

Calcolo Numerico
TEST del 31 AGOSTO 2018

Cognome e nome _____ Matricola _____

Informatica

Postazione _____

FIRMA PER CONSEGNARE _____

FIRMA PER RITIRARSI _____

SI RACCOMANDA AGLI STUDENTI DI **commentare adeguatamente** SCRIPT E FUNCTION MATLAB.

- Creare una function di nome `simpson.m` che implementi l'algoritmo relativo alla formula composta di Cavalieri Simpson. La function deve avere come parametri in ingresso la funzione integranda `f`, gli estremi dell'intervallo di integrazione $[a, b]$ ed il numero m di suddivisioni dell'intervallo di integrazione. Il parametro in uscita deve essere l'approssimazione dell'integrale ottenuta I . La function avrà quindi la seguente intestazione:

```
function I = simpson (f,a,b,m);  
% CAVALIERI-SIMPSON COMPOSTA. Formula di Cavalieri-Simpson composta  
%  
% I = simpson (f,a,b,m);  
%  
% Dati di ingresso:  
% f: funzione integranda  
% a: estremo sinistro dell'intervallo di integrazione  
% b: estremo destro dell'intervallo di integrazione  
% m: numero di sottointervalli (pari!)  
% Dati di uscita:  
% I: approssimazione dell'integrale definito
```

Si osservi che numero m deve essere pari altrimenti si scriva il messaggio di errore `Errore: l'integrale non e' calcolato`, e si ponga `I=NaN`.

- Si implementi uno script `simpson_adattativa` in cui si assegnino come input
 - una funzione `f`,
 - gli estremi dell'intervallo di integrazione `a`, `b`,
 - la tolleranza `toll`,

e come output

- il vettore `I` composto da approssimazioni successive dell'integrale richiesto,
- valore di riferimento Q della soluzione esatta tramite la funzione Matlab `quadl`, con una tolleranza $TOL=10^{(-15)}$.
- il valore binario `flag`.

Le approssimazioni successive $I(n)$ dell'integrale $\int_a^b f(x)dx$ per $n = 1, 2, \dots$, saranno ottenute raddoppiando il numero di sottointervalli precedente e utilizzando `simpson`. Più precisamente, $I(1)$ sarà ottenuto per $m = 2$, $I(2)$ per $m = 4$ e in generale $I(n)$ per $m = 2^n$.

Si imponga `nmax=100` come massimo valore accettabile di n , ovvero l'ultimo valore che il codice può eventualmente calcolare è $I(nmax)$. Le iterazioni dovranno essere arrestate quando per $n = n^*$ la quantità $E_n = |I_{n+1} - I_n| < toll$.

Si ponga `flag=1` se il codice ha calcolato l'integrale richiesto con $E_n < toll$ in al più `nmax` iterazioni, `flag=0` altrimenti.

- Si implementi uno script `esempio` in cui si utilizzi `simpson` assegnando come input
 - la funzione $f(x) = x^{11/2}$,
 - gli estremi dell'intervallo di integrazione $a = 0$, $b = 1$,
 - `toll=10-8` quale tolleranza.

e come output

- il vettore `I` composto da approssimazioni successive dell'integrale richiesto,
- il valore di riferimento Q della soluzione esatta tramite la funzione Matlab `quadl`, con una tolleranza $TOL=10^{(-15)}$.

Quando i risultati ottenuti sono ritenuti corretti, `esempio` produca una figura che contenga in scala semi-logaritmica le coppie (n, E_n^*) per $n = 1, \dots, n^*$, essendo $E_n^* = |I_n - Q|$, $n = 1, \dots, n^*$ la successione degli errori assoluti ottenuta considerando le approssimazioni successive I_n ed il valore di riferimento Q calcolato tramite `simpson_adattativa`.

Si salvi la figura ottenuta in `myplot.jpg`.