x)| = \frac{\left| \prod_{i=0}^m (x - x_i) \right|}{(m+1)!} \cdot \left| f^{(m+1)}(\xi) \right| \leq M_{m+1} \leq \frac{M_{m+1}}{(m+1)!} \left| \prod_{i=0}^m (x - x_i) \right|$$

dove

$$M_{m+1} = \max_{x \in [a, b]} \left| f^{(m+1)}(x) \right|$$

L'errore dipende dai nodi x_i e dal loro numero $m+1$. Ne segue:

(•) esistono distribuzioni di nodi che minimizzano l'errore?

(•) come si comporta $E_m(x)$ al crescere di m ?

Non sempre al crescere di m l'errore cala!!

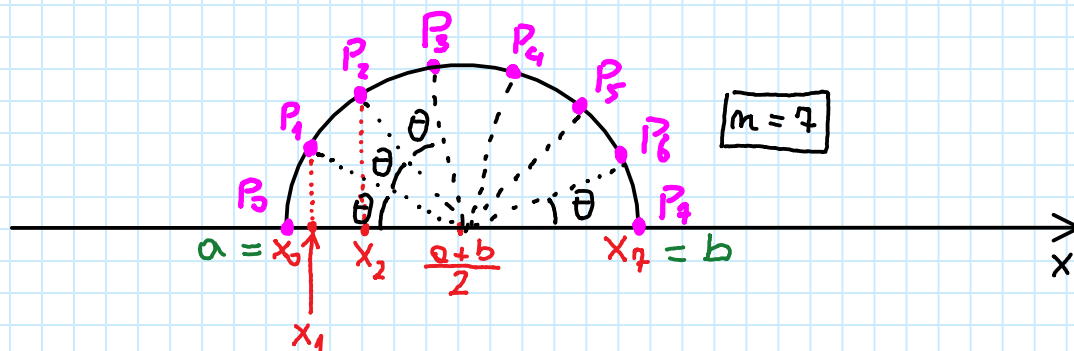
Una scelta di nodi per i quali

$$\lim_{m \rightarrow +\infty} |E_m(x)| = 0 \quad \forall x \in [a, b]$$

sono quelli di Chebyshev definiti da

$$x_i = \frac{a+b}{2} - \frac{b-a}{2} \cos\left(i \frac{\pi}{m}\right), \quad i = 0, \dots, m$$

I nodi x_i sono le ascisse dei punti P_i equispaziati di $\theta = \pi/m$ sulla circonferenza di centro $\frac{a+b}{2}$ e raggio $\frac{b-a}{2}$.
Ad esempio, per $m=7$ abbiamo



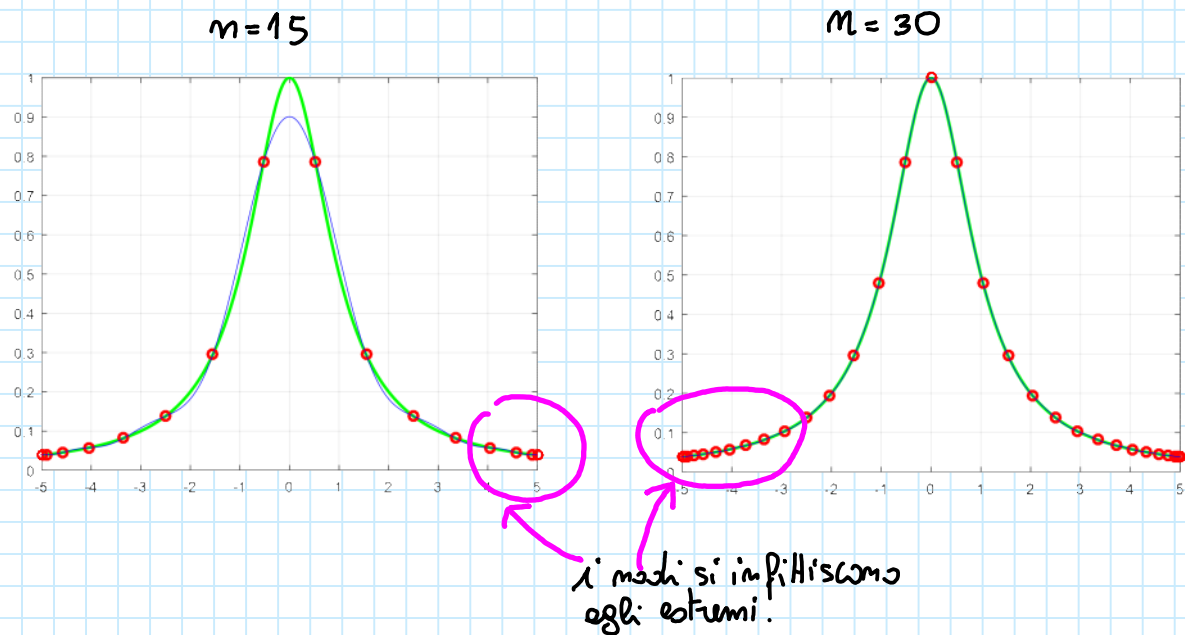
Vediamo il comportamento tenuto dai polinomi di interpolazione quando (questo esempio è dovuto a Runge)

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5]$$

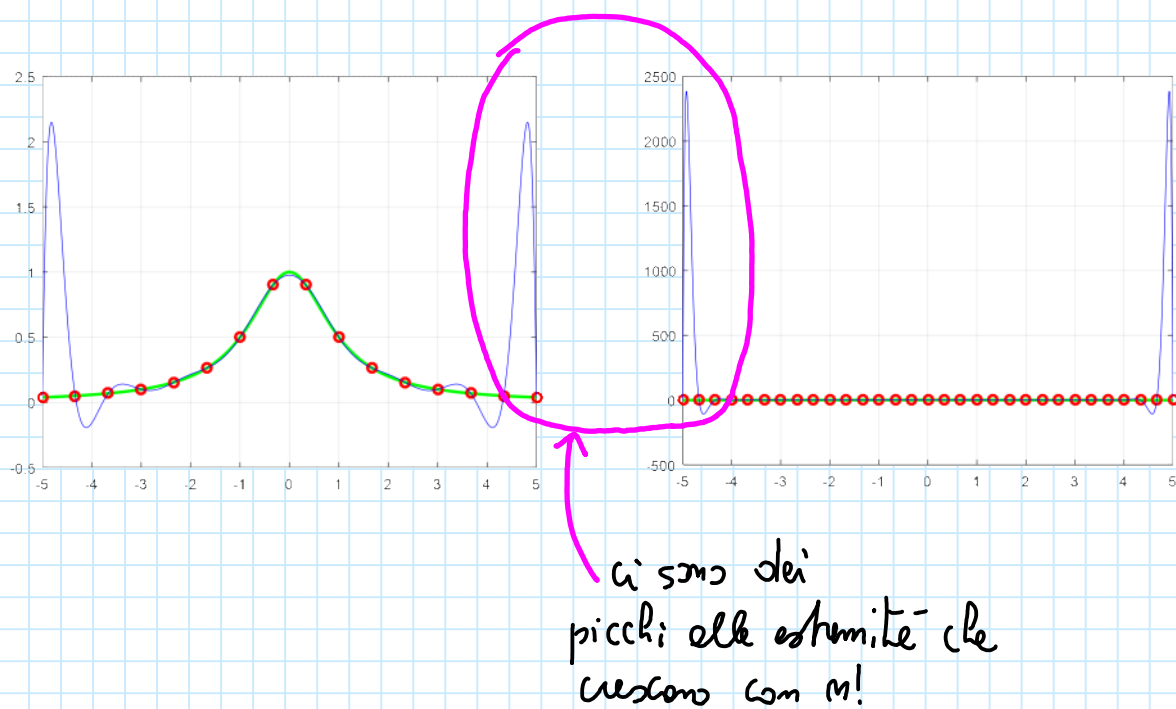
con nodi equispaziati e con i nodi di Chebyshev.

Come si può vedere dai grafici della pagina che segue, al crescere del grado del polinomio l'errore cala con i nodi di Chebyshev mentre cresce (per effetto dei due picchi alle estremità) nel caso dei nodi equispaziati.

Nodi di Chebyshev



Nodi equispaziati



Nella pagina seguente sono riportati i codici Matlab/Octave per le realizzazioni delle figure precedenti.

```
% script di prova della interpolazione

% funzione e suo grafico in [a, b]
a = -5;
b = 5;
f = inline('1./(1+x.^2)'); % funzione di Runge
x = linspace(a,b,1000);
fx = feval(f,x);
plot(x,fx,'g','lineWidth',2)
grid on

% definisco i nodi equispaziati
n = 30;
xi = linspace(a, b, n+1);
fxi = feval(f,xi);
hold on
plot(xi,fxi,'or','lineWidth',2)

% trovo il polinomio di Lagrange e lo disegno
px = lagrange(xi,fxi,x);
plot(x,px,'b')
```

interpolazione su nodi
equispaziati

```
% script di prova della interpolazione

% funzione e suo grafico in [a, b]
a = -5;
b = 5;
f = inline('1./(1+x.^2)'); % funzione di Runge
x = linspace(a,b,1000);
fx = feval(f,x);
plot(x,fx,'g','lineWidth',2)
grid on

% definisco i nodi equispaziati
n = 15;
xi = (a+b)/2-(b-a)/2*cos((pi/n)*(0:n));
fxi = feval(f,xi);
hold on
plot(xi,fxi,'or','lineWidth',2)

% trovo il polinomio di Lagrange e lo disegno
px = lagrange(xi,fxi,x);
plot(x,px,'b')
```

interpolazione su nodi
di Chebyshev

```
function y = lagrange(nodi,fxi,x)
%LAGRANGE polinomio di interpolazione di Lagrange sui nodi
% Y = LAGRANGE(NODI,X) valuta il polinomio di interpolazione di Lagrange
% associato ai nodi x0,...,xn specificati nel vettore nodi nei punti
% specificati nel vettore x. Il vettore y dei risultati ha le stesse
% dimensioni di x.

y = zeros( size(x) );
for i=1:length(nodi)
    y = y+fxi(i)*lagpoly(i,nodi,x);
end
```

```
function y = lagpoly(i,nodi,x)
%LAGPOLY polinomio di Lagrange associato al nodo xi valutato in x
% Y = LAGPOLY(I,NODI,X) valuta il polinomio di Lagrange Li(x) associato
% al nodo i-esimo dei nodi x0, x1, ..., xn nei punti specificati
% dal vettore x e restituisce i valori nel vettore y (stesse dimensioni
% di x).
%
n = length(nodi);

y = ones( size(x) );
for r = 1:i-1
    y = y.*(x-nodi(r))./(nodi(i)-nodi(r));
end

for r = i+1:n
    y = y.*(x-nodi(r))./(nodi(i)-nodi(r));
end
```