Esercizi per casa

25 novembre 2007

1 Introduzione a Matlab

1. *Facoltativo.* Si implementi una funzione `fatt` che calcola il fattoriale *fatt* di un numero naturale n, ricordando che

$$\mathrm{fatt}(n) := n! := 1 \cdot \ldots \cdot n.$$

   La si utilizzi per calcolare mediante un ciclo `for` i fattoriali dei numeri naturali n tra 1 e 20. Si visualizzino i risultati ottenuti. Per gli stessi numeri naturali si usi la funzione di Matlab `gamma` per valutare $\Gamma(n)$. Che relazione sussiste tra $\Gamma(n)$ e $\mathrm{fatt}(n)$?

2. *Facoltativo.* Si utilizzi la funzione fattoriale per calcolare il binomiale

$$A = \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

3. *Facoltativo.* Si considerino le matrici di Hilbert A_j per $j = 1, 2, \ldots, 20$. Si osservi cha la loro costruzione risulta facilitata dal comando `hilb` (si chiami l'help di Matlab per ulteriori informazioni). Utilizzando un ciclo `for`, si calcoli al variare di j il determinante $det(A_j)$ (via il comando `det`). Immagazzinare $det(A_j)$ in un vettore e visualizzare al variare di j tale risultato sul monitor (via il comando `fprintf`). Cosa succede se al posto di `det(A)` utilizzo `prod(eig(A))`? Ricordare che `eig(A)` fornisce il vettore degli n autovalori $\lambda_1, \ldots, \lambda_n$ della matrice quadrata A (avente n righe e n colonne) e che

$$det(A) = \lambda_1 \cdot \ldots \cdot \lambda_n.$$

2 Esercitazione sulla propagazione degli errori

1. Testare il codice `radicesecgrado.m` per l'esempio in cui

```
p=4.999999999995*10^(+4); q=10^(-2); sol=10^(-7);
```

2. Si calcoli l'errore relativo tra 1 e il valore che si ottiene valutando in Matlab/Octave
$$\frac{(1+\eta)-1}{\eta}$$
con $\eta = 10^{-1}, 10^{-2}, \ldots, 10^{-15}$. Si consideri poi $\eta = 8.8817841970012523E-16$ e si calcoli la medesima quantitá, giustificando i risultati ottenuti.

   Suggerimento: quanto vale `eps`?

3. Siano $f := \tan$ e $g := \arctan$. Si consideri la funzione composta
$$f \cdot g(x) := f(g(x)) = \tan(\arctan(x)) = x,\ x \in (-\infty, +\infty).$$
Si calcolino
$$x = 10^k,\ \text{per}\ k = -20+40h,\ \text{e}\ h = 0, 0.01, 0.02, \ldots, 1$$
e si valuti l'errore relativo compiuto. Si puó dire che anche *numericamente* $\tan(\arctan(x)) = x$?

3 Esercitazioni sulla complessità degli algoritmi

1. Si assegni `A=rand(n)` per $n = 10, 25, 50$ e quindi si digiti sulla shell di Matlab/Octave

```
[U,deter]=meg(A); s=deter;
```

   Di seguito si confronti il valore di `s` con quello di `t=det(A)` (ricordiamo che `det` è un comando Matlab che calcola il determinante di una matrice). Quante cifre hanno `s` e `t` in comune? Qual'è l'errore relativo $e_{\text{rel}} := |s-t|/|t|$? E quello assoluto $e_{\text{abs}} := |s-t|$?

2. Si implementino i due algoritmi proposti per il calcolo della potenza di matrice tramite due functions (senza usare l'operatore $\wedge$) e si calcoli l'errore relativo in norma infinito rispetto all'elevamento a potenza di Matlab o GNU Octave per diverse matrici e potenze ($n = 5, 10, 15, \ldots, 50$ e $p = 5, 10, 15, \ldots, 50$). Si confrontino poi i tempi di esecuzione delle due functions per il calcolo di A^{100}, con A matrice di numeri casuali di ordine 200.

4 Soluzione numerica di equazioni nonlineari

1. Si risolva numericamente il problema di calcolare la radice cubica di un numero, col metodo di Newton.

2. *Facoltativo.* Si risolva numericamente il problema di calcolare lo zero di $f(x) = \arctan(x)$, col metodo di Newton partendo prima da $x_0 = 1.2$ ed una seconda volta da $x_0 = 1.4$. Convergono in entrambi casi?

3. *Facoltativo.* Nota la soluzione del problema $x^2 - a = 0$ (si ottiene con il comando `sqrt(a)`) si stimi la velocita' di convergenza del metodo di Newton (**??**) per il calcolo della radice quadrata di un numero. A questo proposito conoscendo $|e_{k+1}|$, $|e_k|$, $|e_{k-1}|$ e ritenendo costante la costante asintotica di errore C calcolare un'approssimazione di p, via il calcolo di un logaritmo.

5 Interpolazione polinomiale

1. *Facoltativo.* Effettuare un programma Matlab che calcoli il valore assunto in 1.09 dal polinomio interpolatore della funzione log relativamente ai punti 1, 1.1, 1.2. E' buona l'approssimazione fornita dal polinomio interpolatore?

   Suggerimento: si noti che

   - il programma è una variante di quanto visto in `esperimento.m`, per una adeguata scelta dei nodi `x`, dei valori `y`, del punto `s`;
   - eliminare la parte relativa ai nodi di Chebyshev-Gauss-Lobatto;
   - il plot non è molto indicativo;
   - siccome il comando `norm(v,inf)` calcola per $v = \{v_i\}$ la quantità

     $$\|v\|_\infty = \max_i |v_i|$$

     visto che si deve valutare l'errore su un solo punto, il comando `norm(...,inf)` in `esperimento.m` può essere sostituito da `abs(...)` (rifletterci sopra);
   - la funzione `log` è predefinita in Matlab/Octave e quindi non serve ridefinirla come funzione.

2. *Facoltativo.* Aiutandosi con Matlab/Octave eseguire un programma che calcoli l'interpolante della funzione

   $$f(x) := \exp(x)$$

   relativamente alle ascisse $x = 1$, $x = 1.1$ ed $x = 1.2$. Si calcoli il polinomio di secondo grado p_2 che interpola tali punti, e lo si valuti in $s = 1.09$. Fornire una stima dell'errore e verificarne la bontà rispetto al risultato esatto.

3. *Facoltativo.* Fissati $a = -1$, $b = 1$, $n = 15$, si plottino i nodi di Chebyshev-Gauss. Si suggerisce di utilizzare la function `cheb.m` e per il plottaggio un comando del tipo

   ```
   plot(x,y,'r-o')
   ```

   Ricordiamo che

   - l'opzione `r-o`

(a) disegna un cerchietto rosso per ognuno dei punti (x_1, y_1), ..., (x_i, y_i), ..., (x_n, y_n);

(b) per $i = 1, \ldots, n-1$, unisce i punti (x_i, y_i), (x_{i+1}, y_{i+1}) con un segmento rosso.

- essendo importante disegnare solo le ascisse (e non le ordinate), quali ordinate si può porre

  ```
  y=zeros(size(x));
  ```

Una volta completato l'esercizio si osservi la particolare disposizione dei nodi di Chebyshev-Gauss. Si accumulano verso gli estremi dell'intervallo?