



SIS Scuola Interdipartimentale
delle Scienze, dell'Ingegneria
e della Salute



Laurea Magistrale in STN

Applicazioni di Calcolo Scientifico e Laboratorio di ACS (12 cfu)

prof. Mariarosaria Rizzardi

Centro Direzionale di Napoli – Isola C4

stanza: n. 423 – Lato Nord, 4° piano

tel.: 081 547 6545

email: mariarosaria.rizzardi@uniparthenope.it



1920 - 2020
UNIVERSITÀ FEDERICO II
DEGLI STUDI
NAPOLI
100° ANNIVERSARIO

Argomenti trattati

➤ **Calcolo Numerico in MATLAB:**

❖ **Metodi di derivazione numerica.**

Derivazione numerica (differenziazione discreta)

Il problema della **derivazione numerica** consiste nell'approssimare la derivata di una funzione f in un certo punto x_0 .

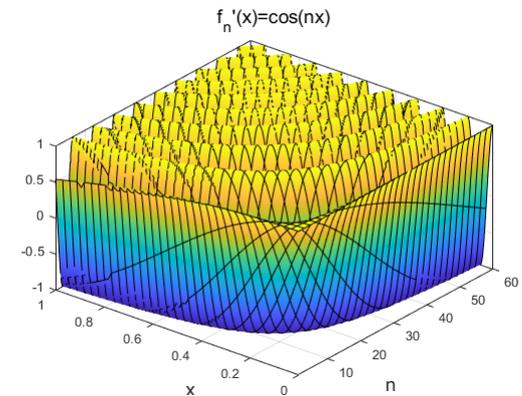
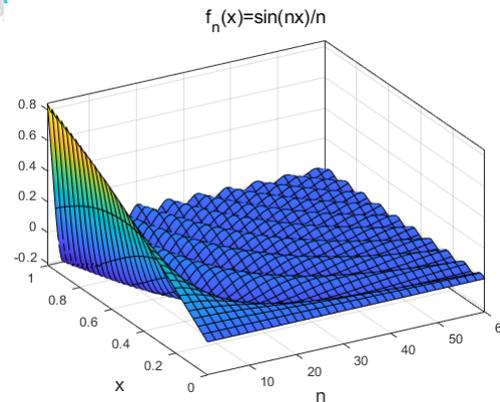
Si potrebbe approssimare f con una successione di funzioni f_n facili da derivare e poi differenziare f_n , valutandola nel punto x_0 .

Purtroppo anche se la successione $\{f_n\}$ converge ad f , ciò non garantisce che la successione delle derivate $\{f_n'\}$ converga a f' .

Esempio

La successione $\{\sin(nx)/n\}$ converge alla funzione identicamente nulla, ma la successione delle sue derivate $\{\cos(nx)\}$ non converge alla derivata della funzione costante 0

```
syms n integer positive
syms x real
assumeAlso(x>=0 & x<=1)
fn=1/n*sin(n*x); % per x in [0,1]
figure; fsurf(fn,[1 60 0 1])
limit(fn,n,Inf)
ans = 0
fn1=diff(fn,x);
figure; fsurf(fn1,[1 60 0 1])
```



Differenze finite per la derivazione numerica

Approssimare le derivate di una funzione utilizzando solo i suoi valori sui punti di una **griglia**

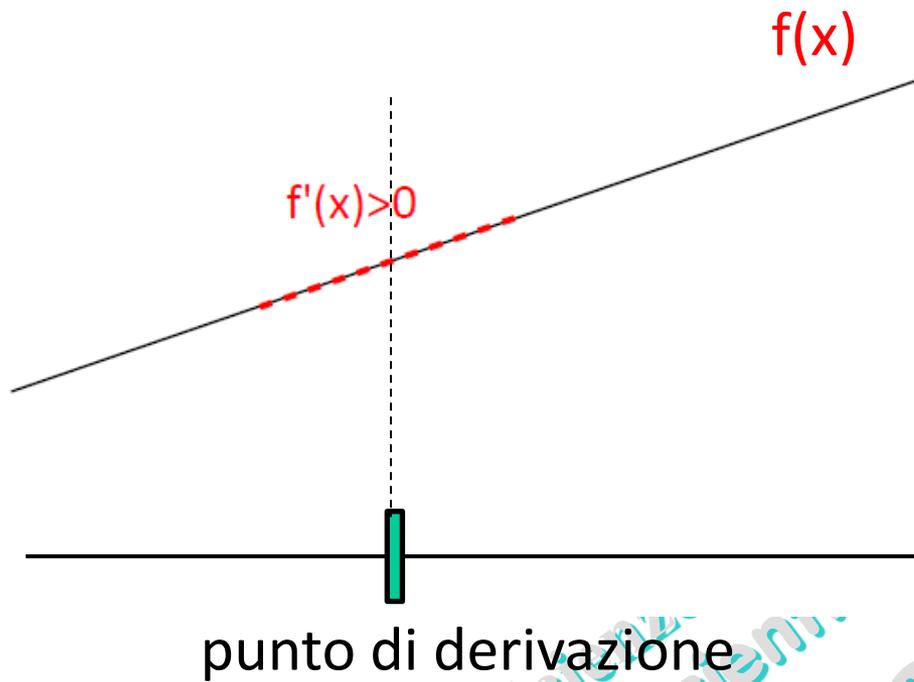
L'idea di base parte dalla **definizione di derivata prima** di una funzione $f(x)$:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

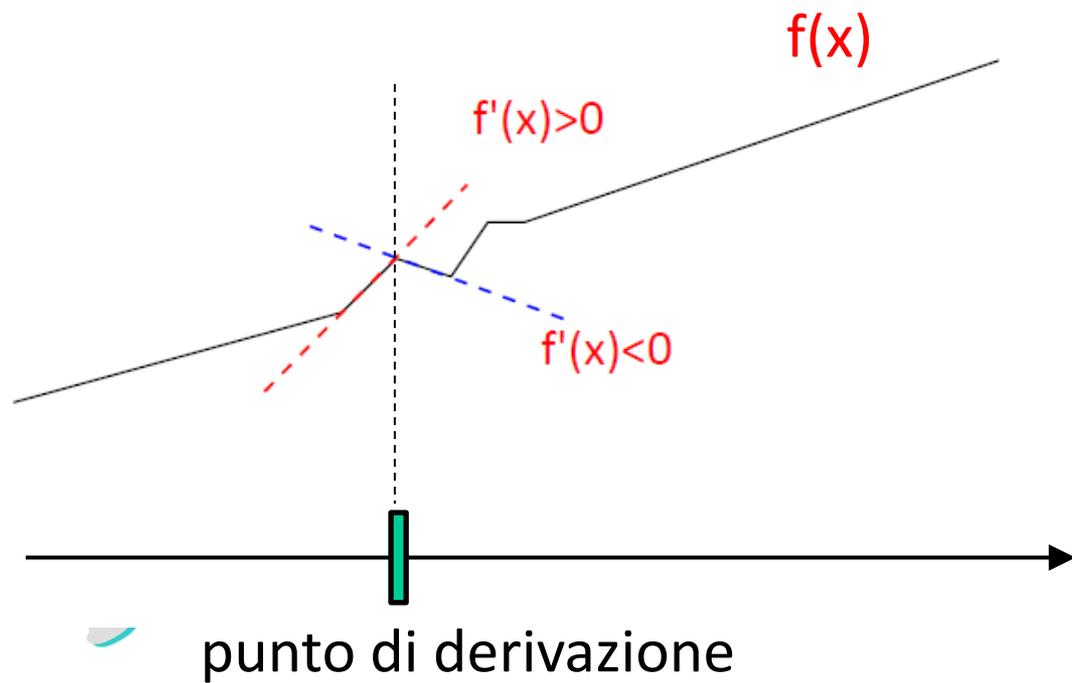
L'operazione di **derivazione** è una operazione **locale** (dipende dal comportamento della funzione nell'intorno del punto dove si deriva)



grande sensibilità a eventuali piccoli errori nei valori della funzione da derivare



funzione **vera**



funzione **con errori**

Derivazione numerica (differenziazione discreta)

Metodi di derivazione numerica

- **Differenze Finite**
 - ❖ sviluppo in serie di Taylor
 - ❖ metodo dei coefficienti indeterminati
 - ❖ interpolazione polinomiale
- **Uso della Serie di Fourier**
- **Integrali curvilinei (in C)**

tecnica delle **differenze finite**

Si approssimano le derivate con delle **differenze finite**, che sono espressioni contenenti solo i valori della funzione valutata in opportuni punti.

1) Sviluppo in Serie di Taylor

Richiami: il Resto della Formula di Taylor

Se f è dotata in un intorno $I(x_0)$, di x_0 , di derivate continue fino all'ordine n , posto

$$S_n(x) = \sum_{k=0}^{n-1} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

↑
ordine ridotta

$S_n(x)$: polinomio di Taylor
di f di grado $n-1$

si ha che

$$\forall x \in I(x_0) \exists \text{ un opportuno punto } \xi : |\xi - x_0| < |x - x_0|$$

per cui risulta:

$$|R_n(x)| = |f(x) - S_n(x)| = \frac{|f^{(n)}(\xi)|}{n!} |x - x_0|^n$$

è un **Errore Assoluto**

Sviluppo in Serie di Taylor

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \dots$$

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi) \leftarrow \text{resto}$$

$\xi \in]x, x+h[$

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$$

differenza in avanti

accuratezza del **primo** ordine

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

Esempio: **approssimare la derivata 1^a**

moltiplicatori

$$1: f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi)$$

$$-1: f(x) = f(x) \quad +$$

~~$$0: f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(\eta)$$~~

$$= f(x+h) - f(x) = hf'(x) + \frac{h^2}{2} f''(\xi)$$

→

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \left(\frac{h}{2}\right) f''(\xi)$$

Differenza in avanti: accuratezza del 1° ordine

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

Esempio: **approssimare la derivata 1^a**

moltiplicatori

~~$0: f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\xi)$~~

$1: f(x) = f(x)$

$-1: f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(\eta)$

+

$= f(x) - f(x-h) = +hf'(x) - \frac{h^2}{2} f''(\eta)$

$f'(x) = \frac{f(x) - f(x-h)}{h} + \left(\frac{h}{2}\right) f''(\eta)$

Differenza all'indietro: accuratezza del 1° ordine

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

Esempio: **approssimare la derivata 1^a**

moltiplicatori

$$1: f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(\xi)$$

~~$$0: f(x) = f(x) \quad +$$~~

$$-1: f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(\eta)$$

$$= f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{6}[f'''(\xi) + f'''(\eta)]$$

valor medio $f'''(\mu) = \frac{f'''(\xi) + f'''(\eta)}{2}, \quad \mu \in]x-h, x+h[$



$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \left(\frac{h^2}{6}\right) f'''(\mu)$$

Differenza centrale: accuratezza del 2° ordine

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

Esempio: **approssimare la derivata 2^a**

1: $f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{iv}(\xi)$

-2: $f(x) = f(x)$

1: $f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{iv}(\eta)$

+

$$= f(x+h) - 2f(x) + f(x-h) = h^2 f''(x) + \frac{h^4}{24} [f^{iv}(\xi) + f^{iv}(\eta)]$$

$$f^{iv}(\mu) = \frac{f^{iv}(\xi) + f^{iv}(\eta)}{2}, \quad \mu \in]x-h, x+h[$$


$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \frac{h^2}{12} f^{iv}(\mu)$$

Differenza centrale: accuratezza del 2° ordine

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza: **forma matriciale**

Esempio: **approssimare le derivate fino alla 4^a**

Si vogliono approssimare le derivate $f^{[k]}_i = f^{[k]}(x_i)$ di f nel punto x_i , fino al 4° ordine ($k=4$), usando i valori della funzione f su 5 punti simmetrici, centrati in x_i :

$$f_{i-2} = f(x_i - 2h), \quad f_{i-1} = f(x_i - h), \quad f_i = f(x_i), \quad f_{i+1} = f(x_i + h), \quad f_{i+2} = f(x_i + 2h)$$

Si parte sempre dagli sviluppi in Serie di Taylor, scrivendoli poi in notazione matriciale:

$$f_{i-2} = f_i - 2hf'_i + 2h^2 f''_i - \frac{4}{3}h^3 f'''_i + \frac{2}{3}h^4 f^{iv}_i + O(h^5)$$

$$f_{i-1} = f_i - hf'_i + \frac{1}{2}h^2 f''_i - \frac{1}{6}h^3 f'''_i + \frac{1}{24}h^4 f^{iv}_i + O(h^5)$$

$$f_i = f_i$$

$$f_{i+1} = f_i + hf'_i + \frac{1}{2}h^2 f''_i + \frac{1}{6}h^3 f'''_i + \frac{1}{24}h^4 f^{iv}_i + O(h^5)$$

$$f_{i+2} = f_i + 2hf'_i + 2h^2 f''_i + \frac{4}{3}h^3 f'''_i + \frac{2}{3}h^4 f^{iv}_i + O(h^5)$$

$A =$

$$\begin{pmatrix} 1 & -2 & 2 & -\frac{4}{3} & \frac{2}{3} \\ 1 & -1 & \frac{1}{2} & -\frac{1}{6} & \frac{1}{24} \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & \frac{1}{2} & \frac{1}{6} & \frac{1}{24} \\ 1 & 2 & 2 & \frac{4}{3} & \frac{2}{3} \end{pmatrix} \begin{pmatrix} f_i \\ hf'_i \\ h^2 f''_i \\ h^3 f'''_i \\ h^4 f^{iv}_i \end{pmatrix} + O(h^5) = \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix}$$

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza: **forma matriciale**

Esempio: **approssimare le derivate fino alla 4^a**

$$\begin{pmatrix} 1 & -2 & 2 & -4/3 & 2/3 \\ 1 & -1 & 1/2 & -1/6 & 1/24 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1/2 & 1/6 & 1/24 \\ 1 & 2 & 2 & 4/3 & 2/3 \end{pmatrix} \begin{pmatrix} f_i \\ hf_i' \\ h^2 f_i'' \\ h^3 f_i''' \\ h^4 f_i^{iv} \end{pmatrix} + O(h^5) = \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix} \iff \begin{pmatrix} f_i \\ hf_i' \\ h^2 f_i'' \\ h^3 f_i''' \\ h^4 f_i^{iv} \end{pmatrix} = A^{-1} \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix} - O(h^5)$$

$$A = \begin{pmatrix} 1 & -2 & 2 & -4/3 & 2/3 \\ 1 & -1 & 1/2 & -1/6 & 1/24 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1/2 & 1/6 & 1/24 \\ 1 & 2 & 2 & 4/3 & 2/3 \end{pmatrix}$$

$$f_i = \frac{1}{h} [A^{-1}]_{2,:} \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix} = \frac{1}{h} \left(\frac{1}{12} f_{i-2} - \frac{2}{3} f_{i-1} + 0 f_i + \frac{2}{3} f_{i+1} - \frac{1}{12} f_{i+2} \right) - O(h^4)$$

$$f_i'' = \frac{1}{h^2} [A^{-1}]_{3,:} \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix} = \frac{1}{h^2} \left(-\frac{1}{12} f_{i-2} + \frac{4}{3} f_{i-1} - \frac{5}{2} f_i + \frac{4}{3} f_{i+1} - \frac{1}{12} f_{i+2} \right) - O(h^3)$$

$$f_i''' = \frac{1}{h^3} [A^{-1}]_{4,:} \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix} = \frac{1}{h^3} \left(-\frac{1}{2} f_{i-2} + f_{i-1} + 0 f_i - 1 f_{i+1} + \frac{1}{2} f_{i+2} \right) - O(h^2)$$

$$f_i^{iv} = \frac{1}{h^4} [A^{-1}]_{5,:} \begin{pmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{pmatrix} = \frac{1}{h^4} (f_{i-2} - 4 f_{i-1} + 6 f_i - 4 f_{i+1} + f_{i+2}) - O(h)$$

```

A=[1 -2 2 -4/3 2/3;
    1 -1 1/2 -1/6 1/24;
    1 0 0 0 0;
    1 1 1/2 1/6 1/24;
    1 2 2 4/3 2/3];
invA=inv(sym(A))
invA =
[ 0, 0, 1, 0, 0]
[ 1/12, -2/3, 0, 2/3, -1/12]
[-1/12, 4/3, -5/2, 4/3, -1/12]
[-1/2, 1, 0, -1, 1/2]
[ 1, -4, 6, -4, 1]
    
```

2) Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

Esempio: metodo dei coefficienti indeterminati

Si supponga di voler approssimare la derivata seconda di f mediante i suoi valori su **3 nodi** centrati in x_i (simmetrici):

$$f_{i-1}=f(x_i-h), f_i=f(x_i), f_{i+1}=f(x_i+h)$$

cioè con una formula del tipo:

$$f''(x_i) \approx Af(x_i+h) + Bf(x_i) + Cf(x_i-h)$$

Si parte sempre dagli sviluppi in Serie di Taylor (supposto $h>0$):

$$f_{i-1} = f_i - hf'_i + \frac{h^2}{2} f''_i - \frac{h^3}{6} f'''_i + \frac{h^4}{24} f^{iv}(\xi_1)$$

$$f_i = f_i$$

$$f_{i+1} = f_i + hf'_i + \frac{h^2}{2} f''_i + \frac{h^3}{6} f'''_i + \frac{h^4}{24} f^{iv}(\xi_2)$$



$$f''(x_i) = A \left[f_i + hf'_i + \frac{h^2}{2} f''_i + \frac{h^3}{6} f'''_i + \frac{h^4}{24} f^{iv}(\xi_2) \right] + Bf_i + C \left[f_i - hf'_i + \frac{h^2}{2} f''_i - \frac{h^3}{6} f'''_i + \frac{h^4}{24} f^{iv}(\xi_1) \right]$$

$$f''(x_i) = f_i [A + B + C] + hf'_i [A - C] + \frac{h^2}{2} f''_i [A + C] + \frac{h^3}{6} f'''_i [A - C] + \frac{h^4}{24} [Cf^{iv}(\xi_1) + Af^{iv}(\xi_2)]$$

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

Esempio: metodo dei coefficienti indeterminati

$$f''(x_i) = f_i[A+B+C] + hf'_i[A-C] + \frac{h^2}{2}f''_i[A+C] + \frac{h^3}{6}f'''_i[A-C] + \frac{h^4}{24}[Cf^{iv}(\xi_1) + Af^{iv}(\xi_2)]$$

Eguagliando i coefficienti delle derivate tra 1° e 2° membro, si ha:

$$\begin{cases} A+B+C=0 \\ A-C=0 \\ \frac{h^2}{2}(A+C)=1 \end{cases} \iff \begin{cases} A+B+C=0 \\ A-C=0 \\ A+C=\frac{2}{h^2} \end{cases} \iff \begin{cases} B=-\frac{2}{h^2} \\ A=C=\frac{1}{h^2} \end{cases}$$

Sostituendo tali valori in $f''(x_i) \approx Af(x_i+h) + Bf(x_i) + Cf(x_i-h)$, si ha:

$$f''(x_i) = \frac{f(x_i+h) - 2f(x_i) + f(x_i-h)}{h^2} + \frac{h^2}{24}[f^{iv}(\xi_1) + f^{iv}(\xi_2)]$$

$$f^{iv}(\mu) = \frac{f^{iv}(\xi_1) + f^{iv}(\xi_2)}{2}, \quad \mu \in]x_i-h, x_i+h[\implies$$

$$f''(x_i) = \frac{f(x_i+h) - 2f(x_i) + f(x_i-h)}{h^2} + \frac{h^2}{12}f^{iv}(\mu)$$

Si ritrova la formula: **Differenza centrale: accuratezza del 2° ordine**

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

Esempio: **metodo dei coefficienti indeterminati**

Il metodo dei coefficienti indeterminati si basa su:

- Ipotesi che la derivata possa esprimersi come combinazione lineare dei valori della funzione in certi punti;
- Esprimere gli sviluppi di Taylor nei punti di approssimazione;
- Eguagliare i coefficienti, di ambo i membri, della funzione e delle sue derivate.

questione aperta

Quanti termini vanno usati per gli sviluppi di Taylor?

3) Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

interpolazione polinomiale

L'**idea** è semplice:

1. Si costruisce un opportuno polinomio interpolante su certi nodi $x_0, x_1, x_2, \dots, x_n$.
2. L'approssimazione della derivata in un nodo è ottenuta dalla derivata del polinomio interpolante.

Per il polinomio interpolante conviene usare la **forma di Newton** (mediante **differenze divise**) che si presta alla costruzione incrementale.

Differenze divise

Polinomio interpolante in forma di Newton

$$\begin{aligned}
 P_n(x) &= f[x_0] + && \text{pol. di grado 0} \\
 &+ f[x_0, x_1](x - x_0) + && \text{pol. di 1° grado} \\
 &+ f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots && \text{di 2° grado} \\
 &+ f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) + \dots && \text{di 3° grado} \\
 &\dots \\
 &+ f[x_0, x_1, \dots, x_n](x - x_0) \cdots (x - x_{n-1}) && \dots \text{di grado } n
 \end{aligned}$$

$$\begin{aligned}
 f[x_i] &= y_i, \quad i = 0, 1, \dots, n && \text{di ordine 0} \\
 f[x_i, x_{i+1}] &= \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}, \quad i = 0, \dots, n-1 && \text{di ordine 1} \\
 f[x_i, x_{i+1}, x_{i+2}] &= \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i}, \quad i = 0, \dots, n-2 && \text{di ordine 2} \\
 &\vdots && \\
 f[x_0, x_1, \dots, x_n] &= \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0} && \text{di ordine } n
 \end{aligned}$$

Schema computazionale per le differenze divise

Esempio: polinomio interpolante di Newton di grado 4 su 5 nodi

coefficienti del polinomio di Newton

dati

Differenze divise

ordine 0	1° ordine	2° ordine	3° ordine	4° ordine
$f[x_0] = y_0$				
$f[x_1] = y_1$	$f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}$			
$f[x_2] = y_2$	$f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1}$	$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$		
$f[x_3] = y_3$	$f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2}$	$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$	$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0}$	
$f[x_4] = y_4$	$f[x_3, x_4] = \frac{f[x_4] - f[x_3]}{x_4 - x_3}$	$f[x_2, x_3, x_4] = \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2}$	$f[x_1, x_2, x_3, x_4] = \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1}$	$f[x_0, x_1, x_2, x_3, x_4] = \frac{f[x_1, x_2, x_3, x_4] - f[x_0, x_1, x_2, x_3]}{x_4 - x_0}$

polinomio interpolante di Newton su 5 nodi

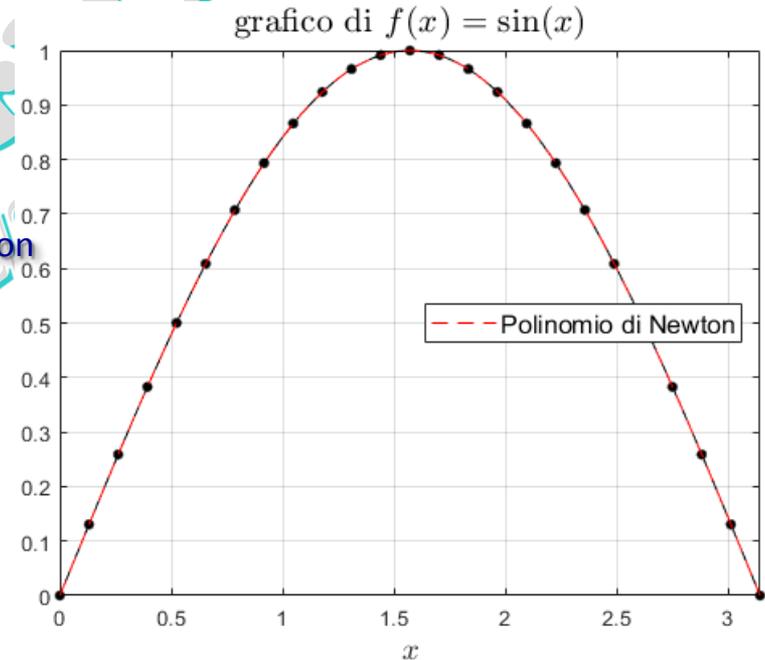
$$\begin{aligned}
 P_4(x) = & f[x_0] + \\
 & + f[x_0, x_1](x - x_0) + \\
 & + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \\
 & + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) + \\
 & + f[x_0, x_1, x_2, x_3, x_4](x - x_0)(x - x_1)(x - x_2)(x - x_3)
 \end{aligned}$$

Esempio: polinomio interpolante di Newton di grado n

$$\begin{aligned} P_n(x) = & f[x_0] \times [1] + \\ & + f[x_0, x_1] \times [1](x - x_0) + \\ & + f[x_0, x_1, x_2] \times [(1)(x - x_0)](x - x_1) + \\ & + f[x_0, x_1, x_2, x_3] \times [(1)(x - x_0)(x - x_1)](x - x_2) + \\ & \dots \\ & + f[x_0, x_1, \dots, x_n] \times [(1)(x - x_0) \cdots (x - x_{n-2})](x - x_{n-1}) \end{aligned}$$

```
f=@sin; % funzione da interpolare versione seriale
x=linspace(0,pi,100); y=f(x);
% nodi di interpolazione e campioni della funzione
n=24; xi=linspace(0,pi,n+1)'; yi=f(xi);
dd=divided_differences(xi,yi);
dd=diag(dd); % coefficienti del polinomio di Newton
newton=dd(1); prod=1;
for k=2:numel(dd)
    prod=prod.*(x-xi(k-1));
    newton=newton + dd(k)*prod;
end
```

```
function DD=divided_differences(X,Y) versione seriale
N=numel(X);
DD=zeros(N,N);
DD(:,1)=Y;
for j=2:N % NON VETTORIALIZZABILE!
    for i=j:N % VETTORIALIZZABILE
        DD(i,j)=(DD(i,j-1)-DD(i-1,j-1))/(X(i)-X(i-j+1));
    end
end
end
```



Esempio: polinomio interpolante di Newton di grado n

Algoritmo misto simbolico-numerico vettoriale

indici rinumerati

$$\begin{aligned}
 P_n(x) = & f[x_1] \times (1) + \\
 & + f[x_1, x_2] \times (x - x_1) + \\
 & + f[x_1, x_2, x_3] \times (x - x_1)(x - x_2) + \\
 & + f[x_1, x_2, x_3, x_4] \times (x - x_1)(x - x_2)(x - x_3) + \\
 & \dots \\
 & + f[x_1, x_2, \dots, x_{n+1}] \times (x - x_1) \dots (x - x_n)
 \end{aligned}$$

X_i

$\text{cumprod}(X_i)$

(1)
 $(x - x_1)$
 $(x - x_2)$
 $(x - x_3)$
 \dots
 $(x - x_n)$



(1)
 $(x - x_1)$
 $(x - x_1)(x - x_2)$
 $(x - x_1)\dots(x - x_3)$
 \dots
 $(x - x_1)\dots(x - x_n)$

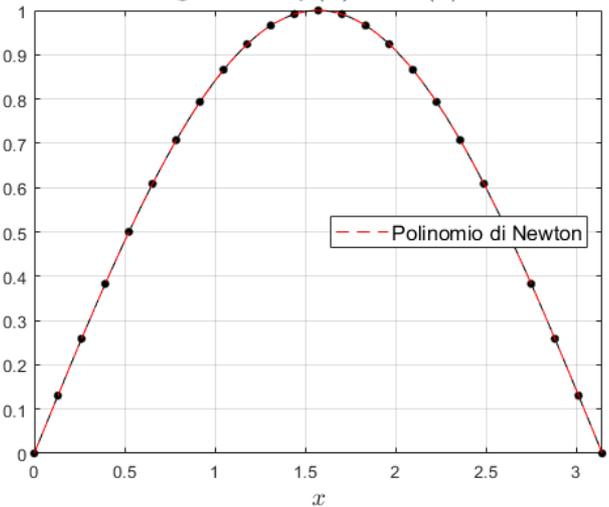
```

f=@sin; % funzione da interpolare
% nodi di interpolazione e campioni della funzione
n=24; xi=linspace(0,pi,n+1)'; yi=f(xi);
dd=divided_differences_vec(xi,yi);
dd=diag(dd); % coefficienti del polinomio interpolante
syms x real
Xi=x - xi(1:end-1); Xi=[1;Xi];
Xi=cumprod(Xi); P=dot(dd,Xi);
hP=matlabFunction(P); % Anonymous Function da espressione simbolica
x=linspace(0,pi,100); y=f(x);
plot(x,y,'k',xi,yi,'.k', x,hP(x),'r--','MarkerSize',15)
    
```

versione vettoriale

polinomio interpolante di Newton

grafico di $f(x) = \sin(x)$

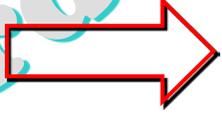


```

function DD=divided_differences_vec(X,Y)
N=numel(X);
DD=zeros(N,N);
DD(:,1)=Y;
for j=2:N
    DD(j:N,j)=(DD(j:N,j-1)-DD((j:N)-1,j-1))./(X(j:N)-X(1:(N-j+1)));
end
end
    
```

versione vettoriale

Derivate del polinomio interpolante di Newton di grado 4 su 5 nodi

$$\begin{aligned} P_4(x) = & f[x_0] + \\ & + f[x_0, x_1](x - x_0) + \\ & + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \\ & + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) + \\ & + f[x_0, x_1, x_2, x_3, x_4](x - x_0)(x - x_1)(x - x_2)(x - x_3) \end{aligned}$$


$$\begin{aligned} P_4'(x) = & f[x_0, x_1] + \\ & + f[x_0, x_1, x_2][(x - x_0) + (x - x_1)] + \\ & + f[x_0, x_1, x_2, x_3][(x - x_0)(x - x_1) + (x - x_0)(x - x_2) + (x - x_1)(x - x_2)] + \\ & + f[x_0, x_1, x_2, x_3, x_4][(x - x_0)(x - x_1)(x - x_2) + (x - x_0)(x - x_1)(x - x_3) + \\ & \quad + (x - x_0)(x - x_2)(x - x_3) + (x - x_1)(x - x_2)(x - x_3)] \end{aligned}$$

$$\begin{aligned} P_4''(x) = & 2f[x_0, x_1, x_2] + \\ & + 2f[x_0, x_1, x_2, x_3][(x - x_0) + (x - x_1) + (x - x_2)] + \\ & + 2f[x_0, x_1, x_2, x_3, x_4][(x - x_0)(x - x_1) + (x - x_0)(x - x_2) + (x - x_0)(x - x_3) + \\ & \quad + (x - x_1)(x - x_2) + (x - x_1)(x - x_3) + (x - x_2)(x - x_3)] \end{aligned}$$

Le derivate del polinomio possono essere calcolate simbolicamente ...

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

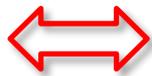
interpolazione polinomiale

Esempio

Per $i=1, 2, \dots, n$ si costruisce l'unico polinomio $P_i(x)$, di grado ≤ 1 , interpolante $f(x)$ sui nodi (x_i, y_i) e (x_{i+1}, y_{i+1}) , cioè $P_i(x_i)=y_i$ e $P_i(x_{i+1})=y_{i+1}$ e si pone: $f'(x_i) \approx P_i'(x_i)$

Polinomio interpolante di Newton di grado al più 1:

$$P_i(x) = f[x_i] + f[x_i, x_{i+1}](x - x_i)$$



$$P_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i)$$



$$f'(x_i) \approx P_i'(x_i) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

si ritrova la formula delle **differenze in avanti**

Analogamente si ritrova:

➤ la formula delle **differenze all'indietro** se, per $i=1, 2, \dots, n$, si costruisce l'unico polinomio $P_i(x)$, di grado ≤ 1 , interpolante $f(x)$ sui nodi (x_i, y_i) e (x_{i-1}, y_{i-1}) , cioè $P_i(x_i)=y_i$ e $P_i(x_{i-1})=y_{i-1}$ e si pone: $f'(x_i) \approx P_i'(x_i)$

➤ la formula delle **differenze centrali** se, per $i=1, 2, \dots, n$, si costruisce l'unico polinomio $P_i(x)$, di grado ≤ 2 , interpolante $f(x)$ sui nodi $(x_{i-1}, y_{i-1}), (x_i, y_i)$ e (x_{i+1}, y_{i+1}) , cioè $P_i(x_{i-1})=y_{i-1}, P_i(x_i)=y_i, P_i(x_{i+1})=y_{i+1}$ e si pone: $f'(x_i) \approx P_i'(x_i)$

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

interpolazione polinomiale

Esempio

Si costruisce l'unico polinomio $P_3(x)$, di grado ≤ 3 , interpolante $f(x)$ sui nodi

$$x_0 = x - 2h, \quad x_1 = x - h, \quad x_2 = x + h, \quad x_3 = x + 2h$$

e si pone: $f'(x) = P_3'(x)$.

$$P_3(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2)$$



$$P_3'(x) = f[x_0, x_1] + f[x_0, x_1, x_2](2x - x_0 - x_1) + f[x_0, x_1, x_2, x_3][(x - x_0)(x - x_1) + (x - x_0)(x - x_2) + (x - x_1)(x - x_2)]$$

```
syms x0 x1 x2 x3 real; X=[x0;x1;x2;x3]; syms y0 y1 y2 y3 real; Y=[y0;y1;y2;y3];
DD=sym_divided_differences(X,Y); d=simplify(diag(DD)); % coefficienti di P3
syms x real; P3=d(1); prod=1;
for k=2:numel(d), prod=prod.*(x-X(k-1)); P3=P3+d(k)*prod; end
P3=simplify(P3); dP3=simplify(diff(P3,x)); % P3'(x)
syms h real; dP3x=subs(dP3,{x0,x1,x2,x3},{x-2*h,x-h,x+h,x+2*h});
dP3x=simplify(dP3x); % P3'(X)
syms f(x)
dP3xf=subs(dP3x,{y0,y1,y2,y3},{f(x-2*h),f(x-h),f(x+h),f(x+2*h)});
dP3xf=simplify(dP3xf)
dP3xf =
-(8*f(x - h) - f(x - 2*h) + f(2*h + x) - 8*f(h + x))/(12*h)  $P_3'(x)$ 
```

Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

interpolazione polinomiale

Esempio: errore di troncamento in $P_3'(x)$

$$P_3'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$$

Si parte sempre dagli sviluppi in Serie di Taylor:

$$f(x-2h) = f(x) - 2hf'(x) + 2h^2f''(x) - \frac{4h^3}{3}f'''(x) + \frac{2h^4}{3}f^{iv}(x) - \frac{4h^5}{15}f^v(\xi_1) \quad \xi_1 \in]x-2h, x[$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{iv}(x) - \frac{h^5}{120}f^v(\xi_2) \quad \xi_2 \in]x-h, x[$$

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{iv}(x) + \frac{h^5}{120}f^v(\xi_3) \quad \xi_3 \in]x, x+h[$$

$$f(x+2h) = f(x) + 2hf'(x) + 2h^2f''(x) + \frac{4h^3}{3}f'''(x) + \frac{2h^4}{3}f^{iv}(x) + \frac{4h^5}{15}f^v(\xi_4) \quad \xi_4 \in]x, x+2h[$$

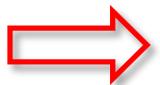
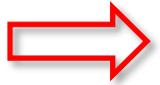
$$1f(x-2h) = \cancel{f(x)} - 2hf'(x) + \cancel{2h^2f''(x)} - \cancel{\frac{4h^3}{3}f'''(x)} + \cancel{\frac{2h^4}{3}f^{iv}(x)} - \frac{4h^5}{15}f^v(\xi_1)$$

$$-8f(x-h) = -8 \left[\cancel{f(x)} - hf'(x) + \cancel{\frac{h^2}{2}f''(x)} - \cancel{\frac{h^3}{6}f'''(x)} + \cancel{\frac{h^4}{24}f^{iv}(x)} - \frac{h^5}{120}f^v(\xi_2) \right]$$

$$8f(x+h) = 8 \left[\cancel{f(x)} + hf'(x) + \cancel{\frac{h^2}{2}f''(x)} + \cancel{\frac{h^3}{6}f'''(x)} + \cancel{\frac{h^4}{24}f^{iv}(x)} + \frac{h^5}{120}f^v(\xi_3) \right]$$

$$-1f(x+2h) = -1 \left[\cancel{f(x)} + 2hf'(x) + \cancel{2h^2f''(x)} + \cancel{\frac{4h^3}{3}f'''(x)} + \cancel{\frac{2h^4}{3}f^{iv}(x)} + \frac{4h^5}{15}f^v(\xi_4) \right]$$

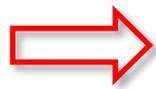
$$+ = \boxed{12hf'(x) - \frac{4h^5}{15}f^v(\xi_1) + \frac{h^5}{15}f^v(\xi_2) + \frac{h^5}{15}f^v(\xi_3) - \frac{4h^5}{15}f^v(\xi_4)}$$



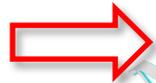
Metodologia per ricavare differenze finite ed il loro ordine di accuratezza

interpolazione polinomiale

Esempio: errore di troncamento in $P_3'(x)$ (cont.)



$$\begin{aligned}
 P_3'(x) &= \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} = \\
 &= f'(x) + \frac{-\frac{4h^5}{15} f^{(v)}(\xi_1) + \frac{h^5}{15} f^{(v)}(\xi_2) + \frac{h^5}{15} f^{(v)}(\xi_3) - \frac{4h^5}{15} f^{(v)}(\xi_4)}{12h} = \\
 &= f'(x) + h^4 \left[-\frac{1}{45} f^{(v)}(\xi_1) + \frac{1}{180} f^{(v)}(\xi_2) + \frac{1}{180} f^{(v)}(\xi_3) - \frac{1}{45} f^{(v)}(\xi_4) \right]
 \end{aligned}$$



$$P_3'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$$

$$M = \max_{x \in]x-2h, x+2h[} f^{(v)}(x)$$



$$|f'(x) - P_3'(x)| \leq h^4 M \left| \frac{-4 + 1 + 1 - 4}{180} \right|$$

Errore di troncamento

$$|f'(x) - P_3'(x)| \leq \frac{h^4}{30} M$$

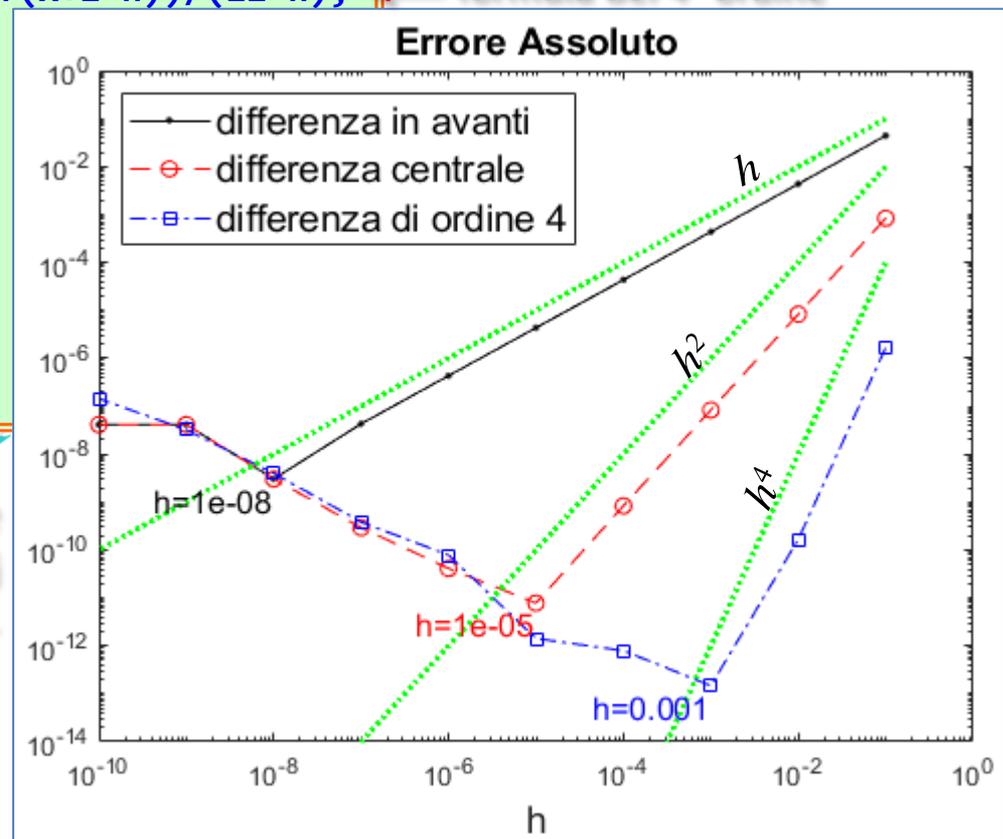
formula del 4° ordine

Come varia con h l'accuratezza di una formula?

L'errore totale può diminuire a piacere con h ?

```
%% funzione f(x)=sin(x) ==> f'(x)=cos(x) per x=pi/3 f'(x)=1/2
f=@sin; fprime=@cos; % funzione f(x) e sua derivata analitica
x=pi/3; % punto dove si vuole approssimare la derivata di f(x)
f1true=fprime(x); % valore esatto della derivata di f
%% differenze finite in test
df1=@(h) (f(x+h) - f(x))/h;
df2=@(h) (f(x+h) - f(x-h))/(2*h);
df4=@(h) (f(x-2*h)-8*f(x-h)+8*f(x+h)-f(x+2*h))/(12*h);
%% errori assoluti
AE1=zeros(10,1);
AE2=zeros(10,1);
AE4=zeros(10,1);
h=0.1.^(1:10)'; % h=10^-1, 10^-2, ..., 10^-10
for i=1:10
    AE1(i)=abs(f1true - df1(h(i)));
    AE2(i)=abs(f1true - df2(h(i)));
    AE4(i)=abs(f1true - df4(h(i)));
end
```

differenza in avanti (1° ordine)
differenza centrale (2° ordine)
formula del 4° ordine



Al diminuire di h , l'errore totale prima si comporta come l'errore di troncamento ma, una volta raggiunto il minimo, prende a crescere

Perché?

Differenze in avanti: stimare l'errore di roundoff

Errore di troncamento nella formula delle differenze in avanti

$$\left| f'(x) - \frac{f(x+h) - f(x)}{h} \right| = \frac{|f''(\xi)|}{2} h \leq \frac{M}{2} h$$
$$|f''(x)| \leq M$$

Un errore $e(x)$ in $f(x)$ comporta che, invece del valore $f(x)$, si abbia

$$\tilde{f}(x) = f(x) + e(x)$$

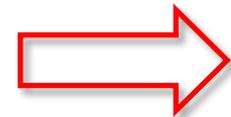
Si vuole allora stimare il seguente errore:

$$E = \left| f'(x) - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} \right|$$

$$\frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} = \frac{f(x+h) - f(x)}{h} + \frac{e(x+h) - e(x)}{h}$$

si supponga che

$$|e(x)| \leq \varepsilon \quad \forall x$$



Errore di roundoff nella formula delle differenze in avanti

$$\left| \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} - \frac{f(x+h) - f(x)}{h} \right| = \left| \frac{e(x+h) - e(x)}{h} \right| \leq \frac{2\varepsilon}{h}$$

Differenze in avanti: **stimare l'errore di roundoff**

Errore di troncamento $\left| f'(x) - \frac{f(x+h) - f(x)}{h} \right| \leq h \frac{M}{2}$

Errore di roundoff $\left| \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} - \frac{f(x+h) - f(x)}{h} \right| \leq \frac{2\varepsilon}{h}$

Errore totale nelle differenze in avanti

$$E = \left| f'(x) - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} \right| = \left| \left[f'(x) - \frac{f(x+h) - f(x)}{h} \right] + \left[\frac{f(x+h) - f(x)}{h} - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} \right] \right| \leq h \frac{M}{2} + \frac{2\varepsilon}{h}$$

decrece con h

crece al decrescere di h

$$E = \left| f'(x) - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} \right| \leq h \frac{M}{2} + \frac{2\varepsilon}{h}$$

predomina per h grande

predomina per h piccolo

Differenze in avanti: **valore ottimale di h**

$$E = \left| f'(x) - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} \right| \leq h \frac{M}{2} + \frac{2\varepsilon}{h}$$

$$h_{\text{opt}} = \arg \min_h \left\{ h \frac{M}{2} + \frac{2\varepsilon}{h} \right\}$$

$$\frac{d}{dh} \left\{ h \frac{M}{2} + \frac{2\varepsilon}{h} \right\} = 0$$

$$\frac{M}{2} - \frac{2\varepsilon}{h_{\text{opt}}^2} = 0$$

$$h_{\text{opt}}^2 = \frac{4\varepsilon}{M}$$

$$h_{\text{opt}} = 2\sqrt{\frac{\varepsilon}{M}}$$

min errore

$$E = \left| f'(x) - \frac{\tilde{f}(x+h) - \tilde{f}(x)}{h} \right| \leq \boxed{2\sqrt{\varepsilon M}}$$

Esempio

Se l'errore nel calcolo di $f(x) = \sin(x)$ è l'*epsilon macchina* ϵ_m

(in doppia precisione) $\epsilon_m \approx 2.2 \times 10^{-16}$

essendo $M=1$ perché $|f''(x)| = |\sin(x)| \leq 1$, allora il minimo errore ottenibile, in corrispondenza di $h_{\text{opt}} = 2\sqrt{\epsilon_m} \approx 3 \times 10^{-8}$, è dato da:

$$E_{\text{min}} = 2\sqrt{\epsilon_m} \approx 3 \times 10^{-8}$$

Cioè, con le **differenze in avanti**, non si può ottenere nella derivata di $\sin(x)$ un numero di cifre corrette maggiore della **metà** della precisione del sistema aritmetico

MATLAB, per default, opera in doppia precisione floating point con **64 bit** per la mantissa che corrispondono all'incirca a **16 cifre decimali**

Derivazione numerica (differenziazione discreta)



tecnica della Serie di Fourier

forma complessa della Serie di Fourier di $f(x)$ nell'intervallo $[0, 2\pi]$

$$\sum_{k=-\infty}^{+\infty} \gamma_k e^{+ikx} \quad \gamma_k = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ikx} dx$$

coefficienti di Fourier in $[0, 2\pi]$

Sotto opportune ipotesi di regolarità della $f(x)$ [parte 2 del corso], si scrive:

$$f(x) = \sum_{k=-\infty}^{+\infty} \gamma_k e^{+ikx}$$

e se in più la serie converge uniformemente (per cui è valida la derivazione termine a termine), si ha:

$$f'(x) = \sum_{k=-\infty}^{+\infty} [(ik) \gamma_k] e^{+ikx}, \quad f''(x) = \sum_{k=-\infty}^{+\infty} [(ik)^2 \gamma_k] e^{+ikx} \quad \dots \quad f^{[n]}(x) = \sum_{k=-\infty}^{+\infty} [(ik)^n \gamma_k] e^{+ikx}$$

Cioè, se si conoscono i **coefficienti di Fourier** $\{\gamma_k\}$ di $f(x)$ in $[0, 2\pi]$, i coefficienti di Fourier della derivata n -sima di f , nello stesso intervallo, sono dati da (... e non vanno ricalcolati):

$$\gamma_k^{[n]} = (ik)^n \gamma_k \quad i \text{ è l'unità immaginaria}$$

Derivazione numerica mediante Serie di Fourier

Più in generale, si può proiettare un qualsiasi intervallo $[a, b]$ in $[0, 2\pi]$:

$$t \in [a, b] \longleftrightarrow x = \frac{2\pi}{b-a}(t-a) \in [0, 2\pi]$$

Forma complessa della Serie di Fourier di $f(t)$ nell'intervallo $[a, b]$

$$f(t) = \sum_{k=-\infty}^{+\infty} \gamma_k e^{+ik \frac{2\pi}{b-a}(t-a)} \quad \gamma_k = \frac{1}{b-a} \int_a^b f(t) e^{-ik \frac{2\pi}{b-a}(t-a)} dt$$

La precedente formula di approssimazione delle derivate di $f(t)$, per l'intervallo $[a, b]$, si scrive:

Se si conoscono i **coefficienti di Fourier** $\{\gamma_k\}$ di $f(t)$ in $[a, b]$, allora i coefficienti di Fourier della derivata n -sima di f , nello stesso intervallo, sono dati da:

$$\gamma_k^{[n]} = \left(\frac{2\pi}{b-a} ik \right)^n \gamma_k \quad i \text{ è l'unità immaginaria}$$

Derivazione numerica mediante Serie di Fourier

$$f(t) \approx \underbrace{\sum_{k=-\frac{N}{2}}^{+\frac{N}{2}} \gamma_k e^{+ik\frac{2\pi}{b-a}(t-a)}}_{\text{ridotta della Serie}} \quad \Rightarrow \quad f^{[n]}(t) \approx \sum_{k=-\frac{N}{2}}^{+\frac{N}{2}} \left[\left(\frac{2\pi}{b-a} ik \right)^n \gamma_k \right] e^{+ik\frac{2\pi}{b-a}(t-a)} \quad t \in [a, b]$$

Come valutare un polinomio trigonometrico in MATLAB

$$P_N(t) = \sum_{k=-\frac{N}{2}}^{+\frac{N}{2}} \gamma_k e^{+ik\frac{2\pi}{b-a}(t-a)} \quad \text{posto} \quad z(t) = e^{+i2\pi\frac{t-a}{b-a}} \quad \Rightarrow \quad P_N[z(t)] = \sum_{k=-\frac{N}{2}}^{+\frac{N}{2}} \gamma_k [z(t)]^k$$

$$P_N[z(t)] = \gamma_{-\frac{N}{2}} z(t)^{-\frac{N}{2}} + \gamma_{-\frac{N}{2}+1} z(t)^{-\frac{N}{2}+1} + \dots + \gamma_0 + \dots + \gamma_{+\frac{N}{2}} z(t)^{+\frac{N}{2}} =$$
$$= z(t)^{-\frac{N}{2}} \left[\gamma_{-\frac{N}{2}} + \gamma_{-\frac{N}{2}+1} z(t)^{+1} + \dots + \gamma_0 z(t)^{+\frac{N}{2}} + \dots + \gamma_{+\frac{N}{2}} z(t)^{+N} \right]$$

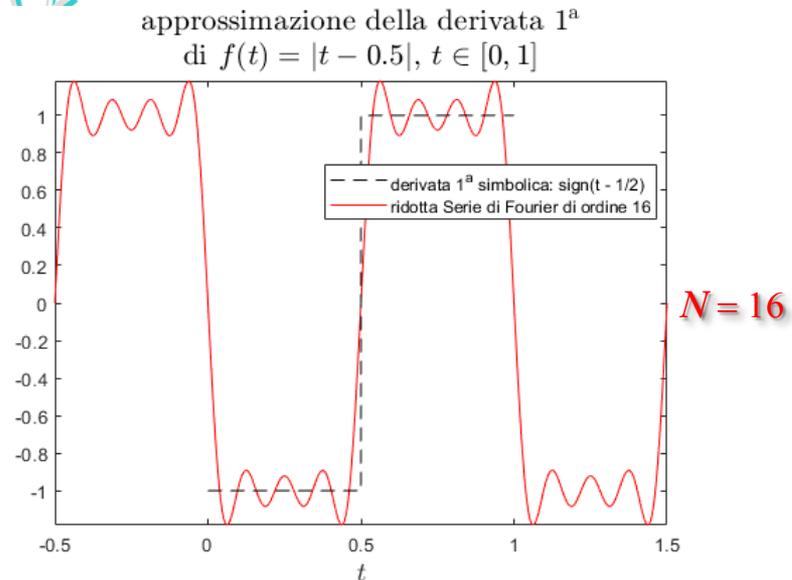
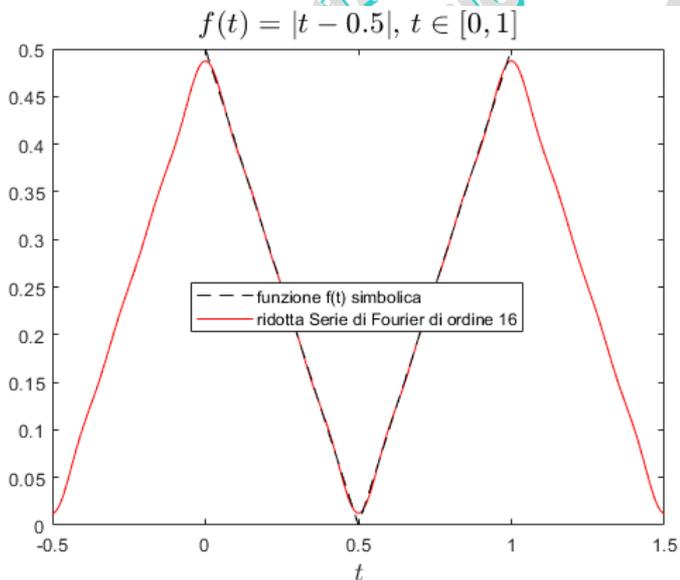
sembra un polinomio algebrico

```
a=...; b=...; N=...; % N deve essere pari
% gamma_k già calcolati come vettore colonna
t=linspace(a,b,100)'; zt=exp(2i*pi*(t-a)/(b-a));
P=zt.^(-N/2).*polyval(flipud(gamma_k),zt);
```

Derivazione numerica mediante Serie di Fourier

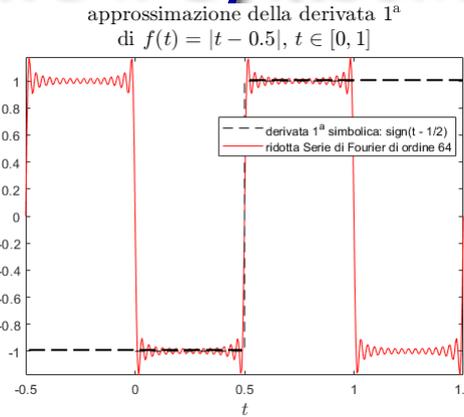
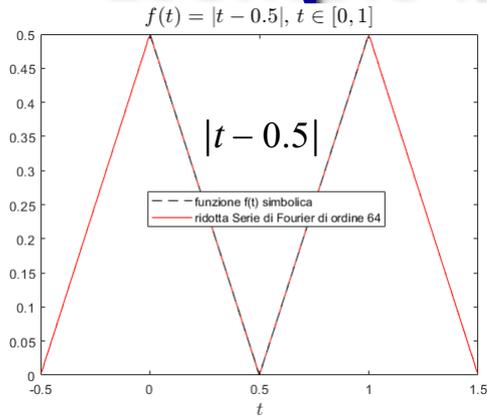
Esempio mediante il Symbolic Math Toolbox

```
a=0; b=1; N=16; f=@(x)abs(x-mean([a b]));  
IF=@(x,k)f(x)*exp(-2i*k*sym(pi)*(x-a)/(b-a)); % funzione integranda nel coefficiente di Fourier  
syms t real; syms K integer  
cK=simplify(1/(b-a)*int(IF(t,K),t,a,b)); % coeff. gamma_k  
c0=simplify(1/(b-a)*int(f(t),t,a,b)); % coeff. gamma_0  
Ck=[subs(cK,K,(-N/2:-1)');c0;subs(cK,K,(1:N/2)')]; Ck=double(Ck); % coeff. Fourier num.  
x=linspace(a-0.5,b+0.5,400)'; % valuta in x la ridotta di ordine N della Serie di Fourier  
zt=exp(2i*pi*(x-a)/(b-a)); FSx=zt.^(-N/2).*polyval(flipud(Ck),zt);  
plot(x,real(FSx),'r'); axis tight; hold on; fplot(f,[a,b],'Color','k','LineStyle','--');  
k=(-N/2:N/2)'; C1k=2i*pi/(b-a)*k.*Ck; % coefficienti della derivata prima  
FS1x=zt.^(-N/2).*polyval(flipud(C1k),zt); % valuta in x la ridotta di ordine N della Serie derivata  
f1=diff(f(t)); % derivata simbolica di f(x)  
plot(x,real(FS1x),'r'); hold on; fplot(f1,[a,b],'Color','k','LineStyle','--');
```



Derivazione numerica mediante Serie di Fourier

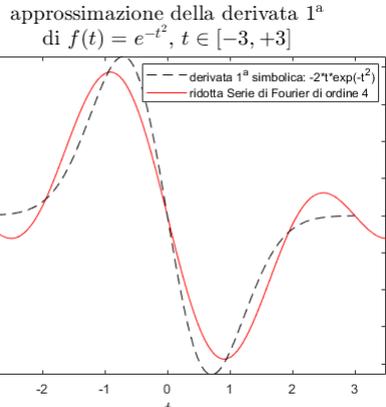
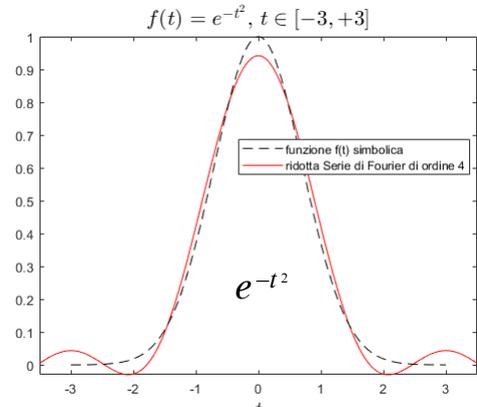
Esempio mediante il Symbolic Math Toolbox



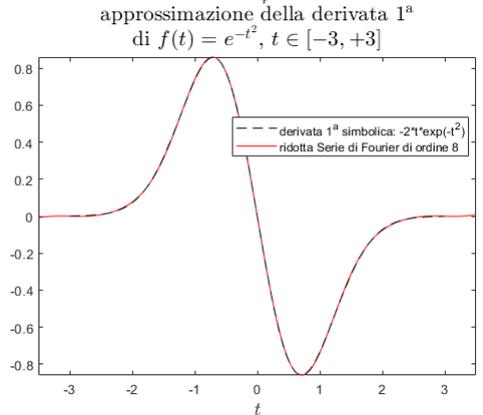
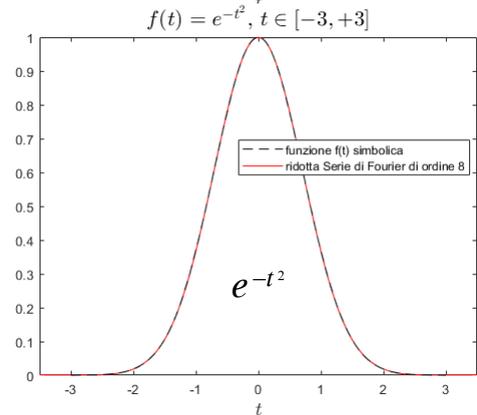
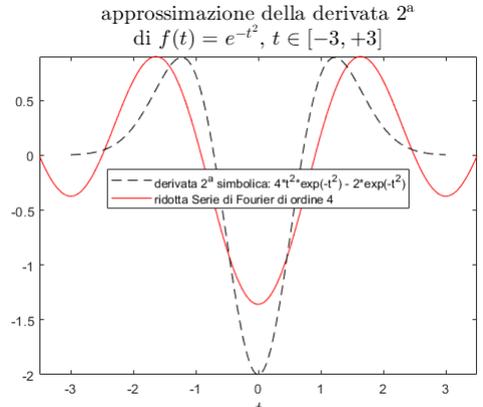
$\text{signum}(t - 0.5)$

$N = 64$

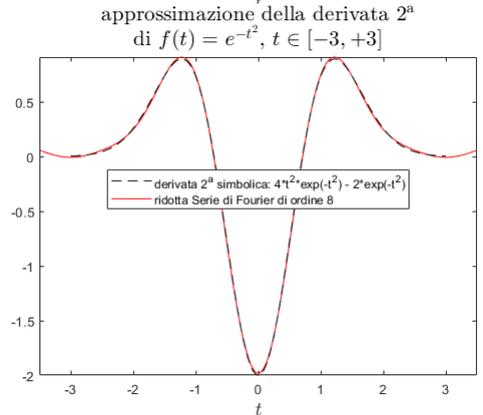
La derivata 1^a di $|t|$ è la funzione $\text{signum}(t)$



$N = 4$



$N = 8$



Derivazione numerica mediante Serie di Fourier

Esempio numerico [derivazione algoritmo nella parte 2]

```
a=0; b=1; N=16; f=@(x)abs(x-mean([a b]));  
L=b-a; Xj=a+L/N*(0:N)'; fj=f(Xj); % N+1 campioni del segnale in [a,b] finestra di ampiezza L  
ff=[mean(fj([1 end])); fj(2:end-1)]; % rende periodico il segnale  
Ck=fftshift(fft(ff)); % calcola la DFT mediante FFT  
Ck=[Ck;Ck(1)]/N; % aggiunge ultimo coeff. e costante moltiplicativa  
x=linspace(a-0.5,b+0.5,1000)'; % valuta in x la ridotta di ordine N della Serie di Fourier  
zt=exp(2i*pi*(x-a)/(b-a)); FSx=zt.^(-N/2).*polyval(flipud(Ck),zt);  
plot(x,real(FSx),'r'); axis tight; hold on; fplot(f,[a,b],'Color','k','LineStyle','--');  
k=(-N/2:N/2)'; C1k=2i*pi/L*k.*Ck; % coefficienti numerici della derivata prima  
FS1x=real(zt.^(-N/2).*polyval(flipud(C1k),zt)); % valuta in x la ridotta di ordine N della Serie derivata  
syms t real; f1=diff(f(t)); % derivata simbolica di f(x) per confronto  
plot(x,FS1x,'r'); hold on; fplot(f1,[a,b],'Color','k','LineStyle','--');
```

coefficienti di
Fourier numerici

Formula “alternativa” per la derivata 1^a numerica calcolata nei nodi X_j

```
...  
kk=[0:(N/2-1) (-N/2):-1]';  
FS1j=real(iff(2i*pi/L*kk.*fft(ff)));  
...
```

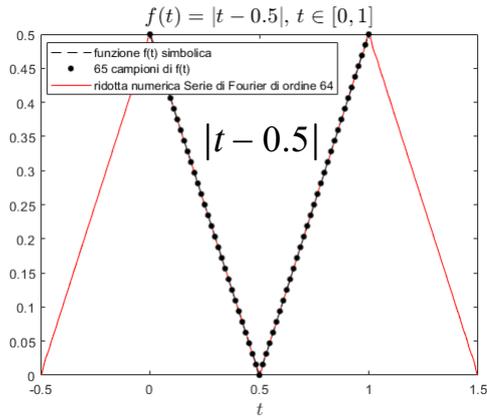
PRO: più efficiente,
più accurata

CONTRO: valida solo
per i nodi

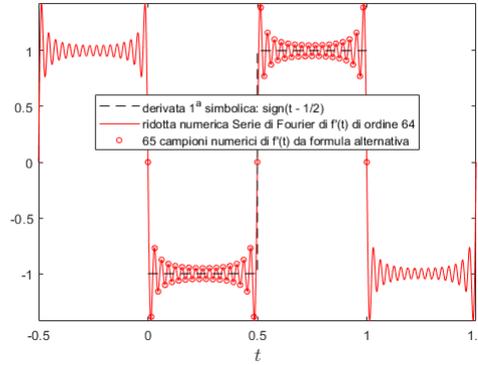
I campioni delle derivate nei nodi X_j possono poi essere interpolati mediante **interpolazione trigonometrica**, per es. con la funzione MATLAB **interpft()**, per valutare le derivate in altri punti e/o su più punti.

Derivazione numerica mediante Serie di Fourier

Esempio numerico

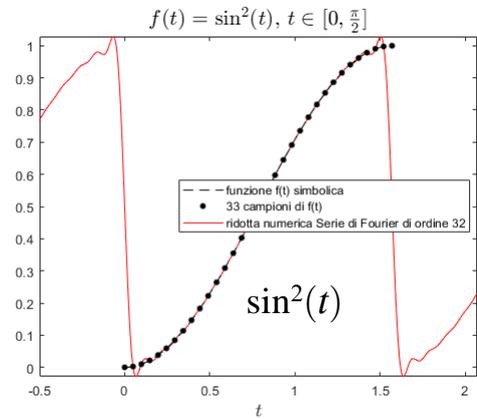


approssimazione numerica della derivata 1^a
di $f(t) = |t - 0.5|, t \in [0, 1]$

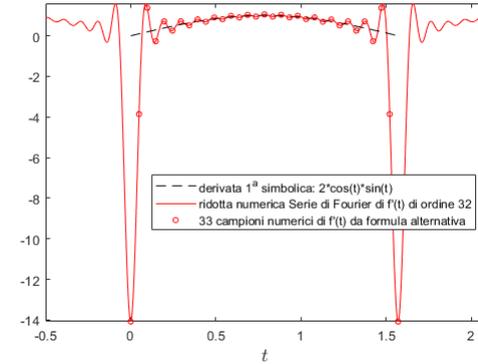


$\text{signum}(t - 0.5)$

$N = 64$

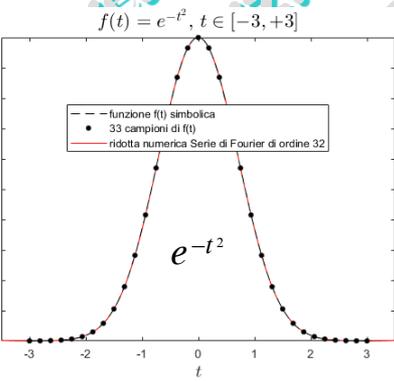
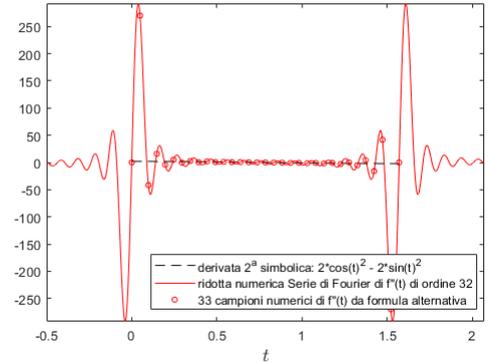


approssimazione numerica della derivata 1^a
di $f(t) = \sin^2(t), t \in [0, \frac{\pi}{2}]$

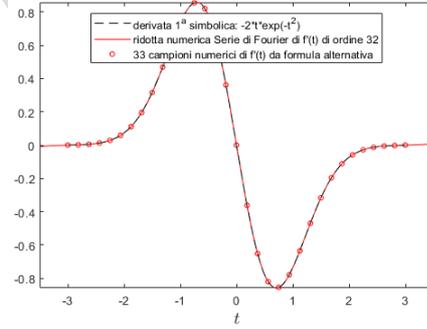


$N = 32$

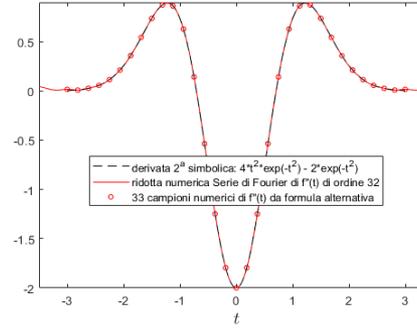
approssimazione della derivata 2^a
di $f(t) = \sin^2(t), t \in [0, \frac{\pi}{2}]$



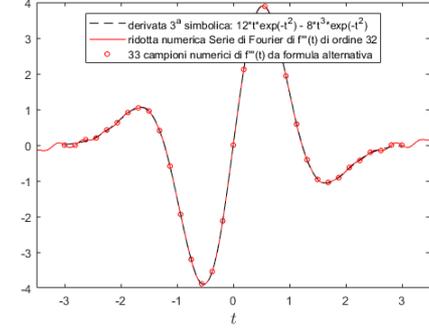
approssimazione numerica della derivata 1^a
di $f(t) = e^{-t^2}, t \in [-3, +3]$



approssimazione della derivata 2^a
di $f(t) = e^{-t^2}, t \in [-3, +3]$



approssimazione della derivata 3^a
di $f(t) = e^{-t^2}, t \in [-3, +3]$



Derivazione numerica (differenziazione discreta)



tecnica con Integrali Curvilinei

Se si considera la funzione f , di cui si vogliono approssimare le derivate, come una funzione di variabile complessa, la teoria delle *funzioni olomorfe* (cioè derivabili in senso complesso) fornisce una formula integrale per il calcolo delle derivate in un punto.

Teorema: Formula Integrale di Cauchy per le derivate

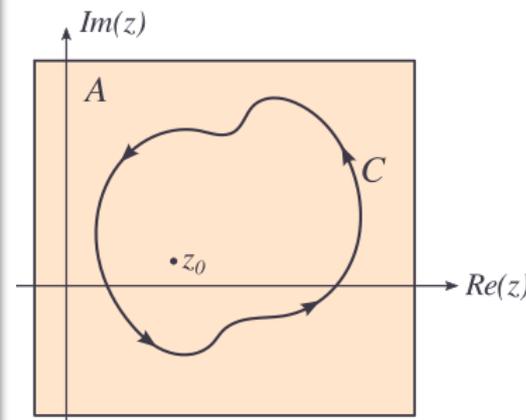
Sia $C \subset \mathbb{C}$ una qualsiasi **curva**, nel campo complesso, semplice e chiusa, orientata in verso antiorario, e sia $f(z)$, $z \in \mathbb{C}$, una *funzione analitica* olomorfa* in una regione contenente nel suo interno la curva C (*cioè ivi sviluppabile in serie di potenze), allora per ogni z_0 interno a C si ha:

$$f^{(n)}(z_0) = \frac{n!}{2\pi i} \oint_C \frac{f(z)}{(z - z_0)^{n+1}} dz, \quad n = 0, 1, 2, \dots$$

integrale curvilineo

semplice: priva di intrecci;

chiusa: punti di inizio e fine coincidono



Se C ha eq. parametrica: $z=z(t)$, $t \in [a, b]$, per calcolare l'integrale si opera la sostituzione di variabile: $z=z(t)$, $dz=z'(t)dt$. Si ottiene un integrale definito con variabile di integrazione reale:

$$f^{(n)}(z_0) = \frac{n!}{2\pi i} \int_a^b \frac{f[z(t)]}{[z(t) - z_0]^{n+1}} z'(t) dt, \quad n = 0, 1, 2, \dots$$

si dimostra che il valore dell'integrale **non dipende da C**

Esempio: circonferenza come curva d'integrazione

Si vogliono approssimare le derivate di $f(x)=\sin(x)$ in $x_0=\pi/3$.

Si passi alla variabile complessa z : derivate di $f(z)=\sin(z)$ in $z_0=\pi/3$.

Come curva C si può scegliere una circonferenza di raggio r che circonda z_0 (per es. di centro z_0). C ha eq. parametrica complessa:

$$z=z(\theta)=z_0+r \exp(i\theta), \theta \in [-\pi, +\pi] \Rightarrow dz=ir \exp(i\theta)$$

```
fun=@(z) sin(z); Z0=pi/3; % funzione da derivare e punto di derivazione
n=...; % ordine di derivazione
funInt=@(z,z0) fun(z)./(z-z0).^(n+1); % funzione integranda
zC=@(theta,z0,r) z0+r*exp(1i*theta); % z=z(theta) su C
DzC=@(theta,r) 1i*r*exp(1i*theta); % derivata di z(theta)
R=...; % raggio circonferenza C
Cfun=@(theta) funInt(zC(theta,Z0,R),Z0).*DzC(theta,R);
der_integral=factorial(n)/(2i*pi) * integral(Cfun,-pi,pi);
```

integral(): funzione numerica per approssimare vari tipi di integrali

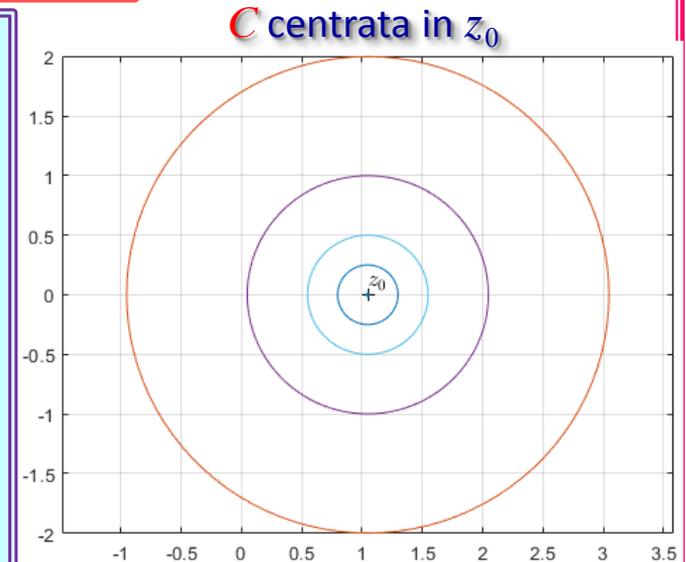
funzione integranda

```
C: circonferenza raggio r=2 e centro C0= 1.0472
Derivata numerica (integral) di ordine 1 di f(z)=sin(z)
[f^(1)(z)=cos(z)] per z0=pi/3 [cos(z0)=0.5] = 0.5 + 9.8702e-18i
```

```
C: circonferenza raggio r=1 e centro C0= 1.0472
Derivata numerica (integral) di ordine 1 di f(z)=sin(z)
[f^(1)(z)=cos(z)] per z0=pi/3 [cos(z0)=0.5] = 0.5 + 3.4787e-17i
```

```
C: circonferenza raggio r=0.5 e centro C0= 1.0472
Derivata numerica (integral) di ordine 1 di f(z)=sin(z)
[f^(1)(z)=cos(z)] per z0=pi/3 [cos(z0)=0.5] = 0.5 + 4.3622e-17i
```

```
C: circonferenza raggio r=0.25 e centro C0= 1.0472
Derivata numerica (integral) di ordine 1 di f(z)=sin(z)
[f^(1)(z)=cos(z)] per z0=pi/3 [cos(z0)=0.5] = 0.5 - 2.3192e-17i
```



Esempio: circonferenza come curva d'integrazione

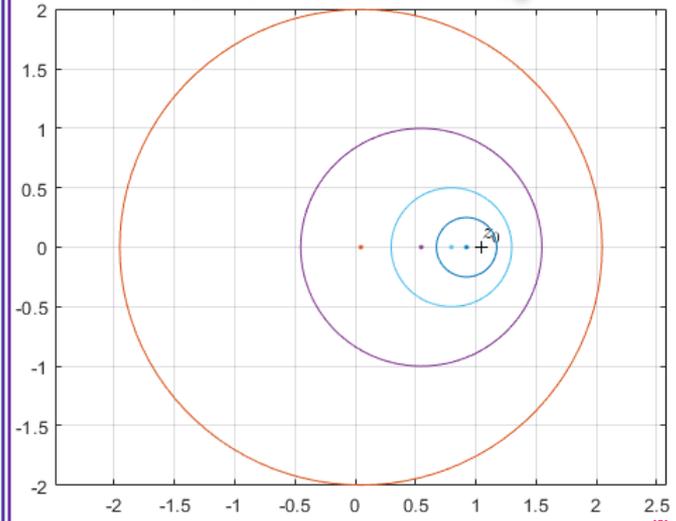
C: circonferenza raggio $r=2$ e centro $C0= 0.92206$
 Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
 $[f^{(3)}(z)=-\cos(z)]$ per $z0=\pi/3 [-\cos(z0)=-0.5] = -0.5+4.53e-17i$

C: circonferenza raggio $r=1$ e centro $C0= 0.92206$
 Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
 $[f^{(3)}(z)=-\cos(z)]$ per $z0=\pi/3 [-\cos(z0)=-0.5] = -0.5+4.75e-16i$

C: circonferenza raggio $r=0.5$ e centro $C0= 0.92206$
 Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
 $[f^{(3)}(z)=-\cos(z)]$ per $z0=\pi/3 [-\cos(z0)=-0.5] = -0.5-4.64e-15i$

C: circonferenza raggio $r=0.25$ e centro $C0= 0.92206$
 Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
 $[f^{(3)}(z)=-\cos(z)]$ per $z0=\pi/3 [-\cos(z0)=-0.5] = -0.5+5.74e-14i$

C non centrata in z_0



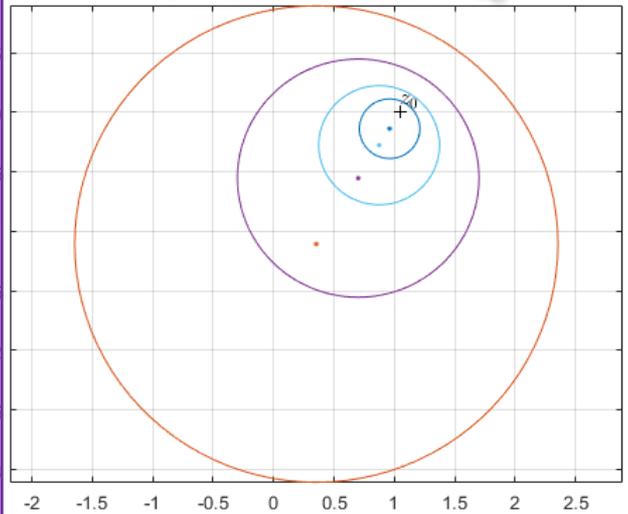
C: circonferenza raggio $r=2$ e centro $C0= 0.96051-0.1383i$
 Derivata numerica (integral) di ordine 4 di $f(z)=\sin(z)$
 $[f^{(4)}(z)=\sin(z)]$ per $z0=\pi/3 [\sin(z0)=0.86603] = 0.86603$

C: circonferenza raggio $r=1$ e centro $C0= 0.96051-0.1383i$
 Derivata numerica (integral) di ordine 4 di $f(z)=\sin(z)$
 $[f^{(4)}(z)=\sin(z)]$ per $z0=\pi/3 [\sin(z0)=0.86603] = 0.86603+1.5e-13i$

C: circonferenza raggio $r=0.5$ e centro $C0= 0.96051-0.1383i$
 Derivata numerica (integral) di ordine 4 di $f(z)=\sin(z)$
 $[f^{(4)}(z)=\sin(z)]$ per $z0=\pi/3 [\sin(z0)=0.86603] = 0.86603-1.9e-13i$

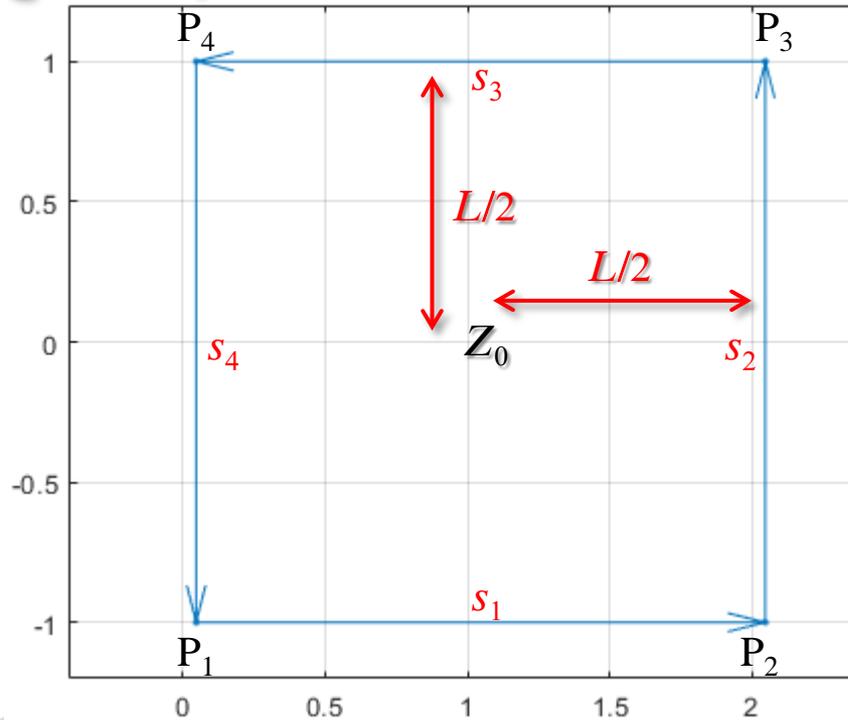
C: circonferenza raggio $r=0.25$ e centro $C0= 0.96051-0.1383i$
 Derivata numerica (integral) di ordine 4 di $f(z)=\sin(z)$
 $[f^{(4)}(z)=\sin(z)]$ per $z0=\pi/3 [\sin(z0)=0.86603] = 0.86603+3.9e-11i$

C non centrata in z_0



Esempio: quadrato come curva d'integrazione

$$P_4 = Z_0 - L/2 * (1 - i)$$



$$P_3 = Z_0 + L/2 * (1 + i)$$

$$P_1 = Z_0 - L/2 * (1 + i)$$

$$P_2 = Z_0 + L/2 * (1 - i)$$

In questo caso, si deve dividere il contorno nei 4 segmenti orientati s_k .

Equazioni parametriche dei 4 segmenti orientati

$$s_1 : z = z(\lambda) = P_1 + \lambda(P_2 - P_1), \quad \lambda \in [0, 1], \quad P_1 [Z_0 - L/2 * (1 + i)], \quad P_2 [Z_0 + L/2 * (1 - i)] = \\ = Z_0 - L/2 * (1 + i) + \lambda[L], \quad \lambda \in [0, 1]$$

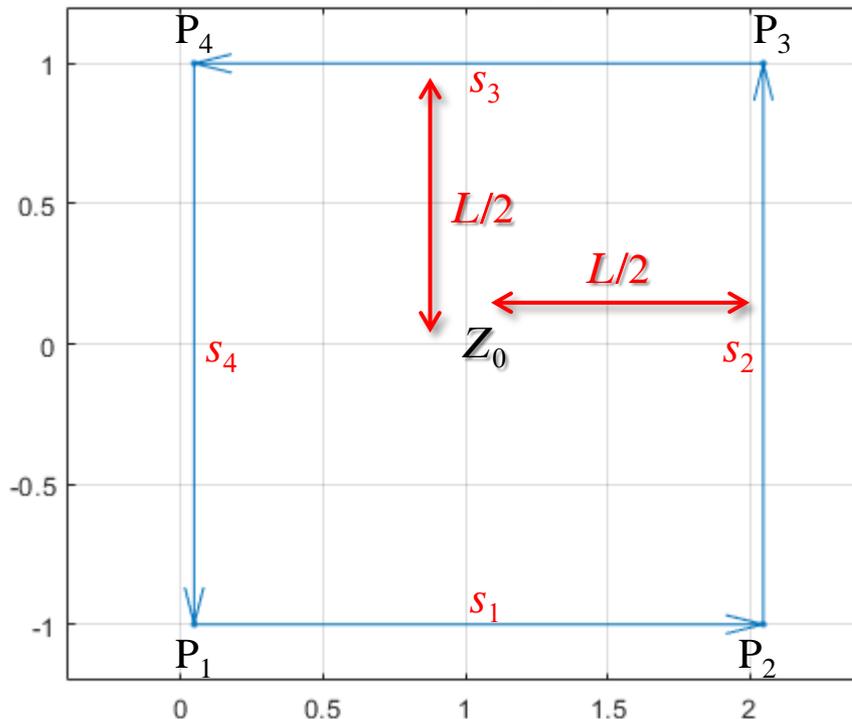
$$s_2 : z = z(\lambda) = P_2 + \lambda(P_3 - P_2), \quad \lambda \in [0, 1], \quad P_2 [Z_0 + L/2 * (1 - i)], \quad P_3 [Z_0 + L/2 * (1 + i)] = \\ = Z_0 + L/2 * (1 - i) + i * \lambda[L], \quad \lambda \in [0, 1]$$

$$s_3 : z = z(\lambda) = P_3 + \lambda(P_4 - P_3), \quad \lambda \in [0, 1], \quad P_3 [Z_0 + L/2 * (1 + i)], \quad P_4 [Z_0 - L/2 * (1 - i)] = \\ = Z_0 + L/2 * (1 + i) - \lambda[L], \quad \lambda \in [0, 1]$$

$$s_4 : z = z(\lambda) = P_4 + \lambda(P_1 - P_4), \quad \lambda \in [0, 1], \quad P_4 [Z_0 - L/2 * (1 - i)], \quad P_1 [Z_0 - L/2 * (1 + i)] = \\ = Z_0 - L/2 * (1 - i) - i * \lambda[L], \quad \lambda \in [0, 1]$$

Esempio: quadrato come curva d'integrazione

$$P_4 = Z_0 - L/2 * (1 - i)$$



$$P_3 = Z_0 + L/2 * (1 + i)$$

$$P_1 = Z_0 - L/2 * (1 + i)$$

$$P_2 = Z_0 + L/2 * (1 - i)$$

Equazioni parametriche e derivate

$$s_1 : z = z(\lambda) = Z_0 - L/2 * (1 + i) + \lambda[L], \quad \lambda \in [0, 1] \quad z' = z'(\lambda) = L$$

$$s_2 : z = z(\lambda) = Z_0 + L/2 * (1 - i) + i * \lambda[L], \quad \lambda \in [0, 1] \quad z' = z'(\lambda) = i * L$$

$$s_3 : z = z(\lambda) = Z_0 + L/2 * (1 + i) - \lambda[L], \quad \lambda \in [0, 1] \quad z' = z'(\lambda) = -L$$

$$s_4 : z = z(\lambda) = Z_0 - L/2 * (1 - i) - i * \lambda[L], \quad \lambda \in [0, 1] \quad z' = z'(\lambda) = -i * L$$

$$\Rightarrow f^{(n)}(z_0) = \frac{n!}{2\pi i} \oint_C \frac{f(z)}{(z - z_0)^{n+1}} dz = \frac{n!}{2\pi i} \left[\oint_{s_1} \frac{f(z)}{(z - z_0)^{n+1}} dz + \oint_{s_2} \frac{f(z)}{(z - z_0)^{n+1}} dz + \oint_{s_3} \frac{f(z)}{(z - z_0)^{n+1}} dz + \oint_{s_4} \frac{f(z)}{(z - z_0)^{n+1}} dz \right]$$

Esempio: quadrato come curva d'integrazione

zione
di ACS
Rizzardi

S: quadrato di centro $Z_0=1.0472$ e lato $L=2$
Derivata numerica (integral) di ordine 2 di $f(z)=\sin(z)$ [$f^{(2)}(z)=-\sin(z)$]
per $z_0=\pi/3$ [$-\sin(z_0)=-0.86603$] = $-0.86603 + 1.005e-16i$

S: quadrato di centro $Z_0=1.0472$ e lato $L=1$
Derivata numerica (integral) di ordine 2 di $f(z)=\sin(z)$ [$f^{(2)}(z)=-\sin(z)$]
per $z_0=\pi/3$ [$-\sin(z_0)=-0.86603$] = $-0.86603 + 3.589e-17i$

S: quadrato di centro $Z_0=1.0472$ e lato $L=0.5$
Derivata numerica (integral) di ordine 2 di $f(z)=\sin(z)$ [$f^{(2)}(z)=-\sin(z)$]
per $z_0=\pi/3$ [$-\sin(z_0)=-0.86603$] = $-0.86603 + 6.361e-16i$

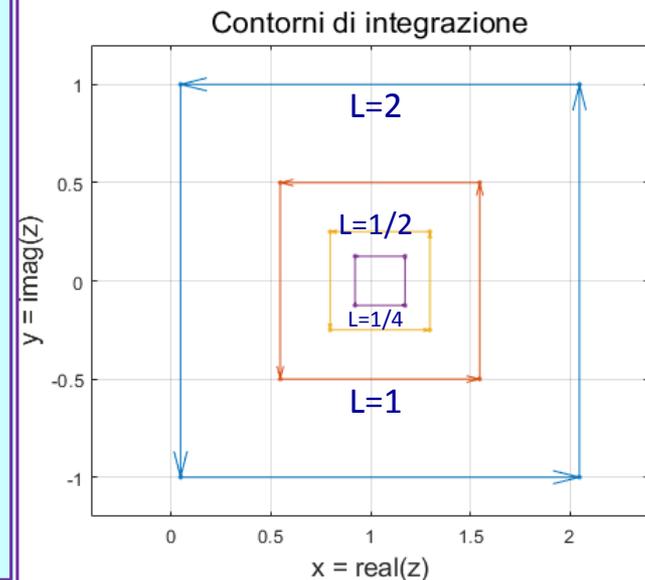
S: quadrato di centro $Z_0=1.0472$ e lato $L=0.25$
Derivata numerica (integral) di ordine 2 di $f(z)=\sin(z)$ [$f^{(2)}(z)=-\sin(z)$]
per $z_0=\pi/3$ [$-\sin(z_0)=-0.86603$] = $-0.86603 - 3.216e-15i$

S: quadrato di centro $Z_0=1.0472$ e lato $L=2$
Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
[$f^{(3)}(z)=-\cos(z)$] per $z_0=\pi/3$ [$-\cos(z_0)=-0.5$] = $-0.5+1.29e-16i$

S: quadrato di centro $Z_0=1.0472$ e lato $L=1$
Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
[$f^{(3)}(z)=-\cos(z)$] per $z_0=\pi/3$ [$-\cos(z_0)=-0.5$] = $-0.5+5.96e-17i$

S: quadrato di centro $Z_0=1.0472$ e lato $L=0.5$
Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
[$f^{(3)}(z)=-\cos(z)$] per $z_0=\pi/3$ [$-\cos(z_0)=-0.5$] = $-0.5+5.87e-15i$

S: quadrato di centro $Z_0=1.0472$ e lato $L=0.25$
Derivata numerica (integral) di ordine 3 di $f(z)=\sin(z)$
[$f^{(3)}(z)=-\cos(z)$] per $z_0=\pi/3$ [$-\cos(z_0)=-0.5$] = $-0.5-1.98e-14i$



in 5/10/50/100/500/1000/2000

La funzione numerica **diff()** di MATLAB

La funzione numerica **diff()** calcola la differenza tra elementi successivi. Se **X** è un vettore di lunghezza **n**, allora **Y=diff(X)** restituisce il vettore **Y** di lunghezza **n-1** contenente:

$$\begin{aligned} Y(1) &= X(2) - X(1), \\ Y(2) &= X(3) - X(2), \\ Y(3) &= X(4) - X(3), \dots \end{aligned}$$

Se **X** è una matrice di size **m×n**, allora **Y=diff(X)** restituisce la matrice di size **(m-1)×n**, dove la differenza tra elementi successivi è applicata a ciascuna colonna della matrice.

diff(X,n) applica **n** volte ricorsivamente l'operatore **diff**.

diff(X,n,dim) applica **n** volte l'operatore **diff** lungo la dimensione **dim**.

Esempi

```
X = [1 1 2 3 5 8 13 21];
```

```
Y = diff(X)
```

```
Y =
```

```
0     1     1     2     3     5     8
```

```
X = [1 1 2 3 5 8 13 21];
```

```
Y = diff(X,2) uguale a diff(diff(X))
```

```
Y =
```

```
1     0     1     1     2     3
```

```
X = [1 1 1; 5 5 5; 25 25 25];
```

```
Y = diff(X)
```

```
Y =
```

```
4     4     4  
20    20    20
```

```
Y = diff(X,2)
```

```
Y =
```

```
16    16    16
```

```
X = [1 1 1; 5 5 5; 25 25 25];
```

```
Y = diff(X,1,2) dimensione
```

```
Y =
```

```
0     0  
0     0  
0     0
```

La funzione numerica **diff()** di MATLAB per approssimare le derivate di una funzione

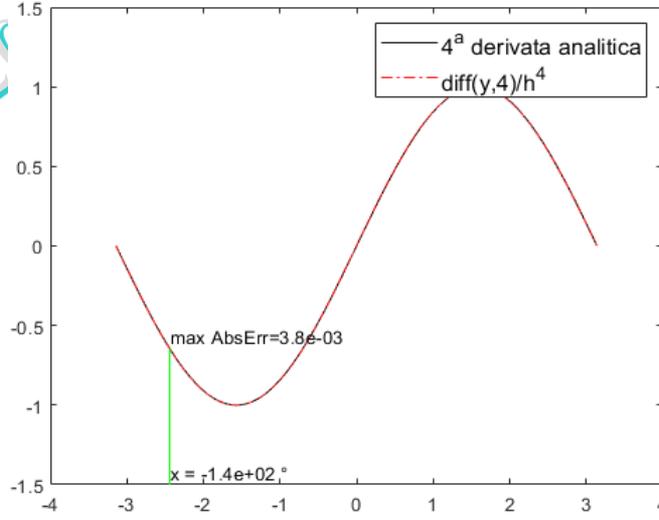
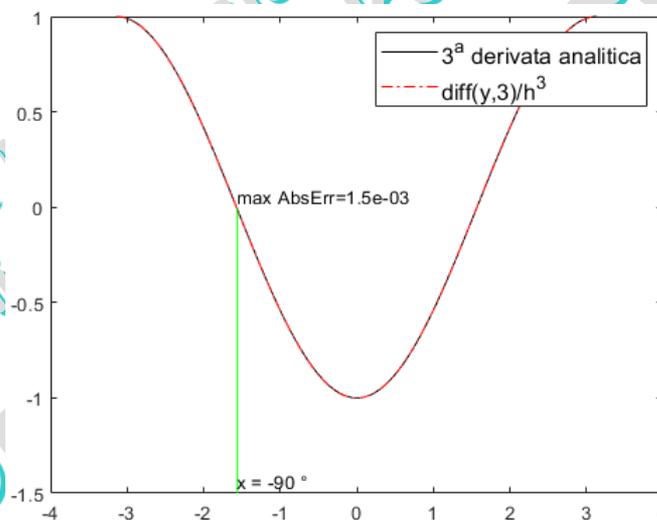
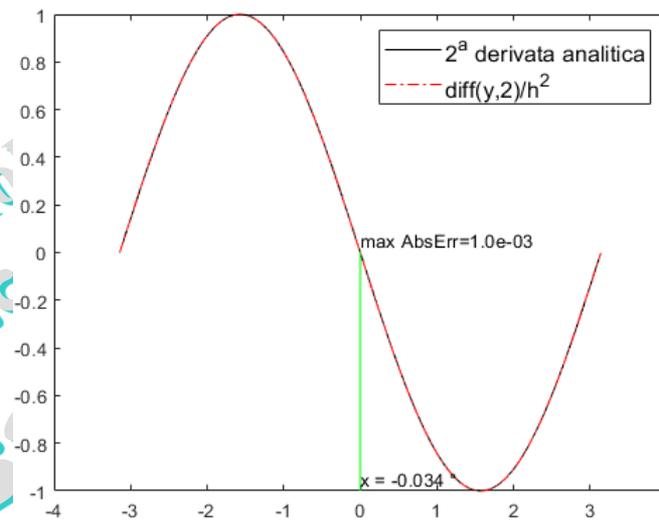
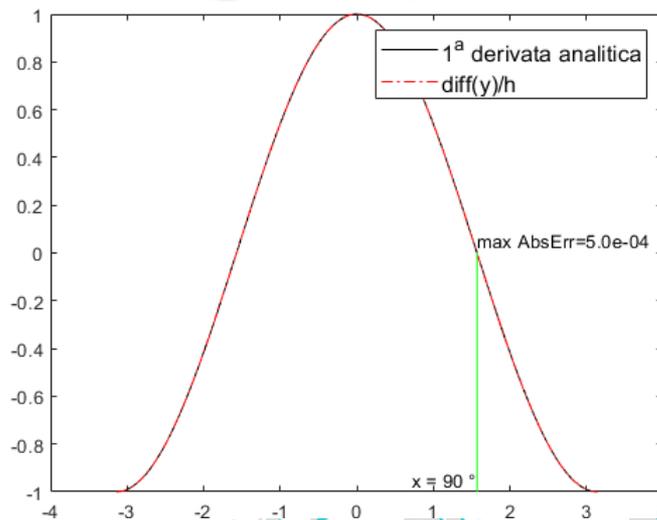
Esempio

Approssima le prime 4 derivate della funzione **sin(x)** e ne visualizza i grafici

```
h=0.001;
x=(-pi:h:pi)';
y=sin(x);
DF={cos(x) -sin(x) -cos(x) sin(x)}; % derivate vere per confronto
for n=1:4
    df=DF{:,n};
    dy=diff(y,n)/h^n;
    figure(n); clf
    plot(x(1:end-n),df(1:end-n),'k',x(1:end-n),dy,'-.r')
end
```

La funzione numerica **diff()** di MATLAB per approssimare le derivate di una funzione

Esempio: approssimare le derivate mediante **diff**



La funzione consente di calcolare le **differenze finite (in avanti)** su tutti i punti di griglia tranne l'ultimo

La funzione numerica **gradient()** di MATLAB per approssimare il gradiente di una funzione di più variabili

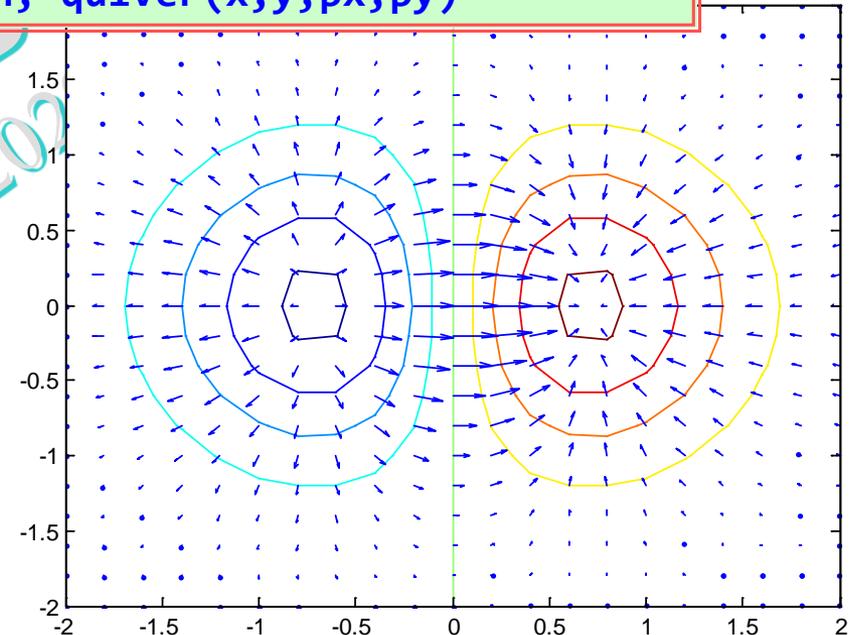
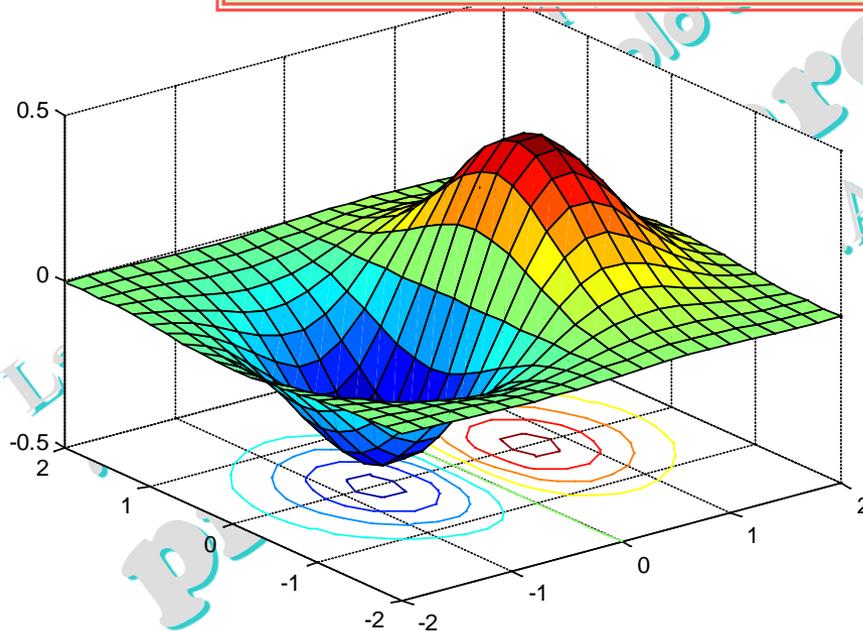
Il **gradiente** di una funzione $f(x,y) \in \mathbb{R}$ è il vettore $\nabla f = \text{grad } f = \begin{pmatrix} \frac{\partial f}{\partial x}(x,y) \\ \frac{\partial f}{\partial y}(x,y) \end{pmatrix}$

Del o nabra



il quale si può pensare come un vettore, nel dominio piano della funzione, che punta nella direzione in cui cresce la f

```
w=0.2; v=-2:w:2; [x,y]=meshgrid(v); z=x.*exp(-x.^2 - y.^2);  
[px,py]=gradient(z,w); figure; surfc(x,y,z);  
figure; contour(v,v,z), hold on, quiver(x,y,px,py)
```



gradiente come campo vettoriale

Come approssima le derivate **gradient()**?

`f=[1 5 3 4 2]';`
`f1=gradient(f, .5);`
`[f f1] =`

1	8
5	2
3	-1
4	-1
2	-4

agli estremi
 differenze in avanti

usa h=0.5

`f=[1 5 3 4 2]';`
`f1=gradient(f, .5);`
`[f f1] =`

1	8
5	2
3	-1
4	-1
2	-4

differenze centrali
 negli interni

usa h=0.5

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$$

nei punti di frontiera

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$

nei punti interni

`f=[1 5 3 4 2]';`
`f1=gradient(f);`
`[f f1] =`

1	4
5	1
3	-0.5
4	-0.5
2	-2

agli estremi
 differenze in avanti

usa h=1

`f=[1 5 3 4 2]';`
`f1=gradient(f);`
`[f f1] =`

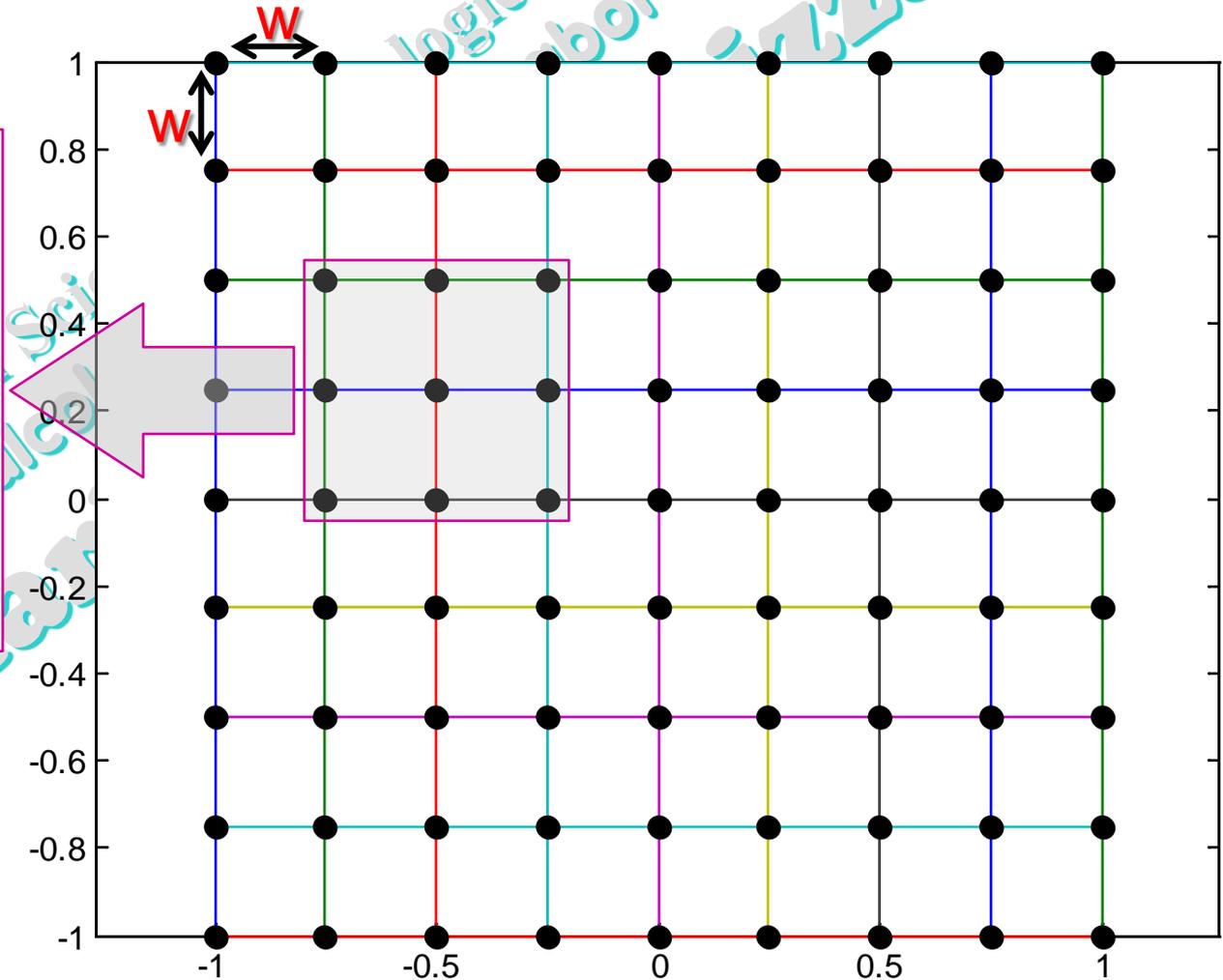
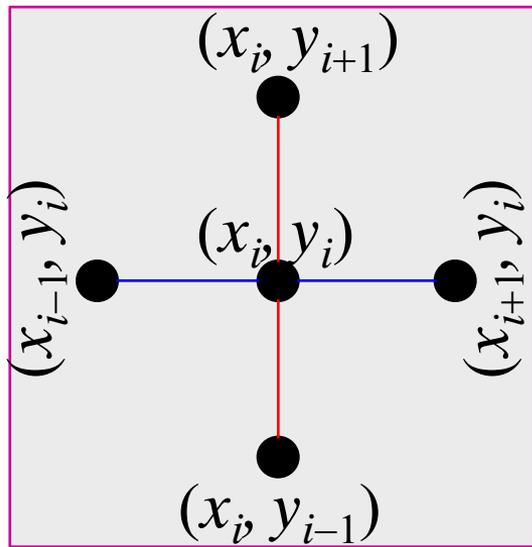
1	4
5	1
3	-0.5
4	-0.5
2	-2

differenze centrali
 negli interni

usa h=1

Esempio: una griglia quadrata

```
w=0.25; [x,y]=meshgrid(-1:w:1);  
h=plot(x,y,x',y',x,y,'ok'); axis equal;  
set(h,'MarkerFaceColor','k')
```

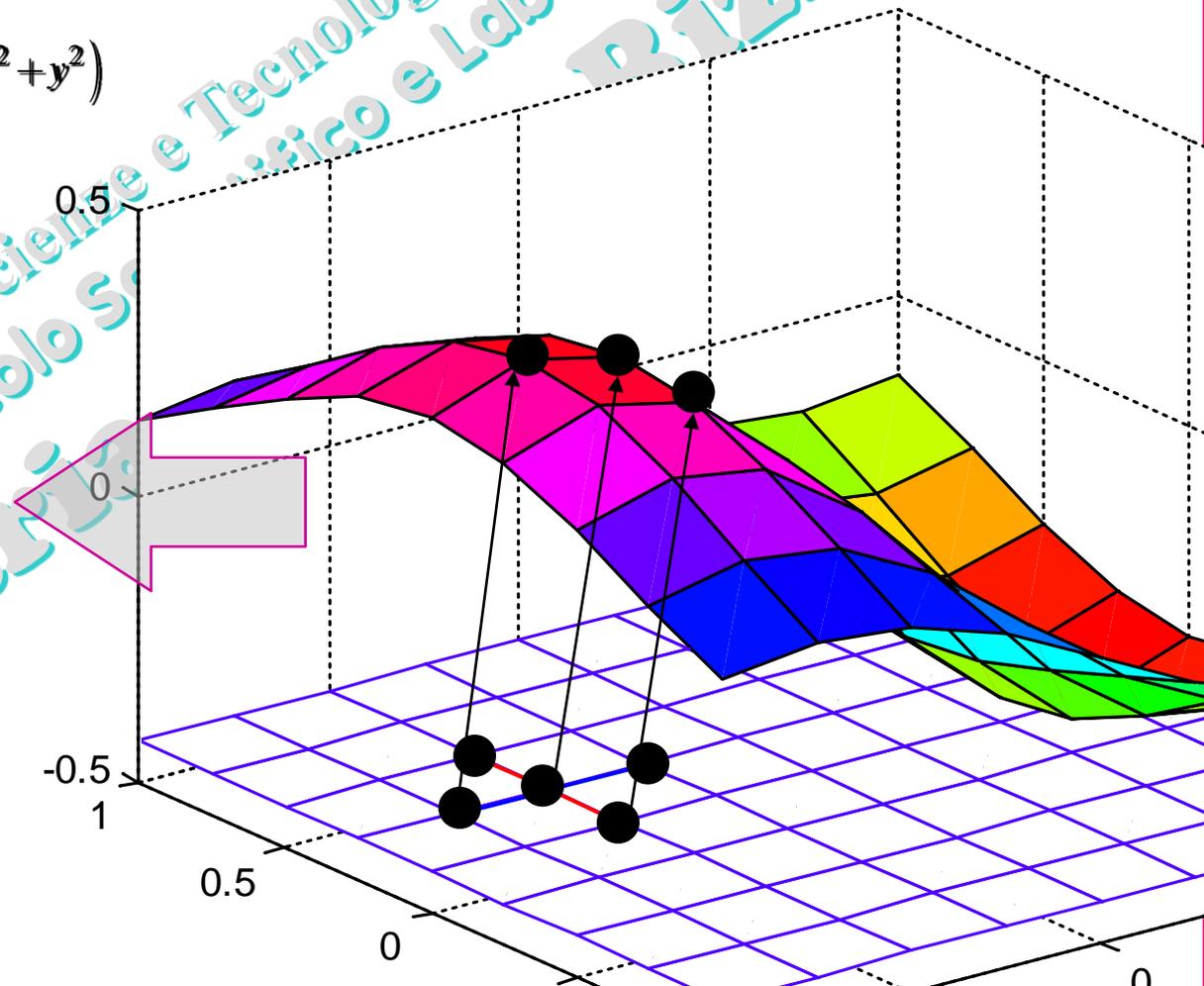
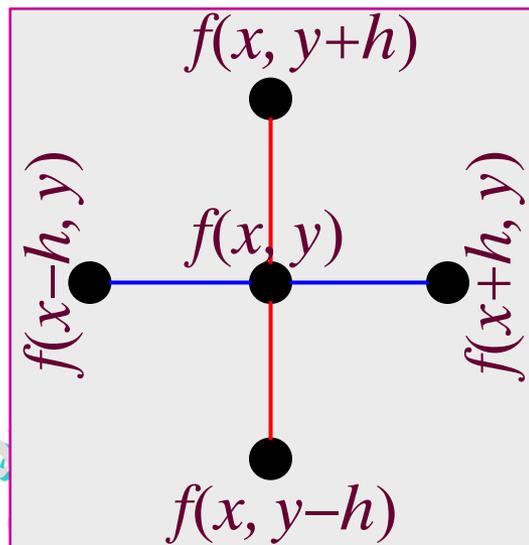


Approssimazione delle derivate parziali

```
h=0.25; [x,y]=meshgrid(-1:h:1); f=-x.*exp(-x.^2 - y.^2);  
[fx,fy]=gradient(f,w);
```

$$f_x \approx \frac{\partial f}{\partial x} = (2x^2 - 1)e^{-(x^2+y^2)}$$

$$f_y \approx \frac{\partial f}{\partial y} = 2xye^{-(x^2+y^2)}$$



Esempio: griglia $[-2,2] \times [-2,2]$

```
h=0.1; [x,y]=meshgrid(-2:h:2); z=x+i*y; f=z.^2;  
[dfdx, dfdy]=gradient(f,h); CR=abs(dfdx+i*dfdy);  
mesh(x,y,CR)
```

dovrebbe essere 0 per il Teor. delle Eq. di Cauchy-Riemann

sui **bordi** usa: $f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$

all'**interno** usa: $f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$

errore

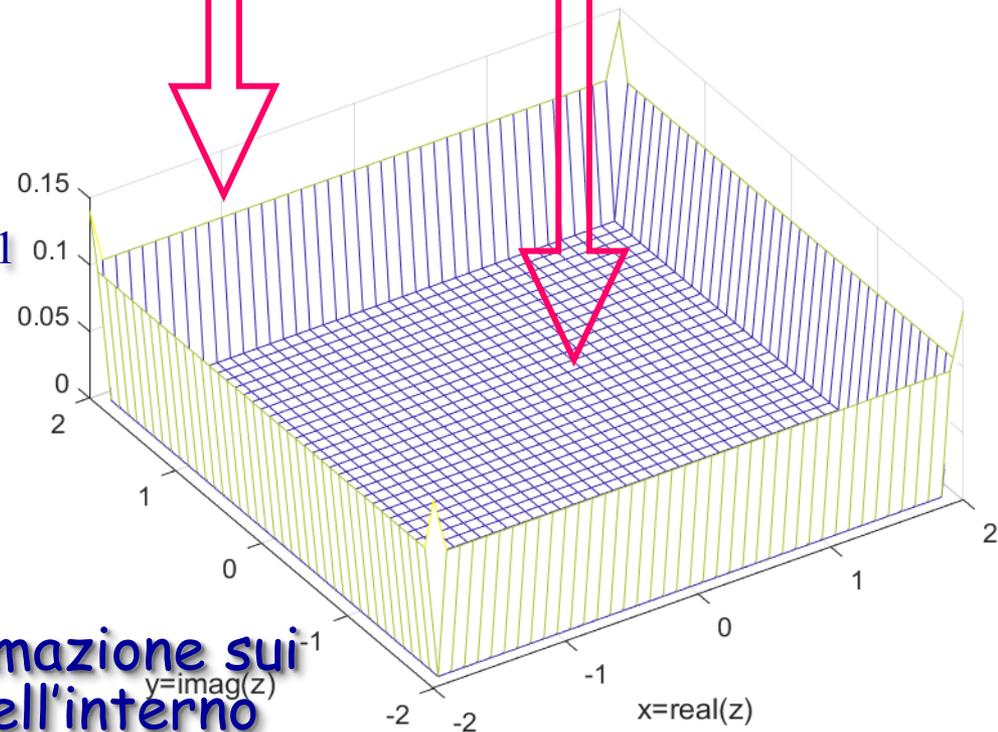
per h piccolo risulta:

$$O(h^2) \ll O(h)$$

$$O(0.01) \ll O(0.1)$$

↑

$$h = 0.1$$



L'**accuratezza** dell'approssimazione sui bordi è inferiore a quella nell'interno

Esempio: griglia $[-4,4] \times [-4,4]$

```
h=0.05; [x,y]=meshgrid(-4:h:4); z=x+i*y; f=z.^2;  
[dfdx, dfdy]=gradient(f,h); CR=abs(dfdx+i*dfdy);  
mesh(x,y,CR)
```

dovrebbe essere 0 per il Teor. delle Eq. di Cauchy-Riemann

sui **bordi** usa: $f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)$

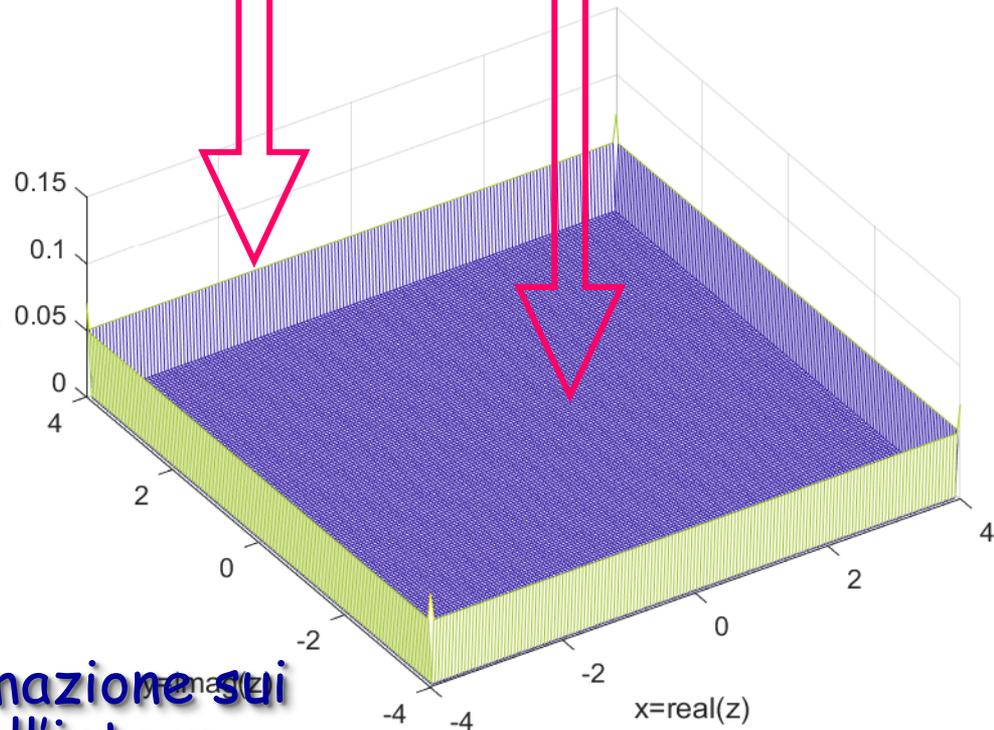
all'**interno** usa: $f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$

errore

per h piccolo risulta:

$$O(h^2) \ll O(h)$$

$h_2=0.05$



L'**accuratezza** dell'approssimazione sui bordi è inferiore a quella nell'interno

Esempio

Allargando il dominio (x,y) il fenomeno “scatola” si presenta nuovamente, ma sempre sulla frontiera: nella parte centrale il valore è minore (essendo dell'ordine di h^2) e sui bordi è maggiore (essendo dell'ordine di h).

