Calcolo Numerico, corso M-Z
Esempi di Prove di Laboratorio

Istruzioni:

(1) il tempo a disposizione per completare la prova è 1 ora;

(2) lo svolgimento della prova deve essere salvato in file denominati `cognomenome1.m`, `cognomenome2.m`, etc.;

(3) è fatto assoluto divieto di uscire dall'ambiente Matlab, pena l'immediata esclusione dalla prova;

(4) ai fini della valutazione contribuiscono correttezza ed efficienza del codice.

Esempi di quesiti:

Quesito 1. Si scriva una funzione Matlab che implementi il metodo delle secanti per il calcolo degli zeri di funzione:

$$x_k = x_{k-1} - \frac{f(x_{k-1})(x_{k-1} - x_{k-2})}{f(x_{k-1}) - f(x_{k-2})}\,.$$

La funzione deve ammettere come dati di input f, punti iniziali x_0 e x_1, tolleranza e numero massimo di iterate; come dati di output l'approssimazione dello zero ed il numero di iterate effettuate. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo.

Si usi la funzione creata per calcolare le prime 10 cifre significative dello zero di $f(x) = \dfrac{\sin(x)}{x} - x$; si scelga $x_0 = 1$ e $x_1 = 0.9$. [La soluzione è $0.87\ldots$.]

Quesito 2. Si scriva una funzione Matlab che implementi il seguente metodo numerico per il calcolo degli zeri di una funzione, detto di *Halley*:

$$x_{k+1} = x_k - \frac{2f(x_k)f'(x_k)}{2\,[f'(x_k)]^2 - f(x_k)f''(x_k)}.$$

La funzione deve ammettere come dati di input f, f', f'', punto iniziale x_0, tolleranza e numero massimo di iterate; come dati di output l'approssimazione dello zero, il numero di iterate effettuate ed il residuo $f(x_{\texttt{end}})$. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo.

Si usi la funzione creata per calcolare le prime 12 cifre significative dell'unico zero strettamente positivo di $f(x) = \sin(x) - x^3$. [La soluzione è $0.92\ldots$]

Quesito 3. Si scriva una funzione Matlab che implementi il seguente metodo numerico *a due passi* (detto *metodo di Traub*) per il calcolo degli zeri di funzione. Dato x_k, per ogni $k \geq 0$, si calcola:

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)},$$
$$x_{k+1} = y_k - \frac{f(y_k)}{f'(x_k)}.$$

La funzione deve ammettere come dati di input f, f', punto iniziale x_0, tolleranza e numero massimo di iterate; come dati di output l'approssimazione dello zero ed il numero di iterate effettuate. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo. Al fine di ottimizzare il numero di valutazioni di funzione, si tenga conto del fatto che ogni passo richiede una sola valutazione di f'.

Si testi la funzione calcolando le prime 10 cifre significative dello zero di $f(x) = 1 - \dfrac{x^2}{2} - x^3$ nell'intervallo [0 1]. [La soluzione è $0.85\ldots$.]

Quesito 4. Si scriva una funzione Matlab che implementi il seguente metodo numerico per il calcolo della radice n-esima di un numero $S > 0$:

$$x_0 = S, \;\; x_{k+1} = \frac{1}{n}\left((n-1)\,x_k + \frac{S}{x_k^{(n-1)}}\right).$$

La funzione deve ammettere come dati di input S, n, tolleranza e numero massimo di iterate; come dati di output l'approssimazione di $\sqrt[n]{S}$ ed il numero di iterate effettuate. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo.

Si testi la funzione calcolando le prime 10 cifre significative di $\sqrt[5]{286797}$. [La soluzione è $\sqrt[5]{286797} = 12.3\ldots$.]

Quesito 5. Si scriva una funzione Matlab che implementi il seguente metodo numerico per il calcolo degli zeri di una funzione, detto di *Steffensen*:

$$x_{k+1} = x_k - \frac{[f(x_k)]^2}{f\left(x_k + f(x_k)\right) - f(x_k)}.$$

La function deve ammettere come dati di input f, punto iniziale x_0, tolleranza e numero massimo di iterate; come dati di output l'approssimazione dello zero, il numero di iterate effettuate ed il residuo $f(x_{\texttt{end}})$. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo.

Si usi la function creata per calcolare le prime 8 cifre significative dello zero di $f(x) = \sin(x) - \log(x)$. [La soluzione è $2.21\ldots$]

Quesito 6. Si scriva una funzione Matlab che implementi il seguente metodo, detto *quasi-Newton*, per il calcolo degli zeri di funzione:

$$x_k = x_{k-1} - h\frac{f(x_{k-1})}{f(x_{k-1}+h) - f(x_{k-1})}\,, k \geq 1$$

La funzione deve ammettere come dati di input f, punto iniziale x_0, incremento $h > 0$, tolleranza e numero massimo di iterate; come dati di output l'approssimazione dello zero, numero di iterate effettuate e residuo $f(x_{\texttt{end}})$. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo.

Si usi la funzione creata per calcolare le prime 8 cifre significative dello zero di $f(x) = e^x - 1 - \cos(x)$; si scelga $x_0 = 0.5$ e $h = 10^{-6}$. [La soluzione è $0.601\ldots$.]

Quesito 7. Dato il sistema lineare $Ax = b$, $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, si consideri il seguente metodo iterativo per la soluzione dei sistemi lineari (detto metodo di *Gauss-Seidel*):

$$Lx_{k+1} = b - Ux_k\,,$$

dove

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \; L = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \; U = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

[Si noti che $A = L + U$.] Si scriva una funzione Matlab che implementi tale metodo. La funzione deve ammettere come dati di input una matrice A, un vettore b, una tolleranza, un vettore iniziale x_0 ed il numero massimo di iterate; come dati di output l'approssimazione della soluzione, il numero di iterate effettuate ed il residuo $\|Ax_{\texttt{end}} - b\|_\infty$. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo in norma $\|\cdot\|_\infty$. Si suggerisce di risolvere i sistemi lineari facendo uso dell'operatore *backslash* (`x=A\b` è la soluzione del sistema lineare $Ax = b$), e di calcolare la norma di un vettore (colonna) facendo uso della funzione `norm`.

Si testi la funzione approssimando, con tolleranza `tol`$=10^{-5}$, la soluzione del sistema lineare $Ax = b$ con:

$$A = \begin{bmatrix} 2.4 & -0.8 & -0.7 \\ 0.5 & 1.5 & 0.7 \\ -0.1 & 0.8 & 2.1 \end{bmatrix}, \; b = \begin{bmatrix} 0.9 \\ 2.7 \\ 2.8 \end{bmatrix}.$$

[La soluzione esatta è $[1\ 1\ 1]^T$. Si suggerisce di scegliere $x_0 = [0.9\ 0.9\ 0.9]^T$.]

Quesito 8. Dato il sistema lineare $Ax = b$, con $A \in \mathbb{R}^{n\times n}$ e $b \in \mathbb{R}^n$, si consideri il seguente metodo iterativo per la soluzione dei sistemi lineari (detto *metodo di Jacobi*):

$$Dx_{k+1} = b - Rx_k \,,$$

dove

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} , \; D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} , \; R = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix} .$$

Si noti che $A = D + R$. Si scriva una funzione Matlab che implementi tale metodo. La funzione deve ammettere come dati di input una matrice A, un vettore b, una tolleranza, un vettore iniziale x_0 ed il numero massimo di iterate; come dati di output l'approssimazione della soluzione, il numero di iterate effettuate ed il residuo $\|Ax_{\texttt{end}} - b\|_\infty$. Si faccia uso di un criterio di arresto basato sul controllo dell'errore relativo in norma $\|\cdot\|_\infty$. Si suggerisce di risolvere i sistemi lineari facendo uso dell'operatore *backslash* (`x=A\b` è la soluzione del sistema lineare $Ax = b$), e di calcolare la norma di un vettore (colonna) facendo uso della funzione `norm`.

Si testi la function approssimando, con tolleranza 10^{-5}, la soluzione del sistema lineare $Ax = b$ con:

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{bmatrix} , \; b = \begin{bmatrix} 7 \\ 8 \\ 7 \end{bmatrix} .$$

[La soluzione esatta è $[1\ 1\ 1]^T$. Si suggerisce di scegliere $x_0 = [1.1\ 1.1\ 1.1]^T$.]

Quesito 9. Si scriva una funzione Matlab che implementi il seguente metodo numerico (detto *dei trapezi*) per il calcolo approssimato degli integrali definiti. Dati una funzione a valori reali f definita nell'intervallo $[a, b]$, $n \geq 1$ intero naturale, siano $h = (b - a)/n$ e $x_j = a + jh$, per ogni $j = 0, \ldots, n$. Allora si ha

$$\int_a^b f(x)\,dx \approx I_n \,,$$

dove

$$I_n = \frac{h}{2}\left[f(x_0) + 2\sum_{j=1}^{n-1} f(x_j) + f(x_n) \right] .$$

La funzione deve ammettere come dati di input funzione f, intervallo $[a, b]$ e numero di sottointervalli n; come dati di output l'approssimazione I_n.

Utilizzando la funzione appena creata, implementare la seguente procedura automatica per approssimare il valore di $\displaystyle\int_0^1 \sin(\pi x)\,dx$ a meno di un errore assoluto inferiore a 10^{-3}: raddoppiare progressivamente il valore di n, a partire da $n = 4$, fino a quando la differenza tra approssimazioni successive I_n e I_{n+1} sarà scesa sotto la tolleranza indicata. Il valore esatto dell'integrale è $2/\pi$.

Quesito 10. Si scriva una funzione Matlab che implementi il seguente metodo numerico (detto *di Simpson*) per il calcolo degli integrali definiti. Dati una funzione a valori reali f definita l'intervallo $[a, b]$, n intero naturale pari, siano $h = (b - a)/n$ e $x_j = a + jh$, per ogni $j = 0, \ldots, n$. Allora si ha

$$\int_a^b f(x)\,dx \approx I_n \,,$$

dove

$$I_n = \frac{h}{3}\left[f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n) \right] .$$

La funzione deve ammettere come dati di input funzione f, intervallo $[a, b]$ e numero di sottointervalli n; come dati di output l'approssimazione I_n.

Utilizzando la funzione appena creata, implementare la seguente procedura automatica per calcolare le prime 4 cifre significative di $\displaystyle\int_0^{\pi/4} \tan(x)\,dx$: raddoppiare progressivamente il valore di n, a partire da $n = 4$, fino a quando la differenza tra approssimazioni successive I_n e I_{n+1} sarà scesa sotto la tolleranza indicata. Il valore esatto dell'integrale è $\log(2)/2$.

Quesito 11. Si scriva una funzione Matlab che implementi il seguente metodo, detto *metodo delle potenze*, per il calcolo dell'autovalore di massimo modulo di una matrice quadrata. Dati una matrice $A \in \mathbb{R}^{n \times n}$ ed un vettore $x_0 \in \mathbb{R}^n$, per ogni $k \geq 0$ si calcola:

$$y_k = \frac{x_k}{\|x_k\|_2}, \ x_{k+1} = A\, y_k, \ \nu_{k+1} = y_k^T x_{k+1}\,.$$

Al termine della procedura, l'approssimazione dell'autovalore di massimo modulo sarà $\lambda_{\texttt{max}} = \nu_{\texttt{end}}$, dove $\nu_{\texttt{end}}$ è l'ultimo valore di ν_k calcolato. La funzione deve ammettere come dati di input una matrice quadrata A, un vettore iniziale x_0 di dimensioni compatibili con A, tolleranza e numero massimo di iterate; come dati di output $\lambda_{\texttt{max}}$ ed il numero di iterate effettuate. Per ciò che riguarda il controllo dell'errore, si faccia uso del seguente criterio:

$$|\nu_{k+1} - \nu_k| < \texttt{tolleranza}\,|\nu_{k+1}|.$$

Si noti che sarà necessario effettuare almeno due iterate per poter implementare tale controllo. Per calcolare le norme vettoriali, si può utilizzare la funzione `norm`.

Si testi la funzione approssimando, con tolleranza 10^{-6}, l'autovalore di massimo modulo di

$$A = \begin{bmatrix} 13 & 2 & 0 \\ 2 & 1 & 3 \\ 0 & 3 & 22 \end{bmatrix}.$$

Si scelga

$$x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

[La soluzione esatta è 22.4 . . . .]

Quesito 12. Si scriva una funzione Matlab che implementi il seguente metodo, detto *metodo delle potenze inverse*, per il calcolo dell'autovalore di minimo modulo di una matrice quadrata. Dati una matrice $A \in \mathbb{R}^{n \times n}$ ed un vettore $x_0 \in \mathbb{R}^n$, per ogni $k \geq 0$:

$$\text{si pone} \quad y_k = \frac{x_k}{\|x_k\|_2},$$
$$\text{si risolve il sistema lineare} \quad Ax_{k+1} = y_k,$$
$$\text{si pone} \quad \nu_{k+1} = y_k^T x_{k+1}\,.$$

Al termine della procedura, l'approssimazione dell'autovalore di minimo modulo sarà $\lambda_{\texttt{min}} = 1/\nu_{\texttt{end}}$, dove $\nu_{\texttt{end}}$ è l'ultimo valore di ν_k calcolato. La funzione deve ammettere come dati di input una matrice quadrata A, un vettore iniziale x_0 di dimensioni compatibili con A, tolleranza e numero massimo di iterate; come dati di output $\lambda_{\texttt{min}}$ ed il numero di iterate effettuate. Per ciò che riguarda il controllo dell'errore, si faccia uso del seguente criterio:

$$|\nu_{k+1} - \nu_k| < \texttt{tolleranza}\,|\nu_{k+1}|.$$

Si noti che sarà necessario effettuare almeno due iterate per poter implementare tale controllo. Si suggerisce di risolvere i sistemi lineari facendo uso dell'operatore *backslash* (`x=A\b` è la soluzione del sistema lineare $Ax = b$), e di calcolare la norma di un vettore (colonna) facendo uso della funzione `norm`.

Si testi la funzione approssimando, con tolleranza 10^{-6}, l'autovalore di minimo modulo di

$$A = \begin{bmatrix} 13 & 2 & 0 \\ 2 & 1 & 3 \\ 0 & 3 & 22 \end{bmatrix}.$$

Si scelga

$$x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

[La soluzione esatta è 0.27 . . . .]

Quesito 13. Si scriva una funzione Matlab che implementi il metodo di *sostituzione in avanti* per la risoluzione dei sistemi lineari con matrice dei coefficienti triangolare inferiore. Dati

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

la soluzione del sistema lineare $Ax = b$ è calcolata come segue:

$$x_1 = \frac{b_1}{a_{11}},$$
$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right) \quad \text{per } i = 2, \ldots, n.$$

La funzione deve ammettere come dati di input la matrice A triangolare inferiore ed il vettore b; come dati di output la soluzione x del sistema $Ax = b$.

Si usi la funzione creata per calcolare la soluzione del sistema lineare con i seguenti dati:

$$A = \begin{bmatrix} 16 & 0 & 0 & 0 \\ 5 & 11 & 0 & 0 \\ 9 & 7 & 6 & 0 \\ 4 & 14 & 15 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 16 \\ 16 \\ 22 \\ 34 \end{bmatrix}.$$

Quesito 14. Si scriva una funzione Matlab che implementi il metodo di *sostituzione all'indietro* per la risoluzione dei sistemi lineari con matrice dei coefficienti triangolare superiore. Dati

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

la soluzione del sistema lineare $Ax = b$ è calcolata come segue:

$$x_n = \frac{b_n}{a_{nn}},$$
$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) \quad \text{per } i = n-1, n-2, \ldots, 1.$$

La funzione deve ammettere come dati di input la matrice A triangolare superiore ed il vettore b; come dati di output la soluzione x del sistema $Ax = b$.

Si usi la funzione creata per calcolare la soluzione del sistema lineare con i seguenti dati:

$$A = \begin{bmatrix} 16 & 2 & 3 & 13 \\ 0 & 11 & 10 & 8 \\ 0 & 0 & 6 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 34 \\ 29 \\ 18 \\ 1 \end{bmatrix}.$$