Prima prova di recupero relativa ai capitoli 1–3 con correzione

---

Esercizio 1 Dimostrare che, se f è una funzione sufficientemente regolare in un punto x assegnato, allora $[f(x+h)-f(x-h)]/(2h) = f'(x) + O(h^2)$.

Supponendo che la funzione $f(x)$ sia sufficientemente regolare, si ha:

$$\begin{aligned} f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2}f''(x) + O(h^3), \\ f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2}f''(x) + O(h^3). \end{aligned}$$

Quindi, sottraendo la seconda equazione dalla prima, si ottiene:

$$f(x+h) - f(x-h) = 2hf'(x) + O(h^3).$$

Dividendo membro a membro per $2h$ si ottiene il risultato.

Esercizio 2 Definire cosa sono l'*errore assoluto* e l'*errore relativo*. Spiegarne il significato.

Sia $x \in \mathbb{R}$ un dato valore numerico, che supponiamo sia approssimato con $\tilde{x}$. L'*errore assoluto* è definito come

$$\Delta x := \tilde{x} - x \qquad \Rightarrow \qquad \tilde{x} = x + \Delta x.$$

Se $x \neq 0$, il corrispondente *errore relativo* si definisce come

$$\varepsilon_x := \frac{\Delta x}{x} \qquad \Rightarrow \qquad \tilde{x} = x(1+\varepsilon_x).$$

Pertanto:

- $|\Delta x|$ misura la distanza dell'approssimazione dal dato esatto;
- $|\varepsilon_x|$ è una misura normalizzata dell'errore, che va rapportata ad 1.

Esercizio 3 Definire cosa misura la precisione di macchina di un'aritmetica finita. Quanto vale la precisione di macchina della *doppia precisione IEEE*?

I numeri reali che cadono nel *range* di rappresentazione dell'aritmetica finita, sia esso $\mathcal{I}$, costituiscono un insieme denso che viene approssimato da un insieme *finito* di *numeri di macchina*, che denoteremo con $\mathcal{M}$. Pertanto, l'implementazione dell'aritmetica finita prevede la definizione di una funzione, sia essa

$$f\ell : \mathcal{I} \to \mathcal{M},$$

che associa ad un numero reale $x \in \mathcal{I}$ un corrispondente numero di macchina $\tilde{x} = f\ell(x)$. Generalmente, si avrà quindi un *errore di rappresentazione*. Il valore assoluto del corrispondente errore relativo è uniformemente maggiorato dalla *precisione di macchina* dell'aritmetica finita, sia essa u. Quest'ultima è definita, se b è la base di rappresentazione utilizzata dall'aritmetica finita e si utilizzano m cifre per la mantissa, come:

- $u = b^{1-m}$, in caso di rappresentazione per *troncamento*;
- $u = \frac{1}{2}b^{1-m}$, in caso di rappresentazione per *arrotondamento*.

Nel caso della doppia precisione IEEE, si ha $b = 2$, $m = 53$, e si utilizza una rappresentazione per arrotondamento. Si ottiene, quindi, $u = 2^{-53} \approx 1.11 \cdot 10^{-16}$.

Esercizio 4 Scrivere una *function* Matlab che implementi efficientemente il metodo di bisezione per la ricerca di uno zero di una funzione.

```
function x = bisezione( a, b, f, tolx )
%
%   x = bisezione( a, b, f, tolx )  Metodo di bisezione per calcolare
%                                   una radice di f(x), interna ad [a,b],
%                                   con tolleranza tolx.
%
if a>=b, error('estremi intervallo errati'), end
if tolx<=0; error('tolleranza non appropriata'), end
fa = feval(f,a);
fb = feval(f,b);
if fa*fb>=0, error('intervallo di confidenza non appropriato'), end
imax = ceil( log2(b-a)-log2(tolx) );
if imax<1, x = (a+b)/2; return, end
for i = 1:imax
   x = (a+b)/2;
   fx = feval( f, x );
   f1x = abs(fb-fa)/(b-a);
   if abs(fx)<=tolx*f1x
```

```
      break
   elseif fa*fx<0
      b = x; fb = fx;
   else
      a = x; fa = fx;
   end
end
return
```

Esercizio 5 Dimostrare che il metodo di Newton converge quadraticamente ad una radice semplice.

Vedere il Teorema 2.1 del libro di testo.

Esercizio 6 Scrivere una function Matlab che, data in ingresso una matrice triangolare superiore U ed un vettore $\boldsymbol{b}$, calcoli efficientemente la soluzione del sistema lineare $U\boldsymbol{x} = \boldsymbol{b}$.

```
function x = Usolve( U, b )
%   function x = Usolve( U, b )   Risolve il sistema lineare U*x = b
%                                 con U matrice triangolare superiore
[m,n] = size(U);
k = length(b);
if m~=n || n~=k, error('dati non consistenti'); end
x = b(:);
for i=n:-1:1
     if U(i,i)==0, error('matrice singolare'); end
     x(i) = x(i)/U(i,i);
     x(1:i-1) = x(1:i-1)-x(i)*U(1:i-1,i);
end
return
```

Esercizio 7 Dimostrare che, se A è una matrice nonsingolare fattorizzabile LU, la sua fattorizzazione è unica.

Vedere il Teorema 3.1 del libro di testo.

Esercizio 8 Dimostrare che una matrice a diagonale dominante è fattorizzabile LU.

Ricordando che una matrice è fattorizzabile LU se tutti i suoi minori principali sono diversi da zero, ed osservando che le sottomatrici principali di una matrice a diagonale dominante sono a diagonale dominante, vedere il Lemma 3.9 del libro di testo.

Esercizio 9 Scrivere la fattorizzazione QR per il vettore $\boldsymbol{a} = \begin{pmatrix} 1, & -4, & 0 \end{pmatrix}^\top$ (ovvero, determinare Q ed R tali che $QR = \boldsymbol{a}$), utilizzando il metodo di Householder.

Indicando con $\boldsymbol{e}_1 \in \mathbb{R}^3$ il primo versore della base canonica, il vettore di Householder

$$\boldsymbol{v} = \boldsymbol{a} + \|\boldsymbol{a}\|_2 \boldsymbol{e}_1 \equiv \begin{pmatrix} 1+\sqrt{17} \\ -4 \\ 0 \end{pmatrix},$$

definisce una corrispondente matrice di Householder,

$$H = I + \frac{2}{\boldsymbol{v}^\top \boldsymbol{v}} \boldsymbol{v}\boldsymbol{v}^\top,$$

tale che

$$H\boldsymbol{a} = -\|\boldsymbol{a}\|_2 \boldsymbol{e}_1 \equiv \begin{pmatrix} -\sqrt{17} \\ 0 \\ 0 \end{pmatrix} =: R.$$

Pertanto, ricordando che $H^\top = H$, $HR = \boldsymbol{a}$ è la fattorizzazione QR cercata.
