Esame di Calcolo Numerico — 27 giugno 2022

Corso di Laurea in Ingegneria Chimica
Tempo a disposizione: 2 ore. È consentito consultare appunti e testi (cartacei).

Esercizio 1 (15 punti)

1. Scrivere una `function [L1, B] = lu1(A)` che prende in input una matrice quadrata $A \in \mathbb{R}^{n\times n}$ ed esegue il primo passo della fattorizzazione LU (senza pivoting): restituisce una matrice elementare di Gauss $L_1 \in \mathbb{R}^{n\times n}$ e una matrice $B \in \mathbb{R}^{n\times n}$ con $B_{21} = B_{31} = \cdots = B_{n1} = 0$ tali che $L_1 B = A$. Riportare sul foglio il codice della funzione scritta.

2. Riportare le matrici L_1, B calcolate eseguendo la funzione sulla matrice $A_3 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$. Verificate (con carta e penna) che effettivamente $(L_1 B)_{33} = A_{33}$. Quali operazioni è necessario fare?

3. Consideriamo la matrice $M = 0.1 A_3$, ottenuta moltiplicando per 0.1 tutti gli elementi di A_3. La matrice M è invertibile? Esiste la sua fattorizzazione LU (senza pivoting)?

Esercizio 2 (15 punti) Consideriamo il problema ai valori iniziali

$$y' = f(t, y), \quad y(1) = 1, \quad [a, b] = [1, 2]. \tag{1}$$

Vogliamo risolverlo utilizzando il metodo di Runge–Kutta

$$y_{n+1} = y_n + \frac{5}{6} h f(t_n, y_n) + \frac{1}{6} h f\left(t_n + \frac{1}{3}h, y_n + \frac{1}{3} h f(t_n, y_n)\right). \tag{2}$$

1. Qual è la tavola di Butcher di questo metodo?

2. Scrivere una `function [t, Y] = rk2(f, N)` che applica il metodo (2) al problema (1). Riportare sul foglio il codice della funzione. Se riuscite, cercare di scrivere questo codice in modo che la funzione `f` venga richiamata due volte per ogni passo del metodo anziché tre o più.

3. Per il problema $y' = -0.1y$ con $N \in \{50, 100, 200\}$, riportare l'errore globale massimo $e_N = \max_{n=1,\dots,N} |y_n - y(t_n)|$ tra la soluzione numerica calcolata con il metodo (2) e quella esatta $\exp(-0.1(t-1))$. Cosa indicano i valori ottenuti sull'ordine di convergenza di questo metodo?

Soluzioni

Esercizio 1 (15 punti)

1. Una possibile soluzione è la seguente.

```
function [L1, B] = lu1(A)
n = size(A,1);
L1 = eye(n);
B = A;
for i = 2:n
    L1(i, 1) = A(i,1) / A(1,1);
    B(i, 1) = 0;
    B(i, 2:end) = B(i, 2:end) - L1(i,1)*B(1, 2:end);
end
```

```
>> A = [1 2 3; 4 5 6; 7 8 9];
>> [L1, B] = lu1(A)
L1 =
     1     0     0
     4     1     0
     7     0     1
B =
     1     2     3
     0    -3    -6
     0    -6   -12
```

Per verificare che $(L_1B)_{33} = A_{33}$, serve calcolare

$$(L_1)_{31}B_{13} + (L_1)_{32}B_{23} + (L_1)_{33}B_{33} = 7 \cdot 3 - 0 \cdot 6 + 1 \cdot (-12) = 21 + 0 - 12 = 9 = A_{33}.$$

3. Le sottomatrici principali di testa M_1 e M_2 di M sono invertibili, visto che hanno determinante 0.1 e $0.05 - 0.08 = -0.03$ rispettivamente; quindi M ha una fattorizzazione LU. La matrice M però è singolare, come è possibile verificare in diversi modi: per esempio notando che $\det M = \det(0.1A_3) = (0.1)^3 \det A_3$ e calcolando $\det A_3$, o notando che la seconda colonna è la media aritmetica della prima e della terza, quindi $M \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = 0$. Inserendo la matrice M direttamente in Matlab e calcolandone il determinante si ottiene `4.7343e-18`, ma il fatto che questo numero sia diverso da zero è dovuto agli errori dell'aritmetica di macchina.

Esercizio 2 (15 punti)

1. Ponendo $k_1 = f(t_n, y_n)$ e $k_2 = f\left(t_n + \frac{1}{3}h, y_n + h\frac{1}{3}k_1\right)$, vediamo che (2) corrisponde alla tavola di Butcher

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 \\ \hline & \frac{5}{6} & \frac{1}{6} \end{array}.$$

2. Una soluzione possibile è la seguente.

```
function [t, Y] = rk2(f, N)
a = 1;
b = 2;
h = (b-a)/N;
t = a:h:b;
```

```
Y = zeros(1, N+1);
Y(1) = 1;
for n = 1:N
    k1 = f(t(n), Y(n));
    k2 = f(t(n) + h/3, Y(n) + h/3*k1);
    Y(n+1) = Y(n) + h*5/6*k1 + h*1/6*k2;
end

>> [t,Y] = rk2(@(t,y) -0.1*y, 50);
>> e50 = max(abs(Y - exp(-0.1*(t-1))))

e50 =

   8.0527e-05

>> [t,Y] = rk2(@(t,y) -0.1*y, 100);
>> e100 = max(abs(Y - exp(-0.1*(t-1))))

e100 =

   4.0239e-05

>> [t,Y] = rk2(@(t,y) -0.1*y, 200);
>> e200 = max(abs(Y - exp(-0.1*(t-1))))

e200 =

   2.0114e-05
```

Gli errori si dimezzano al raddoppiare di N; questo indica un metodo con ordine di convergenza 1.