Derivazione Numerica

Derivazione Numerica

Supponiamo di voler approssimare la derivata di una funzione $f : \mathbb{R} \to \mathbb{R}$

Sappiano che la derivata è definita come segue:

$$\frac{df}{dx}(x) = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon) - f(x)}{\varepsilon}$$

Se scegliamo un valore di ε piuttosto piccolo (diciamo $\hat{\varepsilon}$) avremo che:

$$\frac{df}{dx}(x) \simeq \frac{f(x+\hat{\varepsilon}) - f(x)}{\hat{\varepsilon}}$$

- L'espressione a dx è facilmente calcolabile
- ...A meno di problemi numerici, che però non approfondiremo

In pratica si usano formule più numericamente stabili, ma a noi questa andrà bene


Derivazione Numerica

L'approccio è valido per qualunque funzione

...Per cui possiamo definire una procedura generica:

In [1]:
```
def num_der1(x, f, eps=1e-9):
    return (f(x + eps) - f(x)) / eps
```

- `x` rappresenta il punto per cui vogliamo calcolare la derivata
- `f` dovrà contenere la funzione da derivare
- `eps` è la differenza da usare per il calcolo approssimato


Derivazione Numerica

Al momento della chiamata possiamo decidere:

- Per quale punto calcolare la derivata (scegliendo cosa passare in `x`)
- Per quale funzione calcolare la derivata (scegliendo code passare in `f`)

In [2]:
```
import math

def num_der1(x, f, eps=1e-9):
    return (f(x + eps) - f(x)) / eps

def f1(x):
    return math.exp(x) * math.sin(x)

def f2(x):
    return x**2 * math.sin(x)

print(num_der1(1, f=f1))
print(num_der1(2, f=f1))
print(num_der1(1, f=f2))
```

```
3.7560496934929684
3.6439180561842472
2.2232444774061833
```

Esercizio: Derivazione Numerica

Si desidera definire un modulo con funzioni per derivazione numerica

- Per prima cosa, si proceda a creare il modulo `num_der` nel pacchetto `sol`
  - Per farlo, creare la cartella `sol` (se non ancora esistente)
  - Quindi, creare al suo interno un file di nome `__init__.py` (se non ancora esistente)
  - Infine, creare all'interno di `sol` il file `num_der.py`
- Per comodità, si abiliti l'estensione `autoreaload` di Jupyter

In [4]:
```
%load_ext autoreload
%autoreload 2
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```


Esercizio: Derivazione Numerica

Nel modulo `sol.num_der`, si definisca la funzione:

```
def num_der1(x, f, eps=1e-9)
```

La funzione deve approssimare la derivata usando la formula:

$$\frac{f(x+\varepsilon) - f(x)}{\varepsilon}$$

Si collaudi la funzione nella cella seguente:

- Se ne verifichi il funzionamento per $x \in \{1, 2, 3\}$
- Come funzioni da derivare si utilizzino:
  - $f1(x) = x\sin(x)$
  - $f2(x) = x\cos(x)$
- Le due funzioni non vanno definite nel modulo `sol.num_der`
- E per questa ragione sono già fornite nella cella

Esercizio: Derivazione Numerica (2)

Nel modulo `sol.num_der`, si definisca la funzione:

```
def num_der2(x, f, eps=1e-9)
```

La funzione deve approssimare la derivata usando la formula:

$$\frac{f(x + 0.5\varepsilon) - f(x - 0.5\varepsilon)}{\varepsilon}$$

Si collaudi la funzione nella cella seguente:

- Se ne verifichi il funzionamento per $x \in \{1, 2, 3\}$
- Come funzioni da derivare si utilizzino:
  - $f1(x) = x^2 \sin(x)$
  - $f2(x) = x^2 \cos(x)$
- Le due funzioni non vanno definite nel modulo `sol.num_der`
- Le si definisca insieme nella cella seguente insieme al codice di collaudo