Esercitazione 8

Metodi Numerici per la Soluzione di Equazioni Differenziali Ordinarie del Secondo Ordine e Sistemi di Equazioni Differenziali Ordinarie

Sistemi di Equazioni Differenziali Ordinarie del primo ordine e $\boldsymbol{\theta}$-metodo

θ-metodo: caso scalare

Per la risoluzione del problema di Cauchy:

$$\begin{cases} y'(t) = f(t,y) & t \in (t_0, t_f), \\ y(t_0) = y_0, \end{cases} \tag{1}$$

introduciamo il cosiddetto θ-metodo, ossia una famiglia di metodi dipendenti dal parametro $\theta \in [0,1]$.

Si consideri un passo di avanzamento temporale h e una suddivisione dell'intervallo temporale $[t_0, t_f]$ in $N_h = (t_f - t_0)/h$ sottointervalli di egual ampiezza h. Dato un parametro $\theta \in [0,1]$, calcoliamo i valori incogniti $u_0, \ldots, u_{N_h}$ come:

$$\begin{cases} u_{n+1} = u_n + h\,[(1-\theta)\,f(t_n,u_n) + \theta\,f(t_{n+1},u_{n+1})] & \text{per } n = 0,\ldots,N_h-1, \\ u_0 = y_0. \end{cases}$$

Per ogni istante temporale discreto $t_n = t_0 + n\,h$, con $n = 0,\ldots,N_h$, il valore u_n *approssima* $y_n = y(t_n)$. L'insieme dei valori $\{u_0, u_1, \ldots, u_{N_h}\}$ rappresenta la soluzione numerica di (1).

Si tratta di un metodo accurato di ordine $p = 1$ per $\theta \in [0,1/2) \cup (1/2,1]$, mentre di ordine $p = 2$ per $\theta = 1/2$, ovvero quando il metodo coincide con il metodo di Crank-Nicolson. Osserviamo che il caso $\theta = 0$ corrisponde al metodo di Eulero in avanti, mentre $\theta = 1$ abbiamo al metodo di Eulero all'indietro.

θ-metodo: caso vettoriale

Si consideri un'equazione differenziale ordinaria di ordine p:

$$\begin{cases} z^{(p)}(t) = f(t, z(t), z'(t), \ldots, z^{(p-1)}(t)) & t \in (t_0, t_f), \\ z(t_0) = z_0,\ z'(t_0) = z_1,\ \ldots\ , z^{(p-1)}(t_0) = z_{p-1}. \end{cases}$$

Ponendo

$$y_1(t) = z(t),\ y_2(t) = z'(t),\ \ldots\ , y_p(t) = z^{(p-1)}(t),$$

l'equazione differenziale di ordine p può essere ricondotta al seguente sistema di p equazioni lineari di ordine 1:

$$\begin{cases} y_1'(t) = y_2(t) \\ \quad\vdots & t \in (t_0, t_f), \\ y_p'(t) = f(t, y_1, \ldots, y_p) \\ y_1(t_0) = z_0,\ y_2(t_0) = z_1,\ \ldots\ , y_p(t_0) = z_{p-1}. \end{cases}$$

Per l'approssimazione numerica del sistema, si possono applicare i metodi di discretizzazione visti per il caso scalare.

Si consideri dunque il seguente sistema generale di m equazioni differenziali ordinarie nelle incognite $y_1 = y_1(t), \ldots, y_m = y_m(t) \, : \, (t_0, t_f) \to \mathbb{R}$:

$$\begin{cases} y_1'(t) = f_1(t, y_1, \ldots, y_m) \\ \quad \vdots & \qquad t \in (t_0, t_f), \\ y_m'(t) = f_m(t, y_1, \ldots, y_m) \\ y_1(t_0) = y_{0,1}, \; \ldots, \; y_m(t_0) = y_{0,m}. \end{cases}$$

Definendo il vettore $\mathbf{y}(t) = (y_1(t), \ldots, y_m(t))^T \, : \, (t_0, t_f) \to \mathbb{R}^m$, possiamo riscrivere il sistema come:

$$\begin{cases} \mathbf{y}'(t) = \mathbf{F}(t, \mathbf{y}(t)) & t \in (t_0, t_f), \\ \mathbf{y}(t_0) = \mathbf{y}_0, \end{cases} \tag{2}$$

dove $\mathbf{y}_0 = (y_{1,0}, y_{2,0}, \ldots, y_{m,0})^T$ è il dato iniziale e $\mathbf{F}(t, \mathbf{y}) : (t_0, t_f) \times \mathbb{R}^m \to \mathbb{R}^m$ con

$$\mathbf{F}(t, \mathbf{y}) = (f_1(t, y_1, \ldots, y_m), \ldots, f_m(t, y_1, \ldots, y_m))^T.$$

Indicato con $\mathbf{u}_n$ l'approssimazione di $\mathbf{y}(t_n)$, il θ-metodo si scrive come:

$$\begin{cases} \mathbf{u}_{n+1} = \mathbf{u}_n + h \, [(1-\theta) \, \mathbf{F}(t_n, \mathbf{u}_n) + \theta \, \mathbf{F}(t_{n+1}, \mathbf{u}_{n+1})] & \text{per } n = 0, \ldots, N_h - 1, \\ \mathbf{u}_0 = \mathbf{y}_0. \end{cases} \tag{3}$$

Osserviamo che il θ metodo è implicito per $\theta > 0$: è dunque necessario risolvere un sistema di equazioni non lineari del tipo:

$$\mathbf{F}_{\theta,n}(\mathbf{w}) = \mathbf{w} - \mathbf{u}_n - h \, [(1-\theta) \, \mathbf{F}(t_n, \mathbf{u}_n) + \theta \, \mathbf{F}(t_{n+1}, \mathbf{w})] \qquad \text{per } n = 0, \ldots, N_h - 1,$$

nello zero $\mathbf{u}_{n+1}$, ovvero tale che $\mathbf{F}_{\theta,n}(\mathbf{u}_{n+1}) = 0$. Utilizziamo per semplicità il metodo delle *iterazioni di punto fisso* per risolvere il sistema di equazioni non lineari precedenti per ogni $n = 0, \ldots, N_h - 1$; scriviamo allora:

$$\begin{cases} \mathbf{w}^{(0)} = \mathbf{u}_n, \\ \mathbf{w}^{(k+1)} = \boldsymbol{\phi}_{\theta,n}(\mathbf{w}^{(k)}) & \text{per } k = 0, 1, \ldots \text{ fino a criterio d'arresto soddisfatto,} \\ \mathbf{u}_{n+1} = \mathbf{w}^{(k+1)}, \end{cases}$$

dove

$$\boldsymbol{\phi}_{\theta,n}(\mathbf{w}) := \mathbf{u}_n + h \, [(1-\theta) \, \mathbf{F}(t_n, \mathbf{u}_n) + \theta \, \mathbf{F}(t_{n+1}, \mathbf{w})] \qquad \text{per } n = 0, \ldots, N_h - 1,$$

è la funzione di iterazione associata.

Come metodo alternativo, possiamo usare un metodo di Runge-Kutta (oppure multistep), resi disponibili in Matlab® con funzioni dedicate built-in, la cui sintassi è:

```
[t,y] = odeXX(fun,[t0 tf],y0)
```

dove `y0` è il vettore delle condizioni iniziali, `[t0 tf]` è l'intervallo temporale all'interno del quale si vuole risolvere il problema, `fun` è una anonymus function specificata dall'utente e `odeXX` è uno dei metodi disponibili in Matlab® (ad esempio, `ode45` usa metodi di Runge-Kutta adattivi, di ordine 4 e 5, confrontando tra i due metodi per decidere se variare il passo temporale; `ode15s` è un metodo multistep di ordine variabile adatto per problemi *stiff*; si veda la documentazione dei metodi per approfondimenti, ad esempio attraverso il comando `doc ode45`). L'utente deve quindi implementare l'anonymus function `fun` che restituisce la valutazione dei termini di destra del sistema per un dato istante; `fun` deve essere per forza definita con argomenti di input `t` e `y` anche se non dipende esplicitamente dal tempo. Ricordiamo che tutte le funzioni della famiglia `odeXX` usano un passo di discretizzazione temporale h adattivo.

Esempio:

```
F = @( t, y ) [ t - y( 1 ) - 2 * y( 2 ); t * y( 1 ) - y( 2 )];
[t,u] = ode45( F, [0 10], [2 3] );
```

fornisce l'approssimazione della soluzione del sistema di equazioni differenziali ordinarie del primo ordine di dimensione $m = 2$ con $\mathbf{F}(t, \mathbf{y}) = (t - y_1 - 2y_2, t\, y_1 - y_2)^T$, $\mathbf{y} = (2, 3)^T$ e nell'intervallo di tempo $(0, 10)$. L'output della funzione è composto dal vettore `t` degli istanti temporali di discretizzazione e dalla matrice `u` di due (m) colonne, che contiene i valori assunti dalle variabili y_1 e y_2 negli istanti considerati.

Esercizio 1

Si consideri il sistema di equazioni differenziali ordinarie del primo ordine (2) con $m = 2$, $\mathbf{F}(t, \mathbf{y}) = A\mathbf{y}$, dove:

$$A = \begin{bmatrix} 0 & 1 \\ -100 & -101 \end{bmatrix},$$

$t_0 = 0$, $t_f = 5$ e $\mathbf{y}_0 = (1, 1)^{\mathrm{T}}$.

1. Implementare il θ-*metodo*, basato sul metodo delle iterazioni di punto fisso, nella funzione Matlab® di cui si riporta l'intestazione:

   ```
   function [t_h,u_h] = theta_method(fun,t_max,y0,h,theta)
   ```

   Il metodo richiede in input la funzione `fun` che rappresenta $\mathbf{F}(t, \mathbf{y})$ definita come anonymous function (`fun = @(t,u) ...`), l'istante finale `t_max`= t_f dell'intervallo temporale di soluzione (l'istante iniziale è sempre $t_0 = 0$), i dati iniziali del problema di Cauchy `y0`= $\mathbf{y}_0$, il passo `h` e il parametro `theta`. Il metodo restituisce in output il vettore `t_h` degli istanti temporali e la matrice `u_h` contenente la soluzione calcolata numericamente.

2. Risolvere il problema con il metodo di Eulero in avanti ($\theta = 0$) con un passo di discretizzazione temporale molto piccolo, ad esempio $h = 10^{-4}$, e rappresentare su un grafico la soluzione numerica ottenuta.

3. Quale condizione deve soddisfare il passo di discretizzazione temporale affinché il metodo di Eulero in avanti sia assolutamente stabile per il sistema considerato? Verificare sperimentalmente.

   Suggerimento: la soluzione esatta del generico problema $\mathbf{y}' = A\mathbf{y}$ con $A \in \mathbb{R}^{n \times n}$ si può scrivere nella seguente forma:

   $$\mathbf{y}(t) = \sum_{j=1}^{n} C_j \mathrm{e}^{\lambda_j t} \mathbf{v}_j \tag{4}$$

   dove $\{\lambda_j, \mathbf{v}_j\}$ sono le coppie autovalore-autovettore della matrice A e le C_j sono n costanti da determinare imponendo le condizioni iniziali.

4. Si risolva il sistema con il metodo `ode45` e si rappresenti la soluzione numerica ottenuta; quanti passi temporali sono stati effettuati? Si rappresenti il valore del passo h utilizzato dal metodo al variare dell'istante t_n (*suggerimento*: utilizzare il comando `diff`); il limite imposto dalla condizione di assoluta stabilità per il metodo di Eulero in avanti è stato superato?

5. Si ripetano i punti 2 e 3 utilizzando il θ-metodo per $\theta = \dfrac{1}{2}$ (Crank-Nicolson).

Esercizio 2

Consideriamo un modello epidemiologico compartimentale di tipo SEIR per lo studio della dinamica di una malattia infettiva in un gruppo di individui (popolazione). Il modello SEIR considera in una popolazione di N individui i seguenti compartimenti: $S(t)$, il numero di individui suscettibili al tempo t; $I(t)$, il numero di individui infettivi; $E(t)$, il numero di individui esposti (per tener conto del periodo di incubazione per individui infetti, ma non ancora infettivi) e $R(t)$ il numero di recuperati o rimossi, ovvero individui che non rientrano più nella dinamica dell'epidemia (guariti, immuni, isolati, deceduti, etc...). Il modello SEIR corrisponde a un sistema di EDO del primo ordine nella forma seguente:

$$\begin{cases} \dfrac{dS}{dt} = -\beta \dfrac{S\,I}{N} & t \in (t_0, t_f), \\ \dfrac{dE}{dt} = +\beta \dfrac{S\,I}{N} - \alpha\, E & t \in (t_0, t_f), \\ \dfrac{dI}{dt} = +\alpha\, E - \gamma\, I & t \in (t_0, t_f), \\ \dfrac{dR}{dt} = +\gamma\, I & t \in (t_0, t_f), \\ S(t_0) = S_0,\ E(t_0) = E_0,\ I(t_0) = I_0,\ R(t_0) = R_0, \end{cases}$$

dove S_0, E_0, I_0 e R_0 sono i valori iniziali, β rappresenta il tasso di infezione o contagio, α il tasso di incubazione (α^{-1} rappresenta il tempo di incubazione medio), γ il tasso di rimozione (γ^{-1} rappresenta il tempo infettivo medio). Il valore del numero di riproduzione di base è $r_0 = \dfrac{\beta}{\gamma}$, il numero di infetti secondari che ogni infezione produce. Osserviamo che, essendo $S(t) + E(t) + I(t) + R(t) = N$ per ogni $t \in [t_0, t_f]$, il precedente sistema di 4 equazioni può essere ridotto a un sistema di EDO con $m = 3$ equazioni in S, E ed I, da cui poi ricavare R.

1. Si riscriva il modello SEIR come un sistema di $m = 3$ EDO del primo ordine. Posti $N = 10^7$, $E_0 = 100$, $I_0 = 30$, $R_0 = 0$, $\beta = 0.7586$ giorni^{-1}, $\alpha^{-1} = 5.2$ giorni, $\gamma^{-1} = 2.9$ giorni, $t_0 = 0$ e $t_f = 300$ giorni, si risolva il problema tramite il metodo di Eulero in avanti con passo $h = 0.25$ giorni usando la funzione `theta_method.m`. Che percentuale della popolazione ($100\, R_\infty / N$) rientra nel gruppo dei rimossi al tempo $t = t_f$?

2. Si risolva ora il problema assumendo che il valore β passi da 0.7586 a 0.25 giorni^{-1} a partire dal giorno $t = 80$ a seguito dell'attuazione di una politica di isolamento della popolazione. Che percentuale della popolazione rientra nel gruppo dei rimossi al tempo $t = t_f$? Cosa succede se il valore di β risale a 0.65 giorni^{-1} al giorno $t = 120$?

Esercizio 3

Si consideri il problema semplificato del moto di un satellite attorno alla Terra. La posizione della Terra è centrata nell'origine degli assi $\mathbf{x}_0 = (0, 0)^T$, mentre la posizione del satellite è data dal vettore posizione $\mathbf{r}(t) = (x(t), y(t))^T$; applicando la legge di gravitazione universale, il moto del satellite è dato dall'equazione differenziale vettoriale del secondo ordine

$$\mathbf{r}'' = -\frac{GM}{|\mathbf{r}|^2} \frac{\mathbf{r}}{|\mathbf{r}|} \tag{5}$$

che, dopo aver esplicitato le componenti del vettore $\mathbf{r}$, diventa:

$$\begin{cases} x'' = -\dfrac{GM}{(x^2 + y^2)^{3/2}} x, \\ \\ y'' = -\dfrac{GM}{(x^2 + y^2)^{3/2}} y. \end{cases} \tag{6}$$

dove $GM = 398600$ Km s^{-2} è il valore della costante di gravitazione universale moltiplicata per la massa della Terra.

1. Si scriva il sistema (6) come un sistema di $m = 4$ equazioni differenziali del primo ordine.

2. Si prendano le seguenti condizioni iniziali:

$$\begin{cases} x(0) & = 405500 \text{ Km}, \\ y(0) & = 0, \\ x'(0) & = 0, \\ y'(0) & = 0.964 \text{ Km s}^{-1}, \end{cases}$$

   che corrispondono ai valori dell'orbita della Luna all'apogeo. Si risolva il sistema utilizzando la funzione Matlab® `ode45` nell'intervallo di tempo $t \in (0, 1 \text{ anno})$ (*n.b.*: si converta 1 anno in secondi). L'orbita dovrebbe essere chiusa, questo si verifica anche numericamente?

3. Per ovviare al problema insorto precedentemente, modificare la tolleranza sull'errore relativo definendo le seguenti opzioni:
   `options = odeset('reltol',1e-6)`
   che vanno passate come ultimo argomento di input alla `function ode45`:
   `[T,Y] = ode45(fun,[t0 tmax],Y0,options)`
   L'orbita della Luna adesso risulta chiusa? Quale è la distanza minima dalla Terra nella sua orbita?

4. Si consideri ora la dinamica con attrito data dall'equazione:

$$\mathbf{r}'' = -\frac{GM}{|\mathbf{r}|^2}\frac{\mathbf{r}}{|\mathbf{r}|} - C\mathbf{r}'. \tag{7}$$

   Si prenda $C = 10^{-6}$ e si studi l'equazione sull'intervallo di tempo $t \in (0, 30 \text{ giorni})$ partendo dalle stesse condizioni iniziali del punto precedente. Dopo quanti giorni la Luna impatta sulla Terra (*n.b.:* si tenga conto del raggio terrestre = 6360 Km e del raggio lunare = 1737 Km)? Con che velocità?