Risoluzione del compito di Analisi Numerica, a.a.2014-2015, Appello 1 del 13/1/2014

Esercizio 1.
a) Dimostriamo che $|x_i - \alpha| \le \rho\lambda^i$. Si procede per induzione su i come nella dimostrazione del teorema del punto fisso delle dispense. Per $i = 0$ ciò è vero essendo $x_0 \in [\alpha - \rho, \alpha + \rho]$. Si supponga che $|x_{i-1} - \alpha| \le \rho\lambda^{i-1}$. Vale $x_i - \alpha = g_i(x_{i-1}) - \alpha = g_i'(\xi_i)(x_{i-1} - \alpha)$, dove abbiamo applicato il teorema di Lagrange per cui ξ_i appartiene all'intervallo aperto di estremi x_{i-1} ed α che è contenuto in $[\alpha - \rho, \alpha + \rho]$. Dall'ipotesi induttiva e dal fatto che $|g_i'(\xi_i)| \le \lambda$, ricaviamo $|x_i - \alpha| = |g_i'(\xi_i)| \cdot |x_{i-1} - \alpha| \le \lambda \cdot \lambda^{i-1} = \lambda^i$. Poiché $\lambda < 1$ ne segue la convergenza di tutte le successioni generate a partire da $x_0 \in [\alpha - \rho, \alpha + \rho]$.

b) Si considera $\lim_i \frac{x_i - \alpha}{x_{i-1} - \alpha} = \lim_i \frac{g_i(x_{i-1}) - \alpha}{x_{i-1} - \alpha}$. Essendo $g_i(x)$ derivabile due volte con continuità applicando il teorema di Taylor possiamo scrivere $g_i(x_{i-1}) = g_i(\alpha) + (x_{i-1} - \alpha)g_i'(\alpha) + (x_{i-1} - \alpha)^2 g_i''(\eta_i)$ per un opportuno η_i nell'intervallo aperto di estremi x_{i-1} e α. Si ha quindi Poiché vale $|g_i''(x)| \le \gamma$ ne segue $\lim_i (x_{i-1} - \alpha)g_i''(\eta_i)) = 0$ da cui

$$\lim_i \left| \frac{x_i - \alpha}{x_{i-1} - \alpha} \right| = \lim_i |g_i'(\alpha) + (x_{i-1} - \alpha)g_i''(\eta_i)| = \lim_i |g_i'(\alpha)| \le \lambda < 1.$$

Per definizione di ordine di convergenza ne segue che se $\lim_i g_i'(\alpha) \ne 0$ la convergenza è lineare, se $\lim_i g_i'(\alpha) = 0$ la convergenza è superlineare.

c) In generale non si ha convergenza. Se ad esempio $g_i(x) = \alpha + \lambda_i(x - \alpha)$ si ha $x_i - \alpha = (x_0 - \alpha)\prod_{j=1}^{i} \lambda_j$. Scegliendo $\lambda_i = t_i/t_{i+1}$ dove $t_i > 0$ è una successione che converge crescendo a 1. risulta

$$\prod_{j=1}^{i} \lambda_j = \frac{t_1}{t_{i+1}} \to t_1 \ne 0$$

Per cui non c'è convergenza ad α

Esercizio 2.
a) La prima colonna di A è $a_{1,1}e_1 + a_{n,1}e_n$. Per cui la matrice elementare di Gauss E_1 cercata è $E_1 = I - ue_1^T$ dove $u = \frac{a_{n,1}}{a_{1,1}}e_n$. Risulta allora

$$A_2 = E_1A_1 = \left[\begin{array}{c|ccc} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ \hline 0 & & & \\ \vdots & & B & \\ 0 & & & \end{array}\right], \quad B = \begin{pmatrix} a_{2,2} & \dots & \dots & \dots & a_{2,n} \\ 0 & a_{3,3} & \dots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,2}^{(2)} & \dots & \dots & \dots & a_{n,n}^{(2)} \end{pmatrix}$$

dove

$$a_{n,j}^{(2)} = a_{n,j} - \frac{a_{n,1}}{a_{1,1}}a_{1,j}, \quad j = 2, \dots, n. \tag{1}$$

Analogamente, se E_1 è scelta nella classe delle matrici di Householder, è per definizione $E_1 = I - \beta vv^T$ dove $v = [a_{1,1} - \alpha, 0, \ldots, 0, a_{n,1}]^T$ e $\beta = 2/v^T v$. Risulta allora

$$A_2 = A_1 - \beta vv^T A_1.$$

Poichè il vettore v ha eventualmente solo la prima e l'ultima componente diverse da zero, la matrice A_2 ha le righe dalla seconda alle $n-1$-esima uguali a quelle della matrice A_1. Pertanto la sottomatrice B ha la struttura desiderata. Si osserva inoltre che

$$\begin{aligned} a_{1,j}^{(2)} &= (1 - \beta v_1^2) a_{1,j} - \beta v_1 v_n a_{n,j} \\ a_{n,j}^{(2)} &= -\beta v_1 v_n a_{1,j} + (1 - \beta v_n^2) a_{n,j} \end{aligned} \tag{2}$$

b) Il metodo per il calcolo del fattore R consiste nel calcolare le matrici

$$A_{k+1} = E_k A_k, \quad k = 1, \ldots, n-1,$$

dove E_k è una matrice elementare di Householder che trasforma in zero gli elementi di indice maggiore di k della k-esima colonna di A_k. Applicando induttivamente la proprietà mostrata al punto a), la sottomatrice principale di coda B_k di dimensione $(n-k) \times (n-k)$ di A_{k+1}, è una matrice con la stessa struttura della matrice A_1. Pertanto le matrici di Householder saranno tutte del tipo $E_k = I - \beta_k v_k v_k^T$ dove v_k ha al più la k-esima e la n-esima componente diverse da zero, e $\beta_k = 2/v_k^T v_k$. La (2) diventa allora

$$\begin{aligned} a_{1,j}^{(k+1)} &= (1 - \beta v_1^2) a_{k,j} - \beta v_1 v_n a_{n,j}^{(k)} \\ a_{n,j}^{(k+1)} &= -\beta v_1 v_n a_{k,j} + (1 - \beta v_n^2) a_{n,j}^{(k)} \end{aligned}$$

per $j = k+1, \ldots, n$. Il suo costo, a meno di costanti additive è di $6(n-k)$ operazioni aritmetiche. Sommando per $k = 1, \ldots, n-1$ si ottiene un costo di $3n^2$ operazioni a meno di termini $O(n)$.

c) La sottomatrice principale di testa di A di dimensione $i \times i$, per $i = 1, \ldots, n-1$, è triangolare superiore e quindi ha determinante $\prod_{j=1}^{i} a_{i,i}$. Quindi se gli $a_{i,i}$ sono non nulli per $i = 1, \ldots, n-1$ allora sono verificate le ipotesi del teorema di esistenza e unicità della fattorizzazione LU. L'algoritmo per il calcolo di U consiste nel calcolare le matrici

$$A_{k+1} = E_k A_k, \quad k = 1, \ldots, n-1,$$

dove E_k è la matrice elementare di Gauss che trasforma in zero gli elementi di indice maggiore di k della k-esima colonna di A_k. Applicando induttivamente la proprietà mostrata al punto a), la sottomatrice principale di coda B_k di dimensione $(n-k) \times (n-k)$ di A_{k+1}, è una matrice con la stessa struttura della matrice A_1. Pertanto le matrici di Gauss saranno tutte del tipo $E_k = I - u_k e_k^T$ dove u_k ha solamente la n-esima componente diversa da zero. La (1) diventa allora

$$a_{n,j}^{(k+1)} = a_{n,j}^{(k)} - \frac{a_{n,1}^{(k)}}{a_{1,1}} a_{1,j}, \quad j = k+1, \ldots, n.$$

Il suo costo, a meno di costanti additive, è di $2(n-k)$ operazioni aritmetiche che sommate per $k = 1, \ldots, n-1$ dà un costo di n^2 operazioni a meno di termini $O(n)$.

Esercizio 3

Soluzione 1 (caso di `p` vettore riga).

```
function q = pol(p)
  q = [1, 3, p] - [p, 0, 0];
endfunction
```

Nel caso di vettore colonna basta sostituire la virgola con un punto e virgola. Usando il comando `if` è possibile scrivere una function che controlla i due casi. Si osservi che la function funziona anche nel caso in cui `p` ha lunghezza 1.

Soluzione 2. Una soluzione in cui non occorre distinguere fra vettore riga o colonna è la seguente

```
function q = pol(p)
  n = length(p);
  q = p*0;
  q(1) = 1;
  q(2) = 3;
  q(3:n+2) = p;
  q(1:n) = q(1:n) - p;
endfunction
```