PROVA DI LABORATORIO DI CALCOLO NUMERICO

INGEGNERIA MECCANICA - MATRICOLE PARI - AA 2016/17

Prof. Stefano De Marchi

`Padova, 28 agosto 2017`

- Il candidato dovrà produrre uno script `.m` per **ogni** esercizio.
- commentare **bene** gli scripts usando il comando %.
- al termine della prova lasciare tutti i files nella propria cartella `home`.
- **vietato usare libri, appunti e naturalmente il cellulare**

`ESERCIZI`

1. Si consideri la funzione $f(x) = (x-1)^2 \log x,\ x \in I = [0.5, 1.5]$ con log che indica il logaritmo naturale. Produrre lo script `esercizio1.m` che faccia le seguenti cose

   (a) Scelti 100 punti nell'intervallo $I = [0.5, 1.5]$, usando `linspace`, fa il plot della funzione. Si vedrà che la funzione ha un unico punto, $\alpha = 1$, dove si annulla. Approssimare α col *metodo iterativo di Newton* (di cui si fornisce la funzione `Newton.m`), con $x_0 = 0.5$, tolleranza `tol=1.e-8` e `kmax=20`. La chiamata della funzione `Newton` sarà

   ```
   [x1, k, err]=Newton(f, fp, x0, tol, kmax)
   ```

   dove le funzioni `f` e `fp` sono la funzione $f(x)$ e la derivata prima $f'(x)$, rispettivamente. Fare il grafico dell'errore `err` in scala semilogaritmica Che cosa si nota? [vale 5/30]

   (b) Modificare la funzione `Newton.m`, costruendo `Newton1.m`, che chiameremo col comando

   ```
   [x1, k, err, p, m]=Newton1(f, fp, x0, tol, kmax,alfa)
   ```

   per calcolare

   i. l'approssimazione dell'ordine p del metodo di Newton, calcolato usando la formula

   $$p_k \approx \lim_{k\to\infty} \frac{\log|x_{k+1}-\alpha|}{\log|x_k-\alpha|}$$

   (nota: $\alpha = 1$).

   ii. l'approssimazione della molteplicità della radice $\alpha = 1$ usando la relazione

   $$m_k = \frac{x_{k-1}-x_{k-2}}{2x_{k-1}-x_k-x_{k-2}}\,.$$

   In pratica p ed m saranno due vettori. [Vale 6/30]

   (c) Determinata la molteplicità della radice, modificare la funzione `Newton.m`, creando una nuova funzione `Newton2.m`

   ```
   [x1, k, err]=Newton2(f, fp, x0, tol, kmax, m)
   ```

   che riceva in input il valore intero della molteplicità m ottenuta al punto precedente, cosicchè il metodo ritorni ad essere un metodo di ordine quadratico. Fare il grafico dell'errore `err` in scala semilogaritmica. Quante iterazioni servono ora? [Vale 4/30]

2. Dati 5 punti equispaziati su $I = [0, \pi]$ e la funzione $f(x) = \sin(x^2 - \cos(x/2))$. Produrre lo script `esercizio2.m` che costruisce il polinomio d'interpolazione

   (a) risolvendo il sistema di Vandermonde che ha matrice

   ```
   V=fliplr(vander(x))
   ```

   con `x` il vettore dei punti equispaziati. Si valuti il polinomio su un insieme di 100 punti target mediante la tecnica di Horner (usare la funzione `Horner.m` allegata). Fare il plot della funzione e del polinomio d'interpolazione sui punti target, nonché dei punti d'interpolazione, calcolare l'errore d'interpolazione in norma infinito e il numero di condizionamento in norma 2 della matrice d'interpolazione. [vale 5/30]

   (b) costuendo i polinomi elementari di Lagrange valutati sugli stessi punti target del punto precedente, mediante la funzione allegata `Lagtarget.m` che consente poi di costruire il polinomio su punti target. Fare il plot della funzione e del polinomio d'interpolazione sui punti target, nonché dei punti d'interpolazione, calcolare l'errore d'interpolazione in norma infinito (fare attenzione alle dimensioni!) e il numero di condizionamento in norma 2 della matrice d'interpolazione. Cosa si nota circa gli errori e i numeri di condizionamento? [vale 5/30]

   (c) ripetere i precedenti punti per $n = 10 : 25$, quindi con un ciclo `for`, memorizzando i vari numeri di condizionamento ed errori sui `cV, error_V` per il punto (a) e `cL, error_L` per il punto (b), rispettivamente. Fare i plots comparativi, in scala semilogaritmica, dei numeri di condizionamento e dei vettori d'errore. Cosa si nota? [vale 6/30].

$\Diamond\Diamond$

Tempo: **2 ore**.