Compito di Analisi Numerica, a.a.2014-2015, Appello 4, 7/7/2015

Esercizio 1.
Per approssimare uno zero reale α di una funzione $f(x) : \mathbb{R} \to \mathbb{R}$ sufficientemente regolare tale che $f'(x) \neq 0$ per $x \neq \alpha$, si considerino le seguenti due modifiche dell'iterazione di Newton in cui si è posto $n(x) = x - f(x)/f'(x)$:

$$x_{k+1} = n(x_k) - f(x_k)/f'(n(x_k)),$$

$$y_{k+1} = n(y_k) - f(n(y_k))/f'(y_k),$$

a partire da x_0 e y_0 in un intorno di α.

a) Se α è uno zero semplice di $f(x)$, dire se i due metodi hanno convergenza locale, studiare il loro ordine di convergenza e confrontare la loro efficienza con quella del metodo di Newton valutando il costo computazionale per passo in termini del numero di valutazioni di funzione (sia $f(x)$ che $f'(x)$).

b) Si risponda alla stesse domande del punto a) nel caso in cui α è uno zero di molteplicità 2.

Esercizio 2.
Sia $A = (a_{i,j})$ la matrice $n \times n$ con elementi $a_{i+1,i} = 1$ per $i = 1, \dots, n-1$, $a_{i,i} = u_i$, per $i = 1, \dots, n$, $a_{i,j} = v_i$ per $j > i$, $a_{i,j} = 0$ altrove.

a) Se $u_i = v_i$ per $i = 1, \dots, n-1$, dare condizioni sui parametri u_i affinché esista unica la fattorizzazione LU di A.

b) Nel caso generale in cui u_i non è necessariamente uguale a v_i, descrivere la matrice ottenuta dopo un passo di eliminazione Gaussiana. Si descriva poi un algoritmo per calcolare gli elementi delle matrici L ed U della fattorizzazione $A = LU$ nel caso essa esista e si valuti il costo computazionale.

c) Descrivere un algoritmo basato sulla fattorizzazione LU per risolvere il sistema lineare $Ax = b$ che impieghi $O(n)$ operazioni aritmetiche.

Esercizio 3.
Sia $A_n = (a_{i,j})$ matrice $n \times n$, $n \geq 2$, tale che $a_{i+1,i} = w_i$, per $i = 1, \dots, n-1$, $a_{1,i} = u_i$, $i = 1, \dots, n-1$, $a_{i,n} = v_i$ per $i = 1, \dots, n$, $a_{i,j} = 0$ altrove.

a) Applicando la regola di Laplace sull'ultima riga di A_n si metta in relazione $\det A_n$ con $\det A_{n-1}$.

b) Verificare che la formula ottenuta per il calcolo di $\det A_n$ richiede al più $5n$ operazioni aritmetiche.

c) Scrivere una function nella sintassi di Octave che la implementa.

Risoluzione

Esercizio 1. Assumendo $f(x)$ sufficientemente regolare e $f'(\alpha) \neq 0$, si ha

$$n'(x) = f(x)f''(x)/f'(x)^2,$$
$$n''(\alpha) = f''(\alpha)/f'(\alpha)$$

a) Con queste notazioni il primo metodo è individuato dalla funzione $g(x) = n(x) - f(x)/f'(n(x))$ vale allora

$$g'(x) = n'(x) - f'(x)/f'(n(x)) + f(x)f''(n(x))n'(x)/f'(n(x))^2$$

da cui, $g'(\alpha) = -1$. Questo implica che se il metodo converge allora la convergenza è sublineare. Non ci sono vantaggi rispetto al metodo di Newton. In generale il metodo non è convergente. Ad esempio con $f(x) = x$ vale $g(x) = -x$, per cui la successione generata a partire da x_0 vale alternativamente x_0 e $-x_0$.

Per il secondo metodo in cui $g(x) = n(x) - f(n(x))/f'(x)$, si ha

$$g'(x) = n'(x) - f'(n(x))n'(x)/f'(x) + f(n(x))f''(x)/f'(x)^2$$

per cui $g'(\alpha) = 0$ inoltre

$$g''(\alpha) = 0$$

per cui questo metodo è almeno del terzo ordine ed è più efficiente del metodo di Newton essendo $\log_3 2 < \log_2 2$.

b) Se lo zero ha molteplicità 2 allora $n'(\alpha) = 1/2$. Per il secondo metodo, usando la regola di de L'Hopital risulta $\lim_{x\to\alpha} \frac{f'(n(x)}{f'(x)} = n'(\alpha) = 1/2$, mentre $\lim_{x\to\alpha} \frac{f(n(x)}{f'(x)^2} = \frac{1}{8f''(\alpha)}$. Per cui $g'(\alpha) = 3/8$. La convergenza è lineare con un fattore di convergenza pari a $3/8$.

Analogamente per il primo metodo risulta $g'(\alpha) = -1/2$.

Esercizio 2.

a) Se $u_i = v_i$ per $i = 1, \ldots, n-1$, posto $A_n = A$ e denotando con $d_n = \det A_n$, dalla regola di Laplace applicata sull'ultima riga di A_n segue $d_n = u_n d_{n-1} - d_{n-1}$ da cui $d_k = u_1 \prod_{i=2}^{k}(u_i - 1)$ per $k \geq 2$. La matrice A_k è non singolare se e solo se $u_1 \neq 0$ e $u_i \neq 1$ per $i = 2, \ldots, k$. Per il teorema di esistenza e unicità della fattorizzazione LU si ha che se $u_1 \neq 0$ e $u_i \neq 1$ per $i = 2, \ldots, n-1$ esiste ed è unica la fattorizzazione LU.

b) La matrice $A^{(2)}$ che si ottiene dopo un passo di eliminazione gaussiana è

$$\begin{bmatrix} u_1 & v_1 & v_1 & v_1 & \ldots, & v_1 \\ & u_2' & v_2' & v_2' & \ldots & v_2' \\ & 1 & u_3 & v_3 & \ldots & v_3 \\ & & \ddots & \ddots & \ldots & \vdots \\ & & & 1 & u_{n-1} & v_{n-1} \\ & & & & 1 & u_n \end{bmatrix}$$

dove $u_2' = u_2 - v_1/u_1$, $v_2' = v_2 - v_1/u_1$. La sottomatrice principale di coda di dimensione $n-1$ ha quindi la stessa struttura della matrice A_n ed è definita dai vettori $(u_2', u_3, \ldots, u_n)$ e $(v_2', v_3, \ldots, v_{n-1})$. Quindi il metodo per calcolare il fattore U è dato da:

$$u_i' = u_i - v_{i-1}'/u_{i-1}', \quad v_i' = v_i - v_{i-1}'/u_{i-1}'$$

la matrice U ha elementi diagonali $u_1, u_2', \ldots, u_n'$ e sulla riga i-esima, nella parte triangolare superiore ha elementi uguali a v_i', dove $v_1' = v_1$. La matrice L ha elementi diagonali uguali a 1, e elementi sottodiagonali sulla i-esima colonna uguali a v_i'/u_i' per $i = 2, \ldots, n-1$ e v_1/u_1 per $i = 1$.

c) L'algoritmo per risolvere il sistema $Ax = b$ calcola prima i fattori L e U mediante le formule del punto b). Successivamente risolve i due sistemi lineari $Ly = b$, $Ux = y$. Il primo sistema fornisce le relazioni $y_1 = b_1$, $y_i = b_i - l_{i,i-1}y_{i-1}$, $i = 2, \ldots, n$ che richiedono al più $2n-1$ operazioni aritmetiche. Il sistema $Ux = y$ si risolve con una sostituzione all'indietro ponendo $x_n = y_n/u_n'$, $x_i = (y_i - v_i' \sum_{j=i+1}^n x_j)/u_i'$, per $i = n-1, \ldots, 1$. La somma $s_i = \sum_{j=i+1}^n x_j$ viene calcolata nel modo seguente

$$s_{n-1} = x_n, \quad s_i = s_{i+1} + x_i, \quad i = n-2, \ldots, 1$$

Il costo totale della risoluzione di $Ux = y$ è di al più $4n$ operazioni aritmetiche, infatti per il calcolo di x_i basta una sola addizione per aggiornare s_i, una moltiplicazione di v_i' per s_i, una addizione e una divisione. Mentre il clacolo della fattorizzazione LU richiede l'esecuzione di 2 operazioni per passo. Il totale, a meno di costanti additive, è quindi di $8n$ operazioni aritmetiche.

Esercizio 3.

La matrice ha la forma (caso $n = 4$)

$$\begin{bmatrix} u_1 & u_2 & u_3 & v_1 \\ w_1 & 0 & 0 & v_2 \\ 0 & w_2 & 0 & v_3 \\ 0 & 0 & w_3 & v_4 \end{bmatrix}$$

Posto $d_n = \det A_n$ vale $d_n = v_n c_{n-1} - w_{n-1}e_{n-1}$ dove c_{n-1} è il determinante della sottomatrice ottenuta cancellando ultima riga e colonna di A_n e quindi vale $c_{n-1} = (-1)^{n+1}u_{n-1}w_1w_2\cdots w_{n-2}$. Il valore di e_{n-1} è dato dal determinante della sottomatrice ottenuta cancellando l'ultima riga e la penultima colonna di A_n. Nel caso $n = 4$ questa sottomatrice è data da

$$\begin{bmatrix} u_1 & u_2 & v_1 \\ w_1 & 0 & v_2 \\ 0 & w_2 & v_3 \end{bmatrix}$$

cioè si ottiene d_{n-1}. Quindi l'espressione diventa

$$d_n = (-1)^{n+1}v_n u_{n-1}w_1w_2\cdots w_{n-2} - w_{n-1}d_{n-1}$$

con $d_1 = u_1$, $d_2 = u_1 v_2 - w_1 v_1$. Il calcolo di d_n richiede al più $5n$ operazioni aritmetiche. Infatti la produttoria $s_{n-1} = w_1 w_2 \cdots w_{n-2}$ si ottiene da s_{n-2} mediante una moltiplicazione. Cioè si ha

$$d_k = v_k u_{k-1} s_{k-1} - w_{k-1} d_{k-1}, \quad s_k = -s_{k-1} w_{k-1}, \quad k = 3, \ldots, n.$$

mentre $d_2 = u_1 v_2 - w_1 v_1$, $s_1 = -w_1$.

```
function d=determinante(u,v,w)
n = length(v);
d = u(1)*v(2)-w(1)*v(1);
s = -w(1);
for k=3:n
    d = v(k)*u(k-1)*s - w(k-1)*d;
    s = -s*w(k-1);
end
```