Metodi Numerici per il Calcolo

Esercitazione 6: Integrazione Numerica

A.A.2024/25

Scaricare dalla pagina web del corso l'archivio matlab_mnc2425_6.zip e scompattarlo nella propria home directory. Verrà creata una cartella con lo stesso nome contenente alcuni semplici script e function Matlab/Octave. Si svolga la seguente esercitazione che ha come obiettivo quello di sperimentare l'integrazione numerica di funzioni e la sua applicazione nel disegno.

A. Integrazione Numerica

Si considerino le seguenti funzioni test di cui si vuole calcolare l'integrale definito sul loro dominio di definizione:

$f_1 = e^{\sqrt{x}}\sin(x) + 2x - 4$	$x \in [0, 12]$	$\int_0^{12} f_1(x)dx$	$= 68.3532891202483$
$f_2 = \frac{32}{1+1024x^2}$	$x \in [0, 4]$	$\int_0^{4} f_2(x)dx$	$= 1.56298398573480$
$f_3 = \frac{e^{-x}}{x}$	$x \in [1, 2]$	$\int_1^{2} f_3(x)dx$	$= 0.170483423687459$
$f_4 = \frac{4}{1+x^2}$	$x \in [0, 1]$	$\int_0^{1} f_4(x)dx$	$= 3.14159265358979$
$f_5 = \sqrt{(}1 - x^2)e^x$	$x \in [0, 1]$	$\int_0^{1} f_5(x)dx$	$= 1.24395050141647$

1. **Formule di Quadrature di Newton-Cotes**
   Le function `trapezi_comp.m` e `simpson_comp.m` implementano rispettivamente la formula dei trapezi e Simpson composte;

   - si eseguano le function in oggetto su alcune funzioni test analizzando gli output;
   - si analizzino gli script e si osservi la costruzione e rappresentazione grafica dei polinomi a tratti;
   - si analizzi lo script `err_trapezi_comp.m` (`err_simpson_comp.m`) che utilizza la function Matlab `integral` per ottenere un valore "esatto", richiama la function `trapezi_comp.m` (`simpson_comp.m`) per avere un valore approssimato e stampa l'errore di integrazione; si usi `help integral` per info sui parametri da dare.

     Ricordando le espressioni degli errore di integrazione per le formule dei trapezi e Simpson composte

     $$R_T = -\frac{b-a}{12}h^2 f^{(2)}(\eta), \qquad R_S = -\frac{b-a}{180}h^4 f^{(4)}(\eta));$$

     si modifichino gli script precedenti per stimare sperimentalmente che l'errore è di ordine 2 (caso trapezi) e ordine 4 (caso Simpson), cioè si riducono come h^2 e come h^4. (I due nuovi script/function si chiamino `err2_trapezi_comp.m`, `err2_simpson_comp.m`). (Sugg. si applichi la formula composta per valori di h che si dimezzano e si stampi il rapporto dei relativi errori; nella cartella sono presenti le derivate della funzione integranda; come possono essere utili?)

2. **Errore di Integrazione ed Estrapolazione di Richardson**
   Si consideri lo script `err2_trapezi_comp.m` dell'esercizio precedente.

   - Si modifichi il codice per aggiungere l'estrapolazione di Richardson

     $$\int_a^b f(x)dx = \frac{4T(h/2) - T(h)}{3} + O(h^4)$$

     e tre nuove colonne di stampa simili a quelle presenti (il nuovo script si chiami `err2_trapezi_rich.m`).

   - A partire dallo script `err2_simpson_comp.m` si realizzare uno script simile a quanto richiesto al punto precedente per aggiungere l'estrapolazione di Richardson, sapendo che:

     $$\int_a^b f(x)dx = \frac{16S(h/2) - S(h)}{15} + O(h^5).$$

     Il nuovo script si chiami `err2_simpson_rich.m`.

B. Lunghezza ed area di una curva di Bézier

1. Data una curva 2D di Bézier lo script `slung_bezier.m` calcola la sua lunghezza. (Sugg. si completi la function `norm_c1_val.m` in modo che possa essere passata come argomento alla builtin function `integral` di Matlab). Si modifichi poi lo script per calcolare la lunghezza di una curva 2D di Bézier a tratti (vedi il file `ppbez_esse.db` e lo script si chiami `slung_ppbez.m`)

2. Data una curva 2D di Bézier lo script `sarea_bezier.m` calcola la sua area. (Sugg. si completi la function `cxc1_val.m` in modo che possa essere passata come argomento alla builtin function `integral` di Matlab). Si modifichi poi lo script per calcolare l'area di una curva 2D di Bézier a tratti. (vedi il file `ppbez_esse.db` e lo script si chiami `sarea_ppbez.m`)

3. Data una curva di Bézier a tratti chiusa (vedi file `ppbez_square_smooth.db`), la si scali affinché abbia area/lunghezza predefinita, per esempio unitaria. La scala della curva sia fatta rispetto al suo baricentro. (Completare i codici `sarea_ppbez_unitaria.m`, `slung_ppbez_unitaria.m`). Si consideri poi il codice `sppbezmodel_curve2d_square_smooth.m` che è un esempio di modellazione procedurale.

4. Si consideri lo script `sppbezplot_tan.m` che legge il file `ppbez_square_smooth.db` contenente una curva chiusa di Bézier a tratti e la disegna. Siamo interessati a conoscere la sua orientazione per esempio determinando e disegnando il vettore tangente nel primo punto del primo tratto di Bézier che compone la curva. Si modifichi poi lo script per disegnare ogni tratto separatamente e i vettori tangenti nei suoi punti estremi; analizzare come sono i vettori tangenti nei punti di raccordo.

5. Lo script `sppbezfill.m` carica un disegno definito da più curve di Bézier a tratti e lo visualizza a colori. Dopo aver analizzato lo script e compreso cosa fa, lo si modifichi per calcolare l'area della regione colorata di rosso. Il nuovo script si chiami `sppbezfill_area.m`. (Sugg. Si consideri l'orientazione di ogni curva del disegno affinché l'area venga calcolata correttamente).