

I polinomi

- ▶ Un polinomio in Matlab è dato da un vettore che contiene i suoi coefficienti ordinati da a_n fino ad a_0 .
- ▶ Per valutare un polinomio in uno o più punti si usa il comando `polyval`:

```
>> y = polyval(p,x)
```

- ▶ x è un vettore dove si specificano le ascisse nelle quale si vuole valutare il polinomio p .
- ▶ y è un vettore che contiene i valori di p in x .

I polinomi

- ▶ Il comando `roots(p)` approssima le radici del polinomio p .
- ▶ I comandi `polyint(p)` e `polyder(p)` calcolano rispettivamente i coefficienti di una primitiva (quella che si annulla in $x = 0$) e della derivata di p .
- ▶ Se x e y sono due vettori di $n + 1$ componenti, il comando `p=polyfit(x,y,m)` calcola i coefficienti del polinomio di grado m che approssima le $n + 1$ coppie di valori $\{(x_i, y_i)\}_{i=0}^n$ nel senso dei minimi quadrati.

Esercizio - Controesempio di Runge

Scrivere uno script di Matlab che:

- ▶ calcoli il polinomio p che interpola la funzione $f(x) = \frac{1}{1+x^2}$ in $n + 1$ punti equispaziati dell'intervallo $[-5,5]$ inclusi gli estremi;
- ▶ disegni il grafico della funzione f e del polinomio p nell'intervallo $[-5,5]$.

Calcolare il polinomio interpolatore di f in nodi equispaziati per $n = 6, 8, 10, 12$ e commentare i risultati.

```
n=input('Grado: ');  
x=linspace(-5,5,n+1); % n+1 nodi equispaziati  
y=1./(1+x.^2);  
p=polyfit(x,y,n);  
xx=linspace(-5,5); % punti per il grafico  
pxx=polyval(p,xx);  
fxx=1./(1+xx.^2);  
plot(xx,fxx,xx,pxx,x,y,'*');  
legend('Funzione','Polinomio', 'Dati')
```

I nodi di Chebyshev

- ▶ Nell'intervallo $[-1,1]$ i nodi di Chebyshev sono

$$\hat{x}_i = -\cos(\pi i/n), \quad i = 0, \dots, n.$$

- ▶ Su un arbitrario intervallo $[a,b]$ sono i nodi

$$x_i = \frac{a+b}{2} + \frac{b-a}{2} \hat{x}_i.$$

Esercizio

- ▶ Scrivere uno script di Matlab che calcoli il polinomio q che interpola la funzione $f(x) = \frac{1}{1+x^2}$ nei $n+1$ nodi di Chebyshev dell'intervallo $[-5,5]$ e disegni il grafico della funzione f e del polinomio q nell'intervallo $[-5,5]$.
- ▶ Calcolare il polinomio interpolatore di f nei nodi di Chebyshev per $n = 6, 8, 10, 12$ e commentare i risultati.

```
n=input('Grado: ');
x=-5*cos(pi*[0:n]/n); % n+1 nodi di Chebyshev
y=1./(1+x.^2);
p=polyfit(x,y,n);
xx=linspace(-5,5); % punti per il grafico
pxx=polyval(p,xx);
fxx=1./(1+xx.^2);
plot(xx,fxx,xx,pxx,x,y,'*');
legend('Funzione','Polinomio','Dati')
```

Integrazione numerica

Il comando quad.

```
>> Iapp=quad(fun,a,b)
```

- ▶ Approssima il valore di $I := \int_a^b f(x) dx$ usando la formula di Simpson adattiva, con errore stimato minore di 10^{-6} .
- ▶ `fun` può essere una funzione oppure una stringa.

Esercizio

Scrivere una funzione di Matlab che riceva in ingresso una funzione f un intervallo $[a, b]$ e una tolleranza $toll$ e restituisca il valore approssimato di

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

calcolato usando il metodo dei trapezi con errore stimato minore di $toll$ e il numero di sottointervalli che ha usato per calcolare questa approssimazione.


```

function [intapp,n]=trapezi(fun,a,b,toll)
n=1;
% CALCOLO DEL PRIMO INTOLD
H=b-a;
x=[a b];
y=feval(fun,x);
intold=H/2*(y(1)+y(2));
%
err=toll+1;
while err > toll
    n=2*n;
    % CALCOLO DI INTAPP
    H=(b-a)/n;
    x=[a:H:b];
    y=feval(fun,x);
    intapp=H/2*y(1)+H*sum(y(2:n))+H/2*y(n+1);
    %
    err=abs(intapp-intold)/3;
    intold=intapp;
end

```