Correzione Esame di Calcolo Numerico

01/07/2019

1. In aritmetica di macchina $a * b = \text{Inf}$ poichè il massimo numero floating point rappresentabile è `realmax`$= 1.797693134862316e+308$ e $a*b$ è maggiore di `realmax`.

In aritmetica di macchina $p*q = 0$ poichè il più piccolo numero floating point in virgola mobile normalizzato rappresentabile è `realmin`$= 2.225073858507201e-308$. Rinunciando alla normalizzazione si possono rappresentare anche alcuni numeri più piccoli (`eps(realmin)`$= 4.940656458412465e-324$) ma $p*q$ è *molto* minore di `realmin`.

2. Risoluzione di sistemi lineari.
2a) Durante il corso abbiamo affrontato, con metodi numerici, il problema della risoluzione di sistemi lineari che, in forma matriciale, si può così descrivere:

Cercare il vettore soluzione x del sistema lineare $Ax = b$
con $A \in \mathbb{R}^{n\times n}$ matrice di rango n e $b \in \mathbb{R}^n$.

A causa degli errori di arrotondamento, implementando un metodo di risoluzione non si otterrà una soluzione esatta del sistema di partenza ma una soluzione approssimata $\widetilde{x}$. Il vettore residuo è definito:

$$r := b - A\widetilde{x}\,.$$

2b) Definita una norma vettoriale e una conseguente norma matriciale $\|\cdot\|$, il numero di condizionamento associato alla matrice A del sistema lineare definito al punto 2a) è definito come:

$$\text{cond}(A) := \|A\|\|A^{-1}\|$$

con A^{-1} matrice inversa di A.

$$\text{cond}(A) := \|A\|\|A^{-1}\| \geq \|AA^{-1}\| = \|I\| = 1$$

con $I \in \mathbb{R}^{n\times n}$ matrice identità.
2c)

$$\|M\|_\infty = \left\| \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix} \right\|_\infty = 3 \qquad \|M^{-1}\|_\infty = \left\| \begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1/3 \end{pmatrix} \right\|_\infty = 1/3$$

da cui

$$\text{cond}_\infty(M) = \|M\|_\infty \|M^{-1}\|_\infty = 1\,.$$

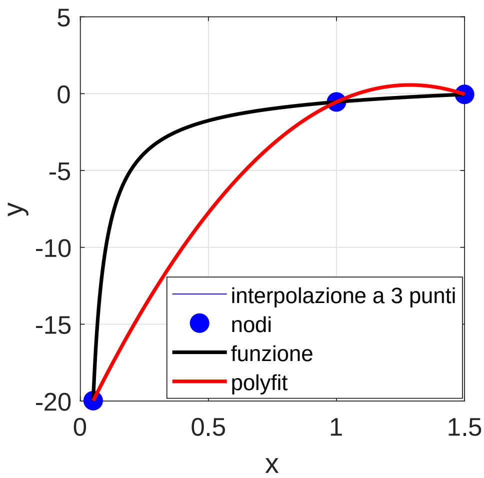
3. I comandi Matlab necessari per lo svolgimento dei punti 3b), 3c), 3d), 3e), 3f) e 3g) sono implementati nel file **Es3.m**.
3a) Dati 3 nodi di coordinate $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ tali che $x_i \neq x_j$ con $i \neq j$ e $i, j = 0, 1, 2$, i relativi polinomi fondamentali di Lagrange sono:

$$\mathcal{L}_i(x) = \prod_{j=0,1,2} \frac{x - x_j}{x_i - x_j} \qquad i = 0, 1, 2$$

e l'espressione del polinomio interpolante in forma di Lagrange è:

$$p(x) = y_0\mathcal{L}_0(x) + y_1\mathcal{L}_1(x) + y_2\mathcal{L}_2(x)\,.$$

3c) Il grafico del polinomio di secondo grado ottenuto con i 3 punti di interpolazione non si sovvrappone al grafico della funzione.
3d) I grafici del polinomio interpolante costruito con i polinomi fondamentali di Lagrange e del polinomio costruito con il comando `polyfit` si sovrappongono.



3g) La tabella risultante è

n. nodi	Errore `polyfit`	Errore Lagrange
5	$5.763e+00$	$5.763e+00$
10	$1.906e+00$	$1.906e+00$
15	$7.134e-01$	$7.134e-01$
20	$2.825e-01$	$2.825e-01$
25	$1.323e-01$	$1.158e-01$
30	$2.989e-01$	$4.850e-02$
35	$5.938e-01$	$2.065e-02$
40	$1.333e+00$	$8.920e-03$
45	$2.278e+00$	$3.887e-03$
50	$3.295e+00$	$1.776e-03$

L'aumento del grado del polinomio interpolante porta ad un iniziale abbattimento dell'errore con entrambi i metodi.
Con il comando `polyfit` si ha un successiva peggioramento dei risultati dovuti ad instabilità che invece non si riscontra con l'interpolazione lagrangiana. Questo fenomeno è dovuto al fatto che il comando `polyfit` ottiene il polinomio interpolatore attraverso la matrice di Vandermonde che al crescere del numero di nodi risulta fortemente mal condizionata.

4. 4a) Dagli sviluppi in serie di Taylor (supponendo f infinitamente derivabile)

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \sum_{k=3}^{\infty} \frac{h^k}{k!} f^{(k)}(x)$$
$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2} f''(x) + \sum_{k=3}^{\infty} (-1)^k \frac{h^k}{k!} f^{(k)}(x)$$

sommando la due equazioni, si ottiene

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + 2\sum_{k=2}^{\infty} \frac{h^{(2k)}}{(2k)!} f^{(2k)}(x)$$

da cui

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - 2\sum_{k=2}^{\infty} \frac{h^{(2k-2)}}{(2k)!} f^{(2k)}(x)\,.$$

Per h *abbastanza piccolo*, eliminando la sommatoria, si ottiene una *buona* approssimazione della derivata seconda della funzione f.

4b) La tabella di errore risultante dal codice implementato nel file **Es4.m** è:

x	Errore
$2.111111111111111e-01$	$1.394348357466611e+00$
$3.722222222222222e-01$	$2.299904736763381e-01$
$5.333333333333333e-01$	$9.942970218380721e-02$
$6.944444444444444e-01$	$5.532275091662133e-02$
$8.555555555555556e-01$	$3.454647017595924e-02$
$1.016666666666667e+00$	$2.287384089702700e-02$
$1.177777777777778e+00$	$1.558011729643994e-02$
$1.338888888888889e+00$	$1.072772504856716e-02$