

Esame di Calcolo Numerico — 24 Gennaio 2022

Corso di Laurea in Ingegneria Chimica

Tempo a disposizione: 2 ore. È consentito consultare appunti e testi (cartacei).

Esercizio 1 (15 punti) Consideriamo la matrice $A \in \mathbb{R}^{n \times n}$ che ha elementi

$$A_{ij} = \begin{cases} 1 & j = 1 \text{ oppure } i = j, \\ 0 & \text{altrimenti,} \end{cases} \quad (1)$$

ad esempio per $n = 4$ si ha

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

Vogliamo approssimare la soluzione di un sistema lineare $Ax = b$ con questa matrice A utilizzando il metodo di Jacobi; sia $x^{(k)}$ la successione generata dal metodo.

1. Scrivere le equazioni che permettono di calcolare gli elementi di $x^{(k+1)}$ a partire da quelli di $x^{(k)}$, sfruttando la forma particolare della matrice (1).
2. Scrivere una `function xkpiu1 = jacobiA(b, xk)` che, presi in ingresso vettori $b, x^{(k)} \in \mathbb{R}^n$, esegue un passo del metodo (dopo aver controllato che i due vettori abbiano la stessa lunghezza). Usare le formule calcolate al punto precedente direttamente, senza costruire le matrici A , N o H all'interno della funzione.
3. Calcolare (per un generico n) la matrice di iterazione H del metodo, e dire perché si ha che $H^2 = 0$. Cosa è possibile concludere sull'errore $e^{(2)}$ ottenuto dopo due passi?
4. Eseguendo la funzione precedente più volte a partire da $b = [1, 2, 3, 4, 5]^T$ e $x^{(0)} = \text{zeros}(5, 1)$, calcolare le iterate $x^{(1)}, x^{(2)}, x^{(3)}$ generate dal metodo. I valori di $x^{(2)}$ e $x^{(3)}$ corrispondono a quelli attesi in vista del punto precedente?

Esercizio 2 (15 punti) Data una funzione $\alpha : \mathbb{R} \rightarrow \mathbb{R}$, consideriamo il problema ai valori iniziali

$$y' = \alpha(t)y, \quad y(0) = y_0 = 1, \quad [a, b] = [0, 1]. \quad (2)$$

Vogliamo approssimare la soluzione di questo problema tramite il metodo di Runge–Kutta con tavola di Butcher

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & 0 \\ \hline \frac{3}{4} & \frac{1}{4} & \frac{3}{4} \end{array}$$

1. Scrivere una `function [t, Y] = rk2(alpha, N)` che applica questo metodo al problema (2), ricevendo in ingresso il numero di passi N e una `function handle` che calcola $\alpha(t)$. (Fare uso all'interno del codice della forma della (2) e della tavola di Butcher data: non è consentito utilizzare direttamente una funzione per risolvere problemi più generali.) Riportare sul foglio il codice della funzione.
2. Per $\alpha(t) = -2t$ e $N \in \{50, 100, 200\}$, riportare l'errore globale massimo $\max_{n=1, \dots, N} |y_n - y(t_n)|$ tra la soluzione numerica e quella esatta (quest'ultima si può calcolare tramite Matlab con `exp(-t.^2)`). Cosa indicano i valori ottenuti sull'ordine di convergenza del metodo?
3. Calcolare la funzione di stabilità $R(q)$ del metodo. Come mai deve esistere un valore di $q \in \mathbb{C}$ con $\text{Re } q < 0$ tale che $|R(q)| \geq 1$? Sapreste trovarne uno esplicitamente?

Soluzioni

Esercizio 1 (15 punti)

1. Sostituendo gli elementi della (1) all'interno delle formule per il metodo di Jacobi, otteniamo

$$\begin{aligned}x_1^{(k+1)} &= b_1, \\x_2^{(k+1)} &= b_2 - x_1^{(k)}, \\x_3^{(k+1)} &= b_3 - x_1^{(k)}, \\&\vdots \\x_n^{(k+1)} &= b_n - x_1^{(k)},\end{aligned}$$

che possiamo anche scrivere come

$$\begin{aligned}x_1^{(k+1)} &= b_1, \\x_i^{(k+1)} &= b_i - x_1^{(k)},\end{aligned}\quad \text{per } i > 1.$$

2. Una possibile implementazione è la seguente.

```
function xkpiu1 = jacobiA(b, xk)
n = length(b);
if not(length(xk)==n)
    error('I vettori devono avere la stessa lunghezza');
end
xkpiu1 = zeros(n,1);
xkpiu1(1) = b(1);
for i = 2:n
    xkpiu1(i) = b(i) - xk(1);
end
```

3. Si ha $M = I$, e

$$H = M^{-1}N = N = \begin{bmatrix} 0 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix},$$

la matrice che ha $H_{ij} = -1$ sugli elementi della prima colonna a partire dal secondo e 0 in tutte le altre posizioni. Calcolando il prodotto H^2 , gli elementi della prima colonna vengono moltiplicati per elementi della prima riga di H che sono tutti uguali a 0, quindi si ha $H^2 = 0$. Visto che $e^{(2)} = H^2 e^{(0)} = 0$, il metodo converge alla soluzione esatta in (al più) due passi.

4.

```
>> b = [1,2,3,4,5]; x0 = zeros(5, 1);
>> x1 = jacobiA(b, x0)
x1 =
     1
     2
     3
     4
     5
>> x2 = jacobiA(b, x1)
x2 =
```

```

1
1
2
3
4
>> x3 = jacobiA(b, x2)
x3 =
1
1
2
3
4

```

Si ha che $x_2 = x_3$, e il vettore x_2 soddisfa $Ax_2 = b$, quindi effettivamente esso è la soluzione esatta del sistema.

Esercizio 2 (15 punti)

- Una possibile soluzione è la seguente.

```

function [t, Y] = rk2(alpha, N)
a = 0;
b = 1;
h = (b-a)/N;
t = a:h:b;
Y = zeros(1, N+1);
Y(1) = 1;
for n = 1:N
    k1 = alpha(t(n)) * Y(n);
    k2 = alpha(t(n) + 2/3*h) * (Y(n) + 2/3 * h * k1);
    Y(n+1) = Y(n) + h*(1/4 * k1 + 3/4 * k2);
end

```

- Una possibile soluzione è la seguente.

```

>> [t, Y] = rk2(@(t) -2*t, 50);
>> E50 = max(abs(Y-exp(-t.^2)))
E50 =
    2.2440e-05
>> [t, Y] = rk2(@(t) -2*t, 100);
>> E100 = max(abs(Y-exp(-t.^2)))
E100 =
    5.4892e-06
>> [t, Y] = rk2(@(t) -2*t, 200);
>> E200 = max(abs(Y-exp(-t.^2)))
E200 =
    1.3574e-06
>> E50/E100, E100/E200
ans =
    4.0880
ans =
    4.0439

```

3. Per calcolare la funzione di stabilità, applichiamo il metodo al problema test, ottenendo

$$\begin{aligned}k_1 &= \lambda y_n, \\k_2 &= \lambda(y_n + \frac{2}{3}hk_1) = \lambda y_n + \frac{2}{3}\lambda q y_n, \\y_{n+1} &= y_n + \frac{1}{4}hk_1 + \frac{3}{4}hk_2 = y_n + \frac{1}{4}q y_n + \frac{3}{4}q y_n + \frac{3 \cdot 2}{4 \cdot 3}q^2 y_n = (1 + q + \frac{1}{2}q^2)y_n.\end{aligned}$$

La funzione di stabilità quindi è $R(q) = 1 + q + \frac{1}{2}q^2$. Prendendo per esempio $q = -4$ si ha $|R(q)| = 5 \geq 1$. (Un valore di q che soddisfa le richieste deve esistere perché il metodo è esplicito, e quindi non può essere A-stabile.)