Esercitazione 14 Maggio

Derivazione e integrazione

IN LABORATORIO

Derivazione, prova con diversi metodi e diverse funzioni.

Differenze finite di ordine 1 in avanti

Introdotta una griglia di punti $\{x_i\}$ nei quali possiamo valutare la funzione e nei quali vogliamo valutare la dervata, si ha che:

$$f'(x_i) \approx \delta_+(f)(x_i) \equiv \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$$

Se supponiamo gli $\{x_i\}$ equispaziati di spaziatura h si può riscrivere come:

$$f'(x_i) \approx \delta_+(f)(x_i) \equiv \frac{f(x_i + h) - f(x_i)}{h}$$

La stime dell'errore vista in aula per questo metodo è:

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

Scrivere una routine per implementare questo metodo (notare che non assegna l'ultimo indice, porre uguale a quello precedente). Un esempio di schema può essere:

```
function [out]=differaula(x,y)
% si considera che y siano le valutazioni della funzione sulla griglia x
inizializzazioni
ciclo sugli indici -> aggiornamento out
```

Testare il metodo su un polinomio di primo grado per controllare che sia esatto.

Fare vedere che dimezzando il passo l'errore si dimezza in un caso in cui il metodo non è esatto, ad esempio per le funzioni:

$$f(x) = x^7 - 6x^3 + 2 \tag{0.1}$$

$$f(x) = x^2 - 7x + 12 \tag{0.2}$$

la cui derivate esatte sono:

$$f'(x) = 7x^6 - 18x^2 \tag{0.3}$$

$$f'(x) = 2x - 7 \tag{0.4}$$

Per calcolare l'errore utilizzare il comando

`max(abs(der_esatta(1:end-1)-der_approx(1:end-1)))`

Differenze finite di ordine 2 centrate, caso nodi equispaziati:

$$f'(\bar{x}) \approx \delta(f)(x_i) \equiv \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

La stime dell'errore vista in aula per questo metodo è:

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

Modificare la function precedente per implementare questo metodo (notare che non assegna né il primo né l'ultimo indice, porre uguale al valore, rispettivamente, successivo e precedente). Ad esempio si puo' pensare ad una function del tipo $function[out] = differaula(x, y, flag)$.

Testare il metodo sul un polinomio di secondo grado visto prima per controllare che il metodo è esatto.

Scrivere uno script che calcoli l'errore commesso da ciascuno dei due metodi dato il numero di punti n su una griglia di punti equispaziati. Uno schema è:

```
fisso x


chiamo la routine per calcolare diffin1
chiamo la routine per calcolare diffin2
pongo err(1) l'errore delle differenze finite di ordine 1
pongo err(2) l'errore delle differenze finite di ordine 2
```

Modificare il precedente script facendo un ciclo sul numero di punti n, considerare, per esempio, potenze di 2: $n = 2^k, k = 3, \dots, 10$. Per fare un esempio non polinomiale modificare $f_2 = sin(20\pi x)$ la cui derivata esatta è $20\pi cos(20\pi x)$
Con i risultati ottenuti, fare plot normali e *loglog*. Vedere anche che i rapporti tendono a quelli studiati in teoria.

Integrazione.

Implementare i tre metodi di quadratura:

$$\int_a^b f(x)\,dx \approx (b-a)f(c)\,,\ c = \frac{(a+b)}{2} \qquad \text{(del punto medio)}$$

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}[f(a)+f(b)] \qquad \text{(dei trapezi)}$$

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2}[f(\gamma_0)+f(\gamma_1)] \qquad \text{(dei trapezi con nodi di Gauss)}$$

$$\gamma_0 = \frac{(a+b)}{2} - \frac{1}{\sqrt{3}}\left(\frac{b-a}{2}\right)\ ,\ \gamma_1 = \frac{(a+b)}{2} + \frac{1}{\sqrt{3}}\left(\frac{b-a}{2}\right)$$

Costruire una routine (modulo di quadratura locale) che abbia in input i dati $(a, b, f, flag)$ (la funzione come stringa) e restituisca la stima dell'integrale. La variabile $flag$ si riferisce al metodo da utilizzare.
Testare i metodi sul polinomio di settimo grado x^7-6x^3+2. Per calcolare in maniera esatta l'integrale si puo' procedere cosi':

```
effe='x.^7-6*x.^3+2';
inteffe='(1/8)*x.^8-(3/2)*x.^4+2*x';
x=b;
int_esatto=eval(inteffe);
x=a;
int_esatto=int_esatto-eval(inteffe);
```

```
clear x;
```

Utilizzare la routine per scrivere una formula composita e studiare l'errore al variare del numero di sottointervalli considerato. Utilizzare comandi del tipo:

```
x=linspace(a,b,n);
Int=0;
for i=1:n-1
    Iloc=quadrat(x(i),x(i+1),effe,flag);
    Int=Int+Iloc;
end
```