

Laboratorio di Calcolo Scientifico  
a.a. 2002/03  
HOMEWORK #2

Udine, 12 giugno 2003 (consegna 23 giugno 2003)

Prof.ssa R. Vermiglio, Dott. Ing. D. Breda  
Dipartimento di Matematica e Informatica,  
Università degli Studi di Udine  
vermiglio@dimi.uniud.it  
dbreda@dimi.uniud.it  
<http://www.dimi.uniud.it/~rossana>  
<http://www.dimi.uniud.it/~dbreda>

*NOTA: quando sono richieste implementazioni MATLAB che danno come output grafici, questi devono essere stampati e consegnati assieme all'homework (che può essere scritto a mano), includendo anche il testo dell'm-file e dei risultati numerici.*

1. Si consideri il problema dell'approssimazione di una quantità  $F$  mediante una formula  $F(h)$  tale che l'errore risulti

$$E(h) = F - F(h) = ch^2 + \mathcal{O}(h^4).$$

In particolare si vogliono approssimare le quantità

$$f'(0) = 1 \quad \text{e} \quad \int_0^{2\pi} f(x) dx = \frac{(3 - 10\pi)e^{-2\pi} - 3}{25}$$

dove

$$f(x) = xe^{-x} \cos 2x.$$

- i. Si scriva una *function* MATLAB con sintassi

`[Dh,EDh]=difcen(h)`

che implementi il metodo delle differenze centrate per l'approssimazione della derivata prima di  $f$  in 0. Sia  $h$  il passo utilizzato,  $Dh$  la derivata approssimata ed  $EDh$  l'errore assoluto commesso rispetto al valore esatto.

- ii. Si scriva una *function* MATLAB con sintassi

`[Th, ETh]=trapezi(h)`

che implementi la formula dei trapezi per approssimare l'integrale di  $f$  tra 0 e  $2\pi$ . Sia  $h$  il passo utilizzato,  $Th$  l'integrale approssimato ed  $ETh$  l'errore assoluto commesso rispetto al valore esatto.

- iii. Si analizzi l'andamento dell'errore assoluto in funzione del passo  $h$ , confermando con i risultati numerici che entrambe le *function* precedenti sono formule di approssimazione del tipo  $F(h)$  descritto sopra. In particolare si rappresenti su scala bilogaritmica l'errore assoluto in funzione del passo per un vettore di valori di  $h = (h_0, h_0/2, h_0/4, h_0/8, \dots)^T$  (con  $h_0$  opportunamente scelto) fino a raggiungere una tolleranza prefissata.

- iv. Si scriva una terza *function* con sintassi

`[Fh, EF]=richard(F, TOL, h0)`

che implementi l'estrapolazione di Richardson.  $h_0$  è un opportuno passo iniziale,  $TOL$  la precisione richiesta ed  $F$  la formula di approssimazione  $F(h)$ . L'algoritmo riportato in [1, §4.3] è il seguente:

$$\begin{aligned} F_{m,0} &= F(h_m), \quad m = 0, 1, \dots \\ F_{m,n+1} &= \frac{4^{n+1}F_{m,n} - F_{m-1,n}}{4^{n+1} - 1}, \quad n = 0, 1, \dots, m-1 \end{aligned}$$

con  $h_{m+1} = \frac{h_m}{2}$ ,  $m = 0, 1, \dots$  e  $h_0$  opportuno. *richard* costruisce la matrice delle approssimazioni

$$\begin{pmatrix} F_{0,0} & & & & \\ F_{1,0} & F_{1,1} & & & \\ F_{2,0} & F_{2,1} & F_{2,2} & & \\ F_{3,0} & F_{3,1} & F_{3,2} & F_{3,3} & \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

a cui viene aggiunta una riga (cioè si dimezza il passo  $h$ ) ogni volta che l'errore assoluto stimato dall'estrapolazione

$$EF_m = |F_{m,m} - F_{m-1,m-1}|, \quad m = 0, 1, \dots$$

risulta superiore alla tolleranza  $TOL$  richiesta in input. Quando l'errore risulta inferiore a  $TOL$ , l'esecuzione termina ed il valore approssimato di  $F$  sarà  $Fh = F_{m,m}$ . *richard* deve restituire in output anche il vettore degli errori  $EF = (EF_1, EF_2, \dots, EF_m)^T$  e deve provvedere alla rappresentazione grafica (su scala bilogaritmica) dell'andamento dell'errore in funzione del passo.

- v. Si esegua *richard* per le formule implementate da *difcen* e da *trapezi*, scegliendo opportunamente il passo iniziale. Si confrontino i corrispondenti andamenti degli errori in funzione del passo ottenuti al punto *iii* con quelli ottenuti al punto *v*. Si commentino i risultati ottenuti.
2. Si scriva una *function* MATLAB per interpolare i punti del piano  $P_i = (x_i, y_i)$ ,  $i = 0, 1, \dots, n$ , con le spline parametriche, prendendo come parametro la lunghezza cumulativa  $t$  data da

$$\begin{aligned} t_0 &= 0 \\ t_i &= \sum_{k=1}^i l_k, \quad i = 1, \dots, n \end{aligned}$$

dove  $l_i$  è la lunghezza di ciascun segmento  $\overline{P_i P_{i-1}}$ ,  $i = 1, \dots, n$ . La sintassi della *function* dev'essere

`[xx,yy]=spline_par(x,y)`

dove i vettori  $x$  e  $y$  rappresentano i punti da interpolare, mentre i vettori  $xx$  e  $yy$  rappresentano i valori assunti dalla spline parametrica in corrispondenza di un opportuno campionamento del parametro  $t$ .

Si esegua la *function* con i dati

$x_i$	8.125	8.4	9	9.445	9.6	9.959	10.166	10.2
$y_i$	0.0774	0.099	0.28	0.6	0.708	1.3	1.8	2.177

confrontando graficamente la spline parametrica con la spline ottenuta con il comando *spline* di MATLAB. Si verifichi inoltre la proprietà di invarianza geometrica delle spline parametriche (i.e. la curva interpolante ha la medesima rappresentazione indipendentemente dal sistema di coordinate usato), ruotando i punti di  $\theta = 36^\circ$  in senso antiorario. Ciò si può ottenere applicando ai vettori  $(x_i, y_i)^T$  la matrice di rotazione (di Givens, vedi [1, Appendice A.1.7])

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

per cui i nuovi punti sono individuati dai vettori

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = R(\theta) \begin{pmatrix} x_i \\ y_i \end{pmatrix}.$$

Si commentino i risultati ottenuti, in particolare si commenti il diverso comportamento tra spline (con opportuno codice MATLAB) e spline parametriche (con *spline\_par*) quando cambia il sistema di riferimento.

(Oss.: ogni punto  $(x, y)$  sul piano cartesiano identifica un vettore  $(x, y)^T \in \mathbb{R}^2$ . La rotazione del vettore di un angolo  $\theta$  (in senso antiorario) comporta

un cambio di coordinate  $(x, y) \mapsto (x', y')$ . Le nuove coordinate sono individuate dal vettore ruotato  $(x', y')^T$  come spiegato nel testo dell'esercizio e corrisponde alla rotazione del sistema di riferimento cartesiano di un angolo  $\theta$  in senso orario.

*Sugg.: per verificare l'invarianza geometrica si applichi il comando*

```
>>axis equal
```

*al grafico corrente: tale istruzione permette di utilizzare la stessa scala (incrementi uguali) per entrambi gli assi.)*

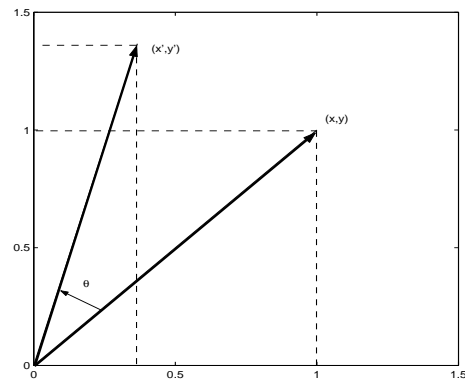


Figura 1: Rotazione antioraria del vettore  $(x, y)^T$  di un angolo  $\theta$ .

## Riferimenti bibliografici

- [1] V. Comincioli, *Analisi Numerica: Metodi, Modelli e Applicazioni*, McGraw-Hill (1990), Milano.