Esame di Programmazione Informatica

23 luglio 2024

Esercizio 1 (15/30)

Date due generiche funzioni $f(x)$ e $g(x)$ derivabili nell'intervallo $[a, b]$, si dimostra che esiste almeno un punto $x_0 \in [a, b]$ tale che, se $f(a) \neq f(b)$ e $g(a) \neq g(b)$, allora:

$$\frac{f'(x_0)}{f(b) - f(a)} = \frac{g'(x_0)}{g(b) - g(a)} \tag{1}$$

dove f' e g' sono le derivate di f e g, rispettivamente.

Scrivere una funzione MATLAB che prenda come argomenti in ingresso:

- quattro *function handle* relativi a f, g, f', g'
- gli estremi a e b dell'intervallo

e restituisca come argomento in uscita il *function handle* di una funzione $h(x)$ che si annulla quando l'equazione (1) è verificata. (Suggerimento 1: spostare e/o riorganizzare i termini nell'equazione (1) in maniera tale da porla nella forma $h(x_0) = 0$; Suggerimento 2: implementare h come funzione anonima all'interno della funzione principale.)

Mostrare poi come utilizzare la precedente funzione per ottenere il *function handle* della funzione $h(x)$ nel caso di $f(x) = \sin(x)$, $g(x) = x^2$, $f'(x) = \cos(x)$, $g'(x) = 2x$, tutte implementate mediante funzioni anonime, ed intervallo $[a, b] = [0, 1]$. Come si potrebbe verificare che la funzione $h(x)$ così ottenuta si annulla almeno in un punto?

Soluzione: Consideriamo la funzione $h(x)$ definita come segue:

$$h(x) = \frac{f'(x)}{f(b) - f(a)} - \frac{g'(x)}{g(b) - g(a)}$$

che si annullerà in almento un punto $x_0 \in [a, b]$ e che verrà implementata mediante funziona anonima:

FunzioneLagrange.m

```
function h = FunzioneLagrange(f, g, df, dg, a, b)
   h = @(x) df(x) / (f(b)-f(a)) - dg(x) / (g(b)-g(a)) ;
end
```

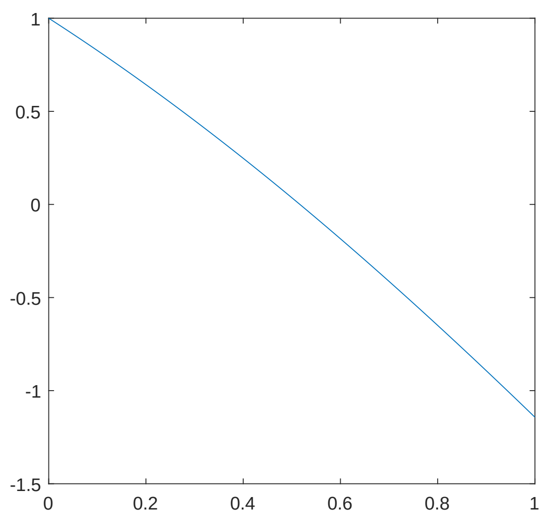
Utilizzo nel caso richiesto:

```
f = @(x) sin(x) ;
g = @(x) x.^2 ;
df = @(x) cos(x) ;
dg = @(x) 2*x ;
a = 0 ;
b = 1 ;
h = FunzioneLagrange(f, g, df, dg, a, b) ;
```

Per verificare che h si annulla in almeno un punto, possiamo fare un plot del grafico della funzione:

```
x = linspace(a, b, 1000) ;
plot(x, h(x)) ;
```

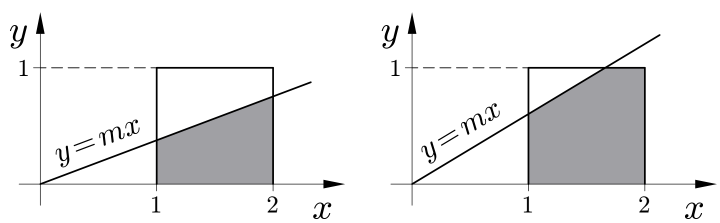
ottenendo il seguente grafico:



Esercizio 2 (18/30)

Scrivere una funzione MATLAB che prenda come argomento in ingresso il coefficiente angolare m della retta $y = mx$ nelle seguenti figure, e restituisca come argomento in uscita il valore dell'area colorata in grigio all'interno del quadrato, assumendo $m \geq 0$. Si dovranno quindi considerare i tre seguenti sottocasi:

- $0 \leq m \leq 1/2$ (primo grafico)
- $1/2 \leq m \leq 1$ (secondo grafico)
- $m \geq 1$ (quadrato completamente riempito)



Per la lode (3 punti): è possibile un'implementazione vettoriale che non faccia uso di esplicite istruzioni condizionali `if`/`if-else`/`if-elseif` nel caso di vettore m in ingresso? Mostrare tale implementazione.

Soluzione: Si ricava facilmente che le aree sono date dalle seguenti formule, che andranno implementate mediante degli `if-else` (oppure `if-elseif`):

- $0 \leq m \leq \frac{1}{2} \Rightarrow A = \frac{3m}{2}$
- $\frac{1}{2} \leq m \leq 1 \Rightarrow A = 2 - \frac{1}{m} + (\frac{1}{m} - 1)\frac{m+1}{2}$
- $m > 1 \Rightarrow A = 1$

areaSottesa.m

```
function A = areaSottesa(m)
  if m < 0.5
    A = 3*m/2 ;
  else
    if m < 1
      A = 2-1/m + (1/m-1)*(m+1)/2 ;
    else
      A = 1 ;
    end
  end
end
```

L'implementazione vettoriale è possibile utilizzando indici vettoriali ed operazioni vettoriali (elemento per elemento):

areaSottesaVettoriale.m

```
function A = areaSottesaVettoriale(m)
  A = 0*m ;
  % Primo caso m <= 0.5
  ix = m <= 0.5 ;
  A(ix) = 3*m(ix)/2 ;
  % Secondo caso 0.5 < m < 1
  ix = m > 0.5 & m <= 1 ;
  A(ix) = 2-1 ./ m(ix) + (1 ./ m(ix)-1) .* (m(ix)+1)/2 ;
  % Terzo caso m > 1
  ix = m > 1 ;
  A(ix) = 1 ;
end
```