8

Modelli matematici ed equazioni differenziali ordinarie

8.1 Alcuni esempi

Molti problemi del mondo fisico sono governati da modelli matematici descritti mediante sistemi di equazioni differenziali. Basti pensare a sistemi meccanici come sistemi di punti materiali, o, più in generale, a sistemi descritti da un numero finito di coordinate, quali il pendolo o un corpo rigido con un punto fisso.

Nella teoria dei circuiti elettrici, l'andamento delle correnti e delle tensioni ai capi di ciascun componente del circuito è descritto da un sistema di equazioni differenziali ordinarie. Nel caso il circuito sia di tipo lineare esistono diverse tecniche, come quelle basate sulle trasformate di Fourier e di Laplace, che consentono una adeguata trattazione analitica del sistema. Nel caso in cui il circuito contenga elementi non lineari, allora l'analisi del comportamento del circuito al variare dei parametri che lo caratterizzano diventa più complessa e sempre più di frequente è necessario far ricorso a tecniche di tipo numerico per la soluzione delle equazioni.

Le equazioni differenziali ordinarie entrano anche in diversi altri settori, come ad esempio la cinetica chimica. La velocità di reazioni chimiche dipende da una quantità di fattori, come concentrazione dei reagenti, temperatura del sistema, pressione, ecc. In particolari condizioni (in pratica se il sistema nel quale avviene la reazione si può considerare omogeneo nello spazio) l'andamento delle reazioni è governato da sistemi di equazioni differenziali ordinarie.

Senza pretendere di presentare una panoramica esauriente sui sistemi retti da equazioni differenziali ordinarie, ci limitiamo a presentare alcuni modelli, che introduciamo in questa sezione. L'applicazione delle tecniche numeriche a tali modelli e la visualizzazione delle soluzioni verrà discussa alla fine del capitolo.

Esempio 8.1 [Un problema della meccanica: il problema degli N corpi] Come è ben noto, il moto dei pianeti attorno al sole segue con ottima approssimazione le leggi di Keplero. Queste affermano che, nel suo moto orbitale,

1. ciascun pianeta descrive attorno al Sole un'orbita ellittica ed il Sole occupa uno dei fuochi

2. nel suo moto, il raggio che collega il pianeta al sole copre aree uguali in tempi uguali

3. il rapporto fra il quadrato del periodo dell'orbita ed il cubo del semiasse maggiore dell'ellisse è uguale per tutti i pianeti.

Queste leggi, enunciate da Keplero all'inizio del diciassettesimo secolo, hanno permesso a Newton di formulare la sua legge di gravitazione universale. La legge di Newton sulla gravitazione, unitamente alla sua ben nota relazione fra massa ed accelerazione, permettono di descrivere, con eccellente accuratezza, il moto dei pianeti, e dei satelliti attorno ad essi.

Consideriamo un sistema di N punti materiali P_i, di massa m_i, $i = 1, \ldots, N$. Ciascun punto P_i è un vettore posizione di coordinate $P_i = (x_i, y_i, z_i)$. Ogni punto è soggetto all'azione della forza attrattiva da parte degli altri $N-1$ punti. Si può dunque scrivere la relazione che lega la accelerazione alla forza (legge di Newton della dinamica)

$$m_i \frac{d^2 P_i}{dt^2} = F_i. \tag{8.1}$$

La forza che agisce su ciascun punto P_i è data proprio dalla legge di Newton

$$F_i = -\gamma \sum_{\substack{j=1 \\ j \neq i}}^{N} m_i m_j \frac{P_i - P_j}{|P_i - P_j|^3}, \tag{8.2}$$

dove la costante γ rappresenta la costante di gravitazione universale.

Le equazioni (8.1,8.2) costituiscono un sistema di $3N$ equazioni differenziali del secondo ordine (poiché ciascun punto materiale è individuato da tre coordinate).

Una volta note posizione e velocità iniziale delle particelle, le equazioni permettono di determinare posizione e velocità di tutte le particelle per ogni istante futuro.

Tuttavia tale sistema di equazioni differenziali può essere risolto analiticamente solo per $N = 2$. Per tre o più corpi non è possibile scrivere la soluzione delle equazioni in forma analitica. Anche studiare il comportamento qualitativo delle soluzioni per tempi asintoticamente lunghi risulta molto difficile per più di due corpi mutuamente gravitanti.

Come vedremo, un accorto uso di appropriati metodi numerici consente di risolvere con notevole accuratezza il problema degli N corpi per tempi relativamente lunghi.

Alla fine del capitolo studieremo in dettaglio come integrare le equazioni che descrivono il moto del sistema Sole-Terra-Luna utilizzando dati realistici. ■

Esempio 8.2 [Un problema di teoria dei circuiti]
I circuiti possono essere descritti a diversi livelli di approssimazione. Una approssimazione molto comune è quella di considerare una rete elettrica come un grafo, dove ciascun ramo è costituito da un dispositivo (bipolo) definito mediante una relazione (funzionale o differenziale) fra la corrente che lo attraversa e la tensione ai suoi capi. La Figura (8.1) mostra i bipoli di uso più comune nelle reti: un resistore, un capacitore (più comunemente detto condensatore) ed un induttore. Tali bipoli sono poi connessi fra loro, e sono collegati ad interruttori e generatori, per formare il circuito vero e proprio. Naturalmente esistono altri elementi più complessi che

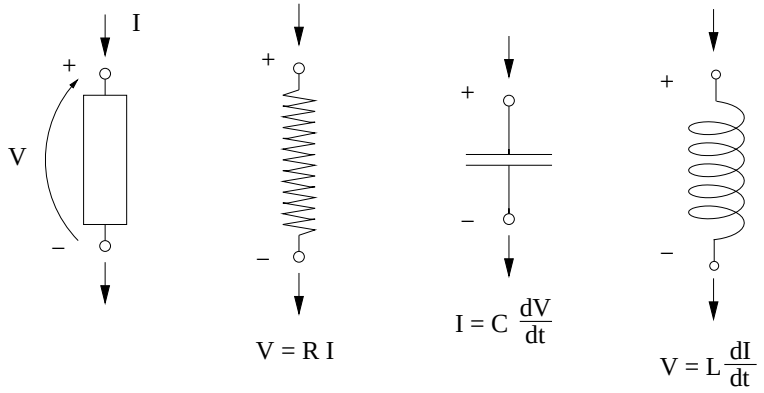

Figura 8.1 Bipoli di uso più comune e relazione fra tensione e corrente: (a) generico bipolo e convenzione di segno fra corrente in ingresso e tensione ai capi, (b) resistore elettrico, (c) capacitore, (d) induttore.

intervengono nella costruzione dei circuiti, come diodi e transistor. I diodi hanno una caratteristica tensione-corrente fortemente non lineare. I transistor sono schematizzati come elementi a tre morsetti (o tre poli), ed il loro comportamento è fortemente non lineare.

Un semplice circuito RLC, illustrato nella Figura (8.2), è governato da una equazione che stabilisce che la somma delle tensioni ai capi di ciascun bipolo deve essere nulla non appena l'interruttore viene chiuso. Le equazioni prendono la forma

$$IR + \frac{1}{C}\int I(t')\,dt' + L\frac{dI}{dt} = 0.$$

Differenziando rispetto al tempo si ottiene una equazione del secondo ordine per la corrente:

$$L\frac{d^2I}{dt^2} + R\frac{dI}{dt} + \frac{I}{C} = 0.$$

L'equazione è lineare e può essere risolta analiticamente. Se però invece del semplice resistore lineare si utilizza un elemento attivo con una diversa caratteristica tensione-corrente, ad esempio una relazione del tipo

$$V = -RI\left(1 - \frac{I^2}{I_0^2}\right),$$

che presenta una resistenza differenziale negativa per piccole correnti. Riscalando opportunamente le variabili ed il tempo, l'equazione risultante per la corrente è della forma

$$I'' - \mu(1 - I^2)I' + I = 0,$$

che rappresenta la equazione dell'oscillatore di van der Pol.[1]

[1] In realtà il circuito di van der Pol è costituito da elementi lineari connessi ad un triodo a vuoto. L'elemento attivo qui utilizzato ha solo una funzione didattica

Tale equazione deve essere risolta numericamente e, come vedremo, malgrado la sua semplice struttura, presenta certe insidie per elevati valori del parametro μ. ■

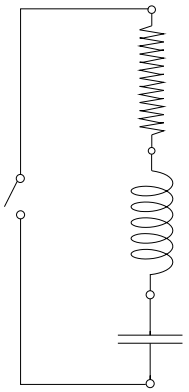

Figura 8.2 Circuito RLC. Le soluzioni con elementi lineari e valori positivi della resistenza sono oscillazioni smorzate.

Esempio 8.3 [Moto inerziale su superfici]
Un interessante problema è rappresentato dalla dinamica di un punto materiale vincolato ad una superficie liscia.

Supponiamo che un punto materiale di coordinate P sia vincolato a stare su una superficie immersa in $\mathbb{R}^n$, di equazione $\varphi(P) = 0$. Esistono diversi modi di trattare il problema, che dipendono fondamentalmente dalla maniera in cui si rappresenta la superficie. Se essa, ad esempio, viene rappresentata mediante due coordinate lagrangiane, $q = (\xi,\eta)$, e se le forze esterne sono *conservative*, cioè si ottengono come gradiente du un potenziale, le equazioni del moto per le due coordinate si ottengono scrivendo la cosiddetta funzione Lagrangiana del sistema, $L(q,q') = T(q,q') - U(q)$, dove T è l'energia cinetica dei sistema e U l'energia potenziale, espresse in funzione delle coordinate q e delle loro derivate prime. Le equazioni del moto sono allora le cosiddette equazioni di Eulero-Lagrange

$$\frac{d}{dt}\frac{\partial L}{\partial q'} - \frac{\partial L}{\partial q} = 0.$$

Un vantaggio della rappresentazione in termini di coordinate lagrangiane è che il numero di equazioni è più piccolo (due anzicché tre), il vincolo della superficie non compare esplicitamente, e viene automaticamente soddisfatto. Un vantaggio dell'approccio basato sulla descrizione di coordinate cartesiane e della rappresentazione della superficie nella forma $\varphi(P) = 0$ è che ammette una descrizione più elementare, fornisce automaticamente il valore della reazione vincolare, che rappresenta la sollecitazione alla superficie dovuta alle accelerazioni del punto, e, soprattutto, permette una rappresentazione globale della superficie, senza problemi legati ad eventuali singolarità introdotte dalla parametrizzazione. Basti pensare anche semplicemente alla singolarità che si presenta quando si rappresenta un punto su una sfera mediante due angoli ϑ,φ. Per $\vartheta = 0$ ("polo Nord" della sfera) e

per $\vartheta = \pi$ ("polo Sud" della sfera) si perde la corrispondenza biunivoca fra punti sulla sfera e coordinate. Lo svantaggio di tale rappresentazione è che il vincolo funzionale diventa un vincolo differenziale. Esso non è più dunque automaticamente soddisfatto dalla soluzione numerica e, se non si usano certi accorgimenti, c'è il rischio che l'errore sul soddisfacimento del vincolo possa diventare grande per lunghi tempi di integrazione.

Se il punto materiale è soggetto ad un campo di forze F, il suo moto sarà descritto dalle equazioni di Newton

$$mP'' = F + \Phi, \tag{8.3}$$

dove la quantità Φ rappresenta la reazione vincolare che impone che il punto stia sulla superficie. Poiché la superficie è liscia, la reazione vincolare è ortogonale alla superficie stessa, quindi è del tipo

$$\Phi = \lambda \nabla \varphi.$$

La quantità λ si ottiene proprio imponendo che il punto rimanga sulla superficie. Infatti, differenziando la relazione $\varphi(P) = 0$ si ha

$$\nabla \varphi \cdot P' = 0, \tag{8.4}$$

che, in componenti, diventa

$$\sum_i \frac{\partial \varphi}{\partial x_i} P_i' = 0, \tag{8.5}$$

e, differenziando quest'ultima,

$$P' \cdot \nabla \nabla \varphi \cdot P' + \nabla \varphi \cdot P'' = 0, \tag{8.6}$$

o, in componenti,

$$\sum_{ij} P_i' \frac{\partial^2 \varphi}{x_i x_j} P_j' + \sum_i \frac{\partial \varphi}{\partial x_i} P_i'' = 0.$$

Moltiplicando la (8.3) per $\nabla \varphi$, sfruttando le equazioni ((8.5)) e ((8.6)), si ottiene per λ l'espressione

$$\lambda = -\frac{mP' \cdot \nabla \nabla \varphi P' + \nabla \varphi \cdot F}{|\nabla \varphi|^2}.$$

Tale relazione, sostituita nelle precedenti, fornisce una equazione del secondo ordine che descrive il moto di un punto vincolato ad una superficie,

$$mP'' = F - \frac{\nabla \varphi \cdot F}{|\nabla \varphi|^2} \nabla \varphi - m \frac{P' \cdot \nabla \nabla \varphi \cdot P'}{|\nabla \varphi|^2} \nabla \varphi. \tag{8.7}$$

Osserviamo che la presenza del vincolo ha due effetti: da un lato il contributo del campo di forze che effettivamente influenza la dinamica è solo la proiezione sul piano tangente alla superficie del campo di forze F (i primi due termini alla destra dell'uguale), e poi c'è un contributo di natura geometrica che rimane anche in assenza di forze esterne. In quest'ultimo caso, in particolare, il punto materiale si muove per inerzia, conservando l'energia cinetica, cioè il modulo della velocità. Le traiettorie percorse sono delle *geodetiche* della superficie, cioè delle linee che hanno la proprietà di essere linee di minima distanza fra due punti vicini.

Nell'ultima sezione vedremo l'applicazione di alcuni metodi numerici al calcolo del moto di traiettorie lungo una superficie, sia in presenza che in assenza di campo di forze. ■

8.2 Cenni di teoria delle equazioni differenziali ordinarie

Un sistema di equazioni differenziali del primo ordine in forma esplicita si può scrivere nella forma

$$\begin{aligned} y_1' &= f_1(t,y_1,\ldots,y_m), \\ y_2' &= f_2(t,y_1,\ldots,y_m), \\ \ldots & \quad \ldots \\ y_m' &= f_m(t,y_1,\ldots,y_m), \end{aligned} \tag{8.8}$$

dove $y_1,\ldots,y_m$ sono m funzioni della variabile indipendente t, l'apice denota la derivata rispetto a t, e $f_1,\ldots,f_m$ sono m funzioni di $m+1$ variabili. La soluzione di tale sistema non è unica, ma dipende da un numero di parametri pari al numero di equazioni del sistema (cioè m). Un tipico problema che, sotto opportune condizioni, ammette esistenza ed unicità della soluzione è il cosiddetto *problema di Cauchy* o *problema ai valori iniziali*. Si cerca una soluzione del sistema (8.8) per valori della variabile indipendente in un certo intervallo $[t_0,T]$, e si assegnano i valori delle funzioni incognite nell'istante iniziale $t = t_0$. Si impone cioè che

$$y_1(t_0) = y_1^0,\ \ldots,\ y_m(t_0) = y_m^0. \tag{8.9}$$

Il sistema (8.8) si può scrivere in forma compatta utilizzando una notazione vettoriale,

$$y' = f(t,y(t)), \quad t \in [t_0 T], \tag{8.10}$$

dove $y : \mathbb{R}^m \to \mathbb{R}$ è un vettore di m dimensioni funzione della variabile indipendente t, $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ e y' denota la derivata della funzione (vettoriale) y rispetto a t. Il problema ai valori iniziali si definisce assegnando la condizione iniziale del tipo

$$y(t_0) = y_0 \in \mathbb{R}^m. \tag{8.11}$$

Notiamo che la trattazione dei sistemi del primo ordine include come caso particolare le equazioni scalari di ordine m. Data infatti l'equazione nella forma esplicita

$$y^{(m)} = f(t,y,y',\ldots,y^{(m-1)}),$$

ponendo $y_1 = y$, $y_2 = y'$, $\ldots$, $y_m = y^{(m-1)}$, questa risulta equivalente al sistema

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= y_3, \\ \ldots & \quad \ldots \\ y_{m-1}' &= y_m, \\ y_m' &= f(t,y_1,\ldots,y_m). \end{aligned} \tag{8.12}$$

Citiamo qui alcuni risultati della analisi relativi alle proprietà matematiche dei sistemi di equazioni differenziali ordinarie.

Quando si considera un problema di matematica, la prima domanda che ci si pone riguarda solitamente l'esistenza e, possibilmente, l'unicità della soluzione.

Le proprietà della soluzione dell'equazione (8.10) dipendono dalle proprietà della funzione f. Un requisito minimo che garantisca esistenza ed unicità della soluzione è la proprietà di lipschitzianità. Questa proprietà rappresenta un grado di regolarità intermedio fra la continuità e la differenziabilità (con derivate limitate), e garantisce che gli incrementi di una funzione possano sempre essere maggiorati con incrementi della variabile indipendente (moltiplicati per una opportuna costante). Più formalmente

Definizione 8.1 *Una funzione $f : \mathbb{R}^m \to \mathbb{R}^m$, definita in un insieme $\Omega \subseteq \mathbb{R}^m$, si dice lipschitziana in Ω di costante L se $\forall x,y \in \Omega$ si ha $\|f(x) - f(y)\| \leq L\|x - y\|$.*

Ovviamente se una funzione è lipschitziana, essa è anche continua in Ω.

Per il problema ai valori iniziali descritto dalla equazione (8.10) vale il seguente teorema

Teorema 8.1 (Teorema di Cauchy) *Se la funzione $f(t,y)$ è continua in t, e lipschitziana nel secondo argomento in tutta la striscia $t_0 \leq t \leq T$, $y \in \mathbb{R}^m$, con costante di Lipschitz L, allora si ha*

- *la soluzione del problema ai valori iniziali (8.10)-(8.9) esiste ed è unica in tutto l'intervallo $[t_0,T]$;*
- *la soluzione dipende con continuità dai dati iniziali. Più precisamente, se indichiamo con $z(t)$ la soluzione del problema $z' = f(t,z)$, $z(t_0) = z_0$, allora per la differenza fra $y(t)$ e $z(t)$ vale la seguente maggiorazione*

$$\|y(t) - z(t)\| \leq \|y_0 - z_0\| \exp(-L(t - t_0)).$$

Tale teorema è di importanza fondamentale, non solo per la teoria delle equazioni differenziali, ma anche per le applicazioni. In particolare, la dipendenza continua dai dati è legata alla possibilità pratica di approssimare la soluzione di un problema differenziale. Un problema per cui si abbia esistenza, unicità e dipendenza continua dai dati iniziali di dice *ben posto*. Quasi tutti i modelli matematici che si incontrano nelle scienze applicate e che portano ad un problema ai valori iniziali sono problemi ben posti. La mancanza della buona posizione di un problema è spesso (anche se non sempre) indice di una cattiva modellizzazione matematica di un fenomeno.

Se cade l'ipotesi di lipschitzianità non è detto che la soluzione del problema ai valori iniziali non esista. Si veda l'Esercizio (8.1) per un controesempio.

Per altri richiami sulla teoria delle equazioni differenziali ordinarie e sulle tecniche analitiche più comuni per la risoluzione di alcune classi di equazioni si veda ad esempio [23].

8.3 Il metodo di Eulero esplicito

Per approssimare la soluzione della equazione differenziale esistono diverse tecniche. Un approccio molto comune consiste nel suddividere l'intervallo di integrazione $[t_0,T]$ in un certo numero N di intervalli mediante i punti $t_0,t_1,\ldots,t_n,\ldots,t_N = T]$, e nel cercare quindi una approssimazione della soluzione nei punti $t_0,t_1,\ldots,t_N$. Una volta nota la soluzione in tali punti, sarà possibile ricostruirne una approssimazione in tutto l'intervallo $[t_0,T]$, ad esempio ricorrendo all'interpolazione

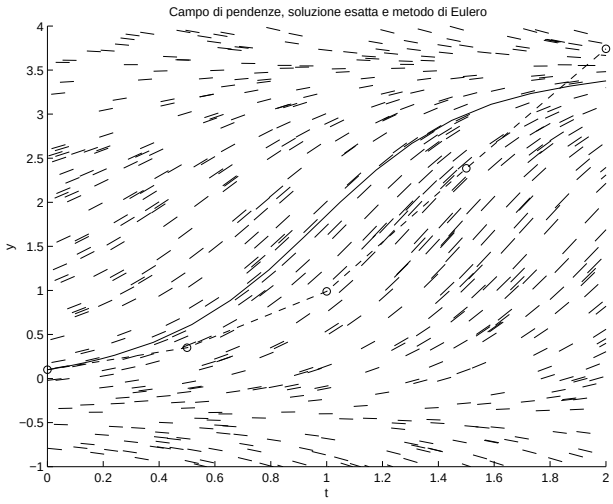

Figura 8.3 Viene visualizzato il campo di pendenza relativo alla equazione $y' = (2 + 3\sin(t))(2+5\sin(y))/10$, la soluzione della equazione corrispondente al dato iniziale $y(0) = 0.1$, e la soluzione numerica ottenuta con il metodo di Eulero con passo $h = 0.5$.

polinomiale a tratti. Per il momento consideriamo una suddivisione uniforme dell'intervallo $[t_0,T]$, cioè sia $t_n = t_0 + nh$, $n = 0, \ldots, N$, con $h = (T - t_0)/N$.

Consideriamo il seguente problema ai valori iniziali per una equazione scalare

$$y' = f(t,y), \quad y(t_0) = y_0.$$

Cercare la soluzione a tale problema significa cercare una funzione che passi per il punto (t_0,y_0) e che, per ogni punto$(t,y(t))$, abbia una pendenza pari a $f(t,y(t))$. La funzione f infatti individua nel piano t,y un campo di pendenze (si veda, ad esempio, la Figura (8.3)).

Consideriamo adesso la retta che passa per il punto (t_0,y_0). Tale retta risulta tangente alla soluzione del problema ai valori iniziali, poiché nel punto (t_0,y_0) retta e soluzione $y(t)$ hanno la stessa pendenza (vedi Figura (8.3)). Il valore della retta viene calcolato nel punto t_1, individuando così l'approssimazione numerica y_1 della soluzione nel punto t_1

$$y_1 = y(t_0) + hy'(t_0) = y_0 + hf(t_0,y_0).$$

Osserviamo che se h è un infinitesimo, allora $y(t_1) - y_1 = O(h^2)$, poiché la funzione e la retta tangente hanno un contatto del primo ordine nel punto (t_0,y_0). Più precisamente, utilizzando la formula di Taylor per lo sviluppo della soluzione in un intorno del punto t_0, si ha

$$y(t_1) - y_1 = \frac{1}{2}y''(\xi)h^2, \tag{8.13}$$

dove ξ è un valore compreso fra t_0 e t_1. Questa differenza, di fondamentale importanza per stabilire le proprietà di convergenza e di accuratezza dei metodi che

studieremo, prende il nome di *errore locale di troncamento*, e viene denotato con $\sigma(t_0;h)$. In alcuni casi è possibile una *stima a priori* dell'errore locale di troncamento, poiché la derivata seconda y'' si può esprimere in termini delle derivate della funzione f

$$y''(t) = \frac{\partial f(t,y)}{\partial t} + \frac{\partial f(t,y)}{\partial y} f(t,y).$$

Raggiunto il punto (t_1,y_1) è possibile iterare il procedimento, e costruire la retta passante per tale punto e di coefficiente angolare $f(t_1,y_1)$. L'ordinata del punto di ascissa t_2 di tale retta sarà y_2, ed il procedimento può essere iterato. Si ottiene così una sequenza di punti, generati come segue

$$y_{n+1} = y_n + hf(t_n,y_n), \quad n = 0,\ldots,N-1. \tag{8.14}$$

Il metodo così definito prende il nome di *metodo di Eulero esplicito*.

Una semplice funzione MATLAB che implementa N passi di tale metodo è la seguente

```
function [t,y] = EE(fname,y0,t0,T,N)
%
% Sintassi [t,y] = EE(fname,y0,t0,T,N)
% fname stringa che contiene il nome di una funzione
% della forma f(t,y) con t scalare e y vettore colonna.
% Risolve il problema di Cauchy utilizzando il metodo
% di Eulero esplicito con passo costante h = (T-t0)/N
%
h = (T-t0)/N;
t = linspace(t0,T,N+1)';
y = y0';
for n=1:N
   fn = feval(fname,t(n),y(n,:)');
   y = [y;y(n,:)+h*fn'];
end
```

I valori y_n di questa sequenza dovrebbero essere una approssimazione della soluzione per i corrispondenti valori della variabile indipendente

$$y_n \approx y(t_n).$$

Ci aspettiamo, inoltre, che, al diminuire del passo h di integrazione, l'approssimazione della soluzione numerica sia sempre migliore. Per quantificare meglio questa approssimazione è necessario introdurre il concetto di convergenza.

Definizione 8.2 *Diremo che il metodo numerico è convergente nel punto $t \in [t_0,T]$ per una data equazione differenziale se, data una suddivisione dell'intervallo in n intervalli uguali di ampiezza $h = (t-t_0)/n$, avviene che la soluzione numerica nel punto t tende alla soluzione esatta della equazione nel medesimo punto al tendere all'infinito del numero di suddivisioni, cioè che*

$$\lim_{\substack{n \to \infty \\ nh = t - t_0}} y_n = y(t).$$

Diremo poi che il metodo numerico converge in tutto l'intervallo $[t_0,T]$ se è convergente per ogni valore $t \in [t_0,T]$.

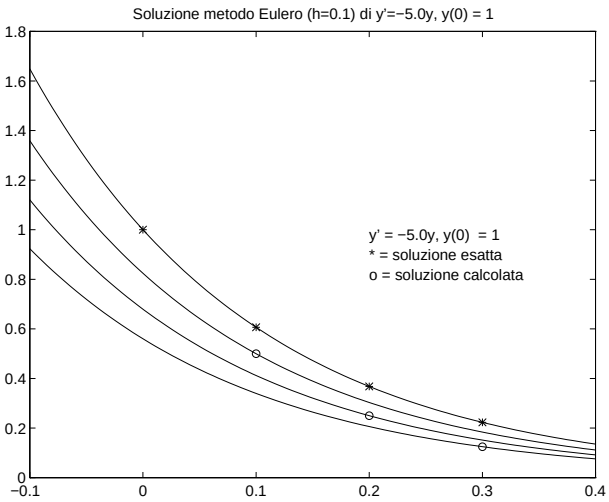

Figura 8.4 Errore di troncamento nel metodo di Eulero con passo $h = 0.1$ applicato al problema $y' = -5y$, $y(0) = 1$.

Diamo qui la dimostrazione della convergenza del metodo di Eulero applicato ad una equazione per la quale valgano le ipotesi del teorema di Cauchy.

Teorema 8.2 (Convergenza del metodo di Eulero esplicito) *Dato un sistema differenziale della forma (8.10), con $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ continua e derivabile nei suoi argomenti con derivata continua, e lipschitziana di costante L nel secondo argomento in tutta la striscia $[t0,T] \times \mathbb{R}^m$, allora il metodo di Eulero esplicito converge alla soluzione, e per ogni istante t, si ha la seguente maggiorazione dell'errore*

$$|e_n| = |y(t_n) - y_n| \leq \frac{1}{2L} \|y''\|_{\infty,[t_0,T]} \exp((L(t - t_0)) - 1)h,$$

dove

$$\|y''\| = \left\| \frac{\partial f}{\partial t} + \frac{\partial f}{\partial y} f \right\|.$$

DIMOSTRAZIONE. Applicando il metodo di Eulero si ottiene una successione di punti. Consideriamo il passaggio dal punto n al punto $n + 1$. Sia $e_n = y(t_n) - y_n$ l'errore che commettiamo al passo n. Vogliamo mostrare che al tendere a zero del passo di integrazione, tale errore tende a zero in tutti i punti t_n (ed in particolare nell'ultimo punto t_N). Per fare questo introduciamo la quantità $z_{n+1} = y(t_n) + hf(t_n,y(t_n))$ che rappresenta il valore che si ottiene applicando un passo del metodo di Eulero a partire dalla soluzione esatta nel punto t_n. La quantità $y(t_{n+1}) - z_{n+1}$ rappresenta l'errore locale di troncamento nel punto t_n, $\sigma(t_n; h)$. Il rapporto fra l'errore locale di troncamento ed il passo di integrazione h è detto *errore locale di discretizzazione*, e si indica comunemente con $d(t_n; h) = \sigma(t_n; h)/h$. Scriviamo l'errore al tempo t_{n+1} come

$$\begin{aligned} |e_{n+1}| &= |y(t_{n+1}) - y_{n+1}| = |y(t_{n+1}) - z_{n+1} + z_{n+1} - y_{n+1}|, \\ &\leq |\sigma(t_n; h)| + |y(t_n) + hf(t_n,y(t_n)) - y_n - hf(t_n,y_n)|, \end{aligned}$$

$$\leq |\sigma(t_n;h)| + |e_n| + h|f(t_n,y(t_n)) - f(t_n,y_n)|,$$
$$\leq |e_n|(1+Lh) + |\sigma(t_n;h)|,$$

dove l'ultima disuguaglianza è stata ottenuta grazie alla proprietà di lipschitzianità della funzione f. Supponiamo di conoscere una stima uniforme D dell'errore locale di troncamento, cioè sia

$$|\sigma(t;h)| \leq D,\ t \in [t_0,T].$$

L'errore e_n soddisfa allora la seguente disequazione alle differenze lineare

$$|e_{n+1}| \leq (1+hL)|e_n| + D.$$

Iterando la disequazione si ottiene

$$\begin{aligned}
|e_{n+1}| &\leq (1+hL)|e_n| + D,\\
&\leq (1+hL)((1+hL)|e_{n-1}| + D) + D,\\
&= (1+hL)^2|e_{n-1}| + [(1+hL)+1]D,\\
&\leq (1+hL)^3|e_{n-2}| + [(1+hL)^2 + (1+hL) + 1]D,\\
&\leq \cdots\\
&\leq (1+hL)^{n+1}|e_0| + D\sum_{j=0}^{n}(1+hL)^j,\\
&= (1+hL)^{n+1}|e_0| + D\frac{(1+hL)^{n+1}-1}{hL}.
\end{aligned}$$

L'ultima uguaglianza è stata ottenuta dalla espressione della somma dei primi n termini di una progressione geometrica. Ricordiamo infatti che data

$$S_n = \sum_{i=0}^{n} q^i,$$

il termine S_{n+1} si può scrivere come

$$S_{n+1} = S_n + q^{n+1} = \sum_{i=0}^{n+1} q^i = 1 + \sum_{i=1}^{n+1} q^i = 1 + q\sum_{i=0}^{n} q^i = 1 + qS_n.$$

Dall'uguaglianza fra il secondo e l'ultimo termine segue

$$S_n = \frac{1-q^{n+1}}{1-q}.$$

Se l'errore iniziale è nullo si ha quindi, $\forall n$,

$$|e_n| \leq D\frac{(1+hL)^n - 1}{hL}.$$

Dall'espressione dell'errore locale di troncamento segue che se la soluzione è una funzione sufficientemente regolare, la quantità D è un infinitesimo del secondo ordine in h, e quindi

$$|e_{n+1}| \leq Qh \to 0, \text{ per } h \to 0.$$

Dalla stima per D ottenuta dalla ((8.13)),

$$D \leq \frac{1}{2}\max_{t_0\leq t\leq T}|y''(t)|h^2,$$

si ottiene una maggiorazione per la quantità Q

$$Q \leq \frac{1}{2}\|y''\|_{\infty,[t_0,T]}\frac{(1+hL)^N - 1}{L} \leq \frac{1}{2}\|y''\|_{\infty,[t_0,T]}\frac{1}{L}(\exp(L(T-t_0)) - 1),$$

in quanto dalla disuguaglianza $1+x < \exp(x)$ discende $(1+x)^N < \exp(Nx)$, con $x = hL$. ■

La stima così ottenuta ci fornisce diverse informazioni. Innanzitutto ci dice che se la funzione f è lipschitziana in y allora il metodo di Eulero converge, e l'errore globale e_n è un infinitesimo di ordine uno in h, esattamente lo stesso ordine di infinitesimo dell'errore locale di discretizzazione. Come vedremo, questa proprietà non è ristretta al metodo di Eulero, ma è comune a tutti i cosiddetti *metodi ad un passo*. Poi vediamo che l'errore globale è stimato da una quantità che cresce esponenzialmente con l'ampiezza dell'intervallo di integrazione. Tale quantità appare anche nella stima del fattore di amplificazione dell'errore iniziale che si osserva quando si studia la dipendenza continua dai dati iniziali. Se il prodotto $L(T - t_0)$ è grande, l'errore locale (e così pure un eventuale errore sui dati iniziali) viene amplificato molto. La costante $\exp(L(T - t_0))$ assume qui il ruolo di numero di condizionamento nella risoluzione di un problema ai valori iniziali. Se tale numero è molto alto, piccole perturbazioni vengono amplificate molto, e quindi ci aspettiamo che un maggiore sforzo debba essere compiuto per ottenere una soluzione accurata.

Il metodo appena studiato è chiamato metodo di Eulero esplicito (abbreviato: EE), o in avanti. È detto esplicito poiché il valore della soluzione numerica al passo $n+1$ è calcolato esplicitamente dal valore al tempo n. Esso è anche detto *in avanti* poiché se si pensa di scrivere l'equazione differenziale ad un certo tempo t, il metodo di Eulero esplicito si ottiene discretizzando la derivata rispetto al tempo della funzione y come

$$\frac{dy(t)}{dt} \approx \frac{y(t+h) - y(t)}{h}.$$

Per contro, se si discretizza la derivata rispetto al tempo *all'indietro* si ottiene il metodo di Eulero implicito (brevemente EI) o all'indietro, definito dalla relazione

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}). \tag{8.15}$$

Per ottenere un passo del metodo di Eulero implicito occorre risolvere una equazione non lineare[2]. Il metodo di Eulero implicito ha lo stesso ordine di accuratezza del metodo di Eulero esplicito. In genere quindi, richiedendo la risoluzione di un'equazione non lineare, sarà molto più costoso avanzare di un passo con EI che con EE. Come vedremo più avanti, i metodi impliciti possono presentare dei vantaggi che li rendono più convenienti in taluni casi.

Osservazione 8.1 I metodi EE ed EI, come pure tutti i metodi che vedremo in questo capitolo, si applicano indifferentemente alla singola equazione o ad un sistema di equazioni. Di fatto l'espressione formale dei metodi è identica nel caso dell'equazione e nel caso dei sistemi, pur di utilizzare una notazione vettoriale.

Un passo del metodo EI applicato ad un sistema di m equazioni differenziali richiede in generale la soluzione di un sistema non lineare di m equazioni. Discuteremo questo problema quando parleremo della implementazione dei metodi.

Prima di addentrarci nello studio generale dei metodi ad un passo, facciamo la seguente considerazione. Il metodo di Eulero in avanti applicato ad un sistema si può dedurre approssimando la funzione $y(t)$ con il suo sviluppo di Taylor

[2] Si potranno a tale scopo utilizzare le tecniche viste nel Capitolo 4 e varranno in particolare le considerazioni fatte sulle richieste necessarie alla convergenza di tali metodi.

nell'intorno del punto t_0, arrestato al primo ordine. Verrebbe naturale pensare di costruire metodi più accurati utilizzando degli sviluppi di Taylor di ordine più elevato.

In effetti questo è possibile. Infatti lo sviluppo di Taylor della soluzione nell'intorno del punto t_0 si scrive

$$y(t) = y_0 + hy'_0 + \frac{1}{2}h^2 y''(y_0) + O(h^3). \tag{8.16}$$

La derivata seconda nel punto t_0 può essere calcolata come derivata di una funzione composta:

$$y''(t) = \frac{d}{dt}f(t,y(t)) = \frac{\partial f(t,y)}{\partial t} + \frac{\partial f(t,y)}{\partial y}y'(t) = \frac{\partial f(t,y)}{\partial t} + \frac{\partial f(t,y)}{\partial y}f(t,y).$$

Notiamo che, nel caso di un sistema, l'ultimo termine rappresenta un prodotto matrice per vettore. Scritto in componenti, esso diventa

$$\left(\frac{\partial f}{\partial y}f\right)_i = \sum_{j=1}^{m} \frac{\partial f_i}{\partial y_j} f_j,$$

dove, per snellire la notazione, sono stati omessi gli argomenti di f. La matrice $\partial f/\partial y$ è chiamata matrice jacobiana del sistema.

Trascurando quindi infinitesimi del terzo ordine in h, possiamo dunque costruire un metodo basato sugli sviluppi di Taylor:

$$y_{n+1} = y_n + hf(t_n,y_n) + \frac{1}{2}h^2\left(\frac{\partial f(t_n,y_n)}{\partial t} + \frac{\partial f(t_n,y_n)}{\partial y}f(t_n,y_n)\right). \tag{8.17}$$

Per un tale metodo l'errore locale di troncamento è $O(h^3)$ e quindi, come vedremo, se il metodo risulta convergente, il suo errore globale sarà un infinitesimo del secondo ordine.

Per la costruzione di un metodo alternativo che fornisca una analoga accuratezza potremmo fare la seguente considerazione. Se riuscissimo a stimare la pendenza della retta secante che passa per i punti (t_0,y_0) e $(t_1,y(t_1))$ meglio che con la retta tangente alla soluzione nel punto (t_0,y_0) otterremmo una approssimazione migliore di quella ottenuta con il metodo di Eulero esplicito (vedi Figura (8.5)); si può immaginare che una buona approssimazione del coefficiente angolare della secante sia il valore della f nel punto della soluzione di ascissa $t_n + h/2$, e che questa, a sua volta, sia ben approssimabile dal valore della f ottenuto nel punto che sta sulla retta tangente alla curva nel punto $(t_n,y(t_n))$, alla stessa ascissa $t_n + h/2$. Tale punto può essere ottenuto con un passo di ampiezza $h/2$ del metodo di Eulero esplicito. Si ottiene così il seguente metodo

$$y_{n+1} = y_n + hf(t_n + h/2, y_n + (h/2)f(t_n,y_n)), \tag{8.18}$$

detto metodo di Eulero modificato. Come vedremo più avanti, esso ha una accuratezza del secondo ordine. Il metodo di Eulero modificato appare più semplice del metodo basato sullo sviluppo di Taylor al secondo ordine. Infatti, mentre lo sviluppo di Taylor richiede, oltre la valutazione della f anche la valutazione della sua derivata rispetto al tempo e della matrice jacobiana, il metodo di Eulero

modificato richiede solo due valutazioni della f. Per questo motivo metodi basati sugli sviluppi di Taylor non vengono comunemente usati nella pratica.

Un altra idea per ottenere un metodo più accurato del metodo di Eulero potrebbe essere basato sull'utilizzo di una media delle pendenze ai tempi t_n e t_{n+1}. Quest'ultima potrebbe essere stimata mediante un passo del metodo di Eulero. Si ottiene così lo schema seguente

$$y_{n+1} = y_n + \frac{1}{2}h(f(t_n,y_n) + f(t_n + h,y_n + hf(t_n,y_n))), \tag{8.19}$$

Tale schema viene detto schema di Heun.

Tutti i metodi visti finora rientrano nella categoria dei cosiddetti *metodi ad un passo*, poiché per calcolare la soluzione numerica al tempo t_{n+1} è sufficiente conoscere la soluzione numerica al tempo t_n. La forma generale dei metodi ad un passo è la seguente

$$y_{n+1} = y_n + h\Phi(t_n,y_n;h,f). \tag{8.20}$$

La funzione $\Phi(t_n,y_n;h,f)$ caratterizza il metodo. Essa rappresenta una approssimazione numerica della media della funzione f nell'intervallo $[t_n,t_{n+1}]$. Osserviamo che la soluzione esatta della equazione soddisfa la relazione

$$\frac{y(t+h) - y(t)}{h} = \Phi(t,y(t);h,f) + d(t;h),$$

dove $d(t;h)$ è l'errore locale di discretizzazione.

Enunciamo adesso un teorema generale di convergenza per i metodi ad un passo, che conclude questa introduzione generale. Premettiamo la definizione di consistenza.

Definizione 8.3 *Un metodo ad un passo si dice consistente con l'equazione (8.10) nell'intervallo* $[t_0,T]$ *se l'errore locale di discretizzazione è infinitesimo per* $h \to 0$. *Più precisamente, se esiste una funzione* $d(h)$ *tale che* $|d(t;h)| \leq d(h)\ \forall t \in [t_0,T]$ *con*

$$\lim_{h\to 0} d(h) = 0.$$

Un metodo ad un passo si dirà poi di ordine di consistenza p *se*

$$d(h) = O(h^p).$$

Il metodo di Eulero esplicito è dunque un metodo ad un passo consistente, con ordine di consistenza 1. Il teorema generale di convergenza per i metodi ad un passo si può enunciare così:

Teorema 8.3 *Se la funzione* Φ *è una funzione lipschitziana rispetto ad* y *di costante* Q *per ogni* $h < h_0$, $h_0 > 0$, *e se* Φ *definisce un metodo consistente di ordine* p *in* $[t_0,T]$, *allora il metodo è convergente in* $[t_0,T]$ *ed ha ordine di convergenza* p.

La dimostrazione è del tutto analoga alla dimostrazione di convergenza del metodo di Eulero esplicito. Invitiamo il lettore a farla in dettaglio per esercizio.

Osserviamo che per definire un metodo ad un passo è sufficiente mostrare come calcolare y_1 a partire da y_0.

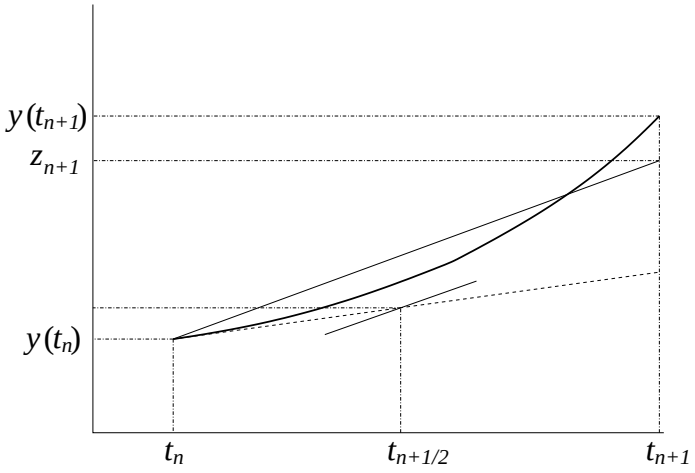

Figura 8.5 Costruzione geometrica del metodo di Eulero modificato

8.4 Metodi Runge-Kutta

Il metodo di Eulero modificato, introdotto nel paragrafo precedente, può essere agevolmente riscritto nella forma

$$\begin{aligned} k_1 &= f(t_0,y_0), \\ k_2 &= f(t_0 + h/2, y_0 + (h/2)k_1), \\ y_1 &= y_0 + hk_2, \end{aligned}$$

mentre lo schema di Heun può essere riscritto nella forma

$$\begin{aligned} k_1 &= f(t_0,y_0), \\ k_2 &= f(t_0 + h, y_0 + hk_1), \\ y_1 &= y_0 + \frac{1}{2}h(k_1 + k_2). \end{aligned}$$

Entrambi gli schemi hanno la seguente forma generale

$$\begin{aligned} k_1 &= f(t_0,y_0), \\ k_2 &= f(t_0 + ch, y_0 + hak_1), \\ y_1 &= y_0 + h(w_1k_1 + w_2k_2). \end{aligned} \tag{8.21}$$

Nel caso del metodo di Eulero modificato si ha $c = 1/2, a = 1/2, w_1 = 0\, w_2 = 1$, mentre per lo schema di Heun risulta $c = 1, a = 1, w_1 = 1/2, w_2 = 1/2$. Uno schema con questa struttura è detto *schema di Runge-Kutta* esplicito a due livelli. Esistono altre scelte valide per i parametri a, c, w_1, w_2? Se si, come fare a determinarle? E come si fa a mostrare che l'accuratezza dei metodi è effettivamente del secondo ordine, come ci suggerisce l'intuito geometrico?

La tecnica generale per verificare l'accuratezza di un metodo di Runge-Kutta esplicito è quella di effettuare lo sviluppo di Taylor della soluzione numerica e

confrontarlo con lo sviluppo della soluzione esatta, che può essere calcolato a partire dalla equazione stessa.

Consideriamo lo schema (8.21). Lo sviluppo di Taylor della soluzione numerica si ottiene a partire dallo sviluppo di Taylor del termine k_2. Si ha infatti:

$$\begin{aligned} k_2(h) &= k_2(0) + k_2'(0)h + O(h^2), \\ &= f + \left(c\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial y}k_1 \right) h + O(h^2), \end{aligned}$$

dove la f e le sue derivate sono calcolate nel punto (t_0,y_0). Sostituendo nella espressione di y_1 nella (8.21) si ha

$$y_1 = y_0 + h(w_1 + w_2)f + h^2 w_2 \left(c\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial y}k_1 \right) + O(h^3).$$

Confrontiamo tale espressione con lo sviluppo di Taylor della soluzione esatta, arrestato al secondo ordine,

$$y(t_0 + h) = y_0 + hf + \frac{1}{2}h^2 \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}f \right) + O(h^3).$$

Uguagliando i termini dello sviluppo di Taylor si ottengono le seguenti condizioni che i coefficienti a, c, w_1, w_2 devono rispettare affinché il metodo sia del secondo ordine:

$$\begin{aligned} w_1 + w_2 &= 1, \\ w_2 c &= 1/2, \\ w_2 a &= 1/2. \end{aligned} \tag{8.22}$$

Si verifica immediatamente che i metodi di Eulero generalizzato e di Heun soddisfano le condizioni (8.22), e quindi sono effettivamente del secondo ordine. Inoltre osserviamo che le condizioni (8.22) costituiscono un sistema di tre equazioni in quattro incognite, e quindi cerchiamo una famiglia ad un parametro di soluzioni. Ad esempio possiamo esprimere a, c, w_1 in funzione di w_2

$$a = c = \frac{1}{2w_2}, \quad w_1 = 1 - w_2.$$

Se provassimo ad imporre un accordo tra le due soluzioni (numerica ed esatta) anche per il successivo termine dello sviluppo di Taylor, scopriremmo che il numero complessivo di condizioni da imporre supera il numero di parametri a disposizione, e che non è possibile ottenere schemi a due livelli della forma (8.21) che siano del terzo ordine.

È possibile, però, considerare schemi di Runge-Kutta con un numero di livelli più elevato. In tal modo si ha a disposizione un maggior numero di parametri che consente di ottenere schemi di ordine più elevato.

Il generico schema di Runge-Kutta esplicito a ν livelli ha la seguente struttura

$$\begin{aligned} k_1 &= f(t_0,y_0), \\ k_2 &= f(t_0 + c_2 h, y_0 + h a_{21} k_1), \\ \dots &\quad \dots \end{aligned}$$

$$
\begin{aligned}
k_\nu &= f(t_0 + c_\nu h, y_0 + h(a_{\nu 1}k_1 + a_{\nu 2}k_2 + \ldots + a_{\nu\nu-1}k_{\nu-1})),\\
y_1 &= y_0 + h(w_1 k_1 + \ldots + w_\nu k_\nu),
\end{aligned}
$$

o, in forma più compatta,

$$k_i = f(t_0 + c_i h, y_0 + h\sum_{j=1}^{i-1} a_{ij}k_j), \tag{8.23}$$

$$y_1 = y_0 + h\sum_{i=1}^{\nu} w_i k_i. \tag{8.24}$$

Il generico metodo di Runge-Kutta esplicito a ν livelli è dunque individuato dalla matrice $A = (a_{ij})$ e dai vettori $\mathbf{c}$ e $\mathbf{w}$. Un modo usuale di rappresentarli è in forma di tabella (*tableau di Butcher*)

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{w}^T \end{array}$$

Osserviamo che la matrice A è una matrice triangolare con elementi sulla diagonale uguali a zero. È proprio questa struttura della matrice che permette di calcolare i k_i esplicitamente in successione, e fa si che il metodo sia esplicito. È possibile considerare anche metodi impliciti. Questi ultimi potranno essere diagonalmente impliciti (detti anche semi-impliciti) se la matrice A è una matrice triangolare inferiore, oppure totalmente impliciti, se la matrice A è piena. Il generico schema di Runge-Kutta a ν livelli si scrive

$$k_i = f(t_0 + c_i h, y_0 + h\sum_{j=1}^{\nu} a_{ij}k_j), \tag{8.25}$$

$$y_1 = y_0 + h\sum_{i=1}^{\nu} w_i k_i. \tag{8.26}$$

Per i metodi diagonalmente impliciti la sommatoria nella espressione dei k andrà di fatto da 1 a i. L'utilizzo di un metodo diagonalmente implicito richiede la soluzione di ν equazioni non lineari in sequenza (o di ν sistemi non lineari $m \times m$ nel caso di sistemi), mentre un metodo Runge-Kutta implicito richiede in generale la soluzione di un sistema di ν equazioni non lineari (o di νm equazioni nel caso di sistemi). Per questa ragione i metodi completamente impliciti, malgrado le eccellenti proprietà di accuratezza e stabilità, sono utilizzati di rado, e solo quando sono richieste caratteristiche particolari.

Per determinare metodi RK di ordine elevato occorre imporre che un certo numero di termini dello sviluppo di Taylor della soluzione esatta e della soluzione numerica siano uguali. Le equazioni risultanti vengono dette *condizioni sull'ordine*.

È possibile semplificare la ricerca delle condizioni sull'ordine osservando che lo studio delle proprietà dei metodi RK si può limitare alla loro applicazione ai cosiddetti *sistemi autonomi*, ossia sistemi di equazioni differenziali nei quali la funzione f non dipende esplicitamente da t.

Infatti, dato un sistema arbitrario di m equazioni, $y' = f(t,y)$, questo può essere considerato come un sistema autonomo di $m + 1$ equazioni, inglobando il

tempo come una delle incognite che soddisfa l'ovvia equazione $t' = 1, t(t_0) = t_0$. Si pone cioè $z = (t; y)$, $F = (1; f)$, e si riscrive il sistema nella forma $z' = F(z)$. Ebbene, si può dimostrare che se un metodo di Runge-Kutta soddisfa le seguenti condizioni

$$\sum_i w_i = 1, \quad \sum_j a_{ij} = c_i,$$

allora l'applicazione di tale metodo al sistema nella forma iniziale $y' = f(t,y)$ o nella forma di sistema autonomo $z' = F(z)$, produce i medesimi risultati. La prima condizione è una condizione di consistenza, necessaria affinché il metodo sia almeno del primo ordine, mentre l'altra condizione è generalmente utilizzata perché appunto semplifica l'analisi dei metodi.

Ricavare le condizioni sull'ordine per ordini elevati non è banale. Basti pensare che le derivate della soluzione esatta, pur limitandosi a considerare sistemi autonomi $y' = f(y)$, hanno espressioni del tipo

$$\begin{aligned} y' &= f, \\ y'' &= f'f, \\ y''' &= f''ff + f'f'f, \\ y^{(4)} &= f'''fff + f''f'ff + f''ff'f + f''ff'f + f'f''ff + f'f'f'f, \end{aligned}$$

dove con f' si è indicata la matrice jacobiana, cioè un oggetto a due indici che moltiplicato per un vettore fornisce un vettore, mentre f'' è un oggetto a tre indici che, moltiplicato per una coppia di vettori, fornisce un vettore, e così via. L'equazione per la derivata terza, scritta in maniera esplicita utilizzando una notazione con indici, diventa

$$y_i''' = \sum_{j\ell} \frac{\partial^2 f_i}{\partial y_j \partial y_\ell} f_j f_\ell + \sum_j \frac{\partial f_i}{\partial y_j} \sum_\ell \frac{\partial f_j}{\partial y_\ell} f_\ell.$$

Considerando che le derivate seconde miste di una funzione (regolare) sono uguali, è possibile semplificare l'espressione per la derivata quarta, ed ottenere

$$y^{(4)} = f'''fff + 3f''f'ff + f'f''ff + f'f'f'f.$$

Ciascun termine che appare nella espressione delle derivate è detto *differenziale elementare*. Esiste una teoria che sfrutta la corrispondenza biunivoca fra i differenziali elementari e gli *alberi radicati*. Tale teoria [5], permette di scrivere in maniera relativamente agevole, le condizioni sull'ordine per i coefficienti A, $\mathbf{c}$, $\mathbf{w}$ anche per metodi RK di ordine elevato.

La teoria delle condizioni sull'ordine esula gli scopi del presente volume. Ci limitiamo qui a riportare alcuni risultati che riguardano il crescere del numero di condizioni in funzione dell'ordine, e le cosiddette barriere sull'ordine, ossia quale è l'ordine massimo che si riesce ad ottenere per un metodo esplicito a ν livelli.

Le prime condizioni sull'ordine per un metodo di RK sono riportate nella Tabella 8.1. Il crescere del numero di condizioni all'aumentare dell'ordine è riportato nella Tabella 8.2.

I metodi impliciti, a parità di livelli, possono essere più accurati dei metodi espliciti, poiché dispongono di un maggior numero di parametri.

Ordine p	Condizioni
1	$\sum_i w_i = 1$
2	$\sum_i w_i c_i = \frac{1}{2}$
3	$\sum_i w_i c_i^2 = \frac{1}{3}$ $\sum_{ij} w_i a_{ij} c_j = \frac{1}{6}$
4	$\sum_i w_i c_i^3 = \frac{1}{4}$ $\sum_{ij} w_i c_i a_{ij} c_j = \frac{1}{8}$ $\sum_{ij} w_i a_{ij} c_j^2 = \frac{1}{2}$ $\sum_{ijk} w_i a_{ij} a_{jk} c_k = \frac{1}{24}$

Tabella 8.1 Condizioni sull'ordine per metodi di Runge-Kutta fino a $p = 4$.

Ordine p	1	2	3	4	5	6	7	8	9	10
Numero di condizioni	1	2	4	8	17	37	85	200	486	1205

Tabella 8.2 Numero di condizioni sull'ordine in funzione dell'ordine per metodi di Runge-Kutta.

È possibile, mediante una considerazione elementare, mostrare che l'ordine di un metodo di Runge-Kutta è limitato dalla relazione

$$p \leq 2\nu.$$

Infatti, consideriamo la semplice equazione differenziale

$$y' = f(t), \quad y(t_0) = y_0. \tag{8.27}$$

La soluzione dopo un passo h è data dall'integrale

$$y(t_0 + h) = y_0 + \int_{t_0}^{t_0+h} f(t)\, dt.$$

Un metodo di Runge-Kutta applicato alla equazione fornisce

$$y_1 = y_0 + h \sum_{i=1}^{\nu} w_i f(t_0 + c_i h).$$

La massima accuratezza possibile si ottiene quando si utilizza per il calcolo dell'integrale una formula di quadratura di Gauss-Legendre con ν nodi. Tale formula ha un ordine polinomiale $2\nu - 1$, quindi è esatta per tutti i termini dello sviluppo di Taylor della funzione f fino all'ordine $2\nu - 1$. L'errore, cioè la differenza fra $y(t_0 + h)$ e y_1 è dunque un infinitesimo $O(h^{2\nu+1})$, che indica che il metodo è al più di ordine 2ν.

Si può dimostrare che in effetti esistono schemi Runge-Kutta impliciti di ordine massimo $p = 2\nu$. Si tratta dei cosiddetti schemi di collocazione gaussiana, che definiremo più avanti.

Per i metodi espliciti vale la limitazione più severa

$$p \leq \nu.$$

Anzi, è possibile dimostrare che per i metodi RK espliciti vale la barriera sull'ordine illustrata schematicamente nella seguente tabella

Numero di livelli ν	1	2	3	4	5	6	7	8	9
Massimo ordine p	1	2	3	4	4	5	6	6	7

Concludiamo la sezione riportando due metodi RK espliciti di ordine rispettivamente 3 e 4 di uso comune

Metodo di RK3, $\nu = p = 3$

$$\begin{aligned} k_1 &= f(t_0,y_0), \\ k_2 &= f(t_0 + h/2,y_0 + h/2k_1), \\ k_3 &= f(t_0 + h,y_0 + h(-k_1 + 2k_2)), \\ y_1 &= y_0 + \frac{h}{6}(k_1 + 4k_2 + k_3), \end{aligned}$$

Metodo di RK4, $\nu = p = 4$

$$\begin{aligned} k_1 &= f(t_0,y_0), \\ k_2 &= f(t_0 + h/2,y_0 + h/2k_1), \\ k_3 &= f(t_0 + h/2,y_0 + h/2k_2), \\ k_4 &= f(t_0 + h,y_0 + hk_3), \\ y_1 &= y_0 + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \end{aligned}$$

ed una funzione MATLAB che implementa un passo dei metodi di Runge-Kutta di ordine $p = 1,\ldots,5$

```
  function [tn1,yn1,fn1] = RKpasso(fname,tn,yn,fn,h,p)
%
% Sintassi [tn1,yn1,fn1] = RKpasso(fname,tn,yn,fn,h,p)
%
% fname stringa che contiene il nome di una funzione
% della forma f(t,y) con t scalare e y vettore colonna.
% h passo di discretizzazione.
% k ordine del metodo Runge-Kutta utilizzato, 1<=p<=5.
%
   switch p
    case 1
      k1 = h*fn;
      yn1 = yn + k1;
    case 2
      k1 = h*fn;
      k2 = h*feval(fname,tn+h,yn+k1);
      yn1  = yn + (k1 + k2)/2;
    case 3
      k1 = h*fn;
      k2 = h*feval(fname,tn+(h/2),yn+(k1/2));
```

```
    k3 = h*feval(fname,tn+h,yn-k1+2*k2);
    yn1  = yn + (k1 + 4*k2 + k3)/6;
  case 4
    k1 = h*fn;
    k2 = h*feval(fname,tn+(h/2),yn+(k1/2));
    k3 = h*feval(fname,tn+(h/2),yn+(k2/2));
    k4 = h*feval(fname,tn+h,yn+k3);
    yn1  = yn + (k1 + 2*k2 + 2*k3 + k4)/6;
  case 5
    k1 = h*fn;
    k2 = h*feval(fname,tn+(h/4),yn+(k1/4));
    k3 = h*feval(fname,tn+(3*h/8),yn+(3/32)*k1+(9/32)*k2);
    k4 = h*feval(fname,tn+(12/13)*h,yn+(1932/2197)*k1-...
         (7200/2197)*k2+(7296/2197)*k3);
    k5 = h*feval(fname,tn+h,yn+(439/216)*k1 - 8*k2 +...
         (3680/513)*k3 -(845/4104)*k4);
    k6 = h*feval(fname,tn+(1/2)*h,yn-(8/27)*k1 + 2*k2 -...
         (3544/2565)*k3 + (1859/4104)*k4 - (11/40)*k5);
    yn1  = yn + (16/135)*k1 + (6656/12825)*k3 +...
         (28561/56430)*k4 - (9/50)*k5 + (2/55)*k6;
end
tn1 = tn+h;
fn1 = feval(fname,tn1,yn1);
```

8.4.1 Metodi di collocazione

Accenniamo brevemente ad una famiglia di metodi Runge-Kutta impliciti, detti metodi di collocazione. Tali metodi si ottengono cercando un polinomio p_ν di grado ν, che in un certo numero ν di punti, $t_0 + c_i h$, $i = 1, \ldots \nu$, detti nodi di collocazione, soddisfi l'equazione:

$$p'(t_0 + c_i h) = f(t_0 + c_i h, p(t_0 + c_i h)), \quad i = 1, \ldots, \nu. \tag{8.28}$$

L'equazione (8.28), unitamente alla condizione di passaggio per il punto (t_0, y_0), costituisce un sistema di $\nu + 1$ equazioni nei $\nu + 1$ parametri che definiscono il polinomio. Posto $k_i = f(t_0 + c_i h, p(t_0 + c_i h))$, si può dimostrare che l'approccio di collocazione è equivalente ad un metodo Runge-Kutta implicito della forma (8.25), con

$$\begin{aligned} a_{ij} &= \int_0^{c_i} L_j(x)\, dx, \quad i,j = 1, \ldots, \nu, \\ w_i &= \int_0^1 L_i(x)\, dx, \quad i = 1, \ldots, \nu, \end{aligned}$$

dove $L_i(x)$ sono i polinomi elementari di Lagrange corrispondenti ai nodi c_i nell'intervallo [0,1]

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{\nu} \frac{x - c_j}{c_i - c_j}.$$

Questi metodi di Runge-Kutta (impliciti poiché A è una matrice piena) hanno molte proprietà che li rendono interessanti. Innanzitutto si può dimostrare che l'ordine p del metodo di Runge-Kutta così ottenuto è $m_c + 1$, dove m_c è l'ordine polinomiale della formula di quadratura interpolatoria costruita sui nodi $\mathbf{c}$. Poiché dalla teoria delle formule di quadratura interpolatorie risulta $m_c \geq \nu - 1$, si ha dunque $p \geq \nu$. In particolare, se $\mathbf{c}$ sono distribuiti come i nodi delle formule di Gauss-Legendre nell'intervallo [0,1], si ottengono le formule di collocazione gaussiana, che hanno ordine massimo pari a $p = 2\nu$.

Inoltre i metodi di collocazione forniscono una approssimazione uniforme in tutto l'intervallo $[t_n,t_n + h]$ della soluzione. Il polinomio di collocazione infatti si può ricostruire a partire dai valori dei k_i dalla espressione

$$p(t_n + \vartheta h) = y_n + h \sum_{i=1}^{\nu} w_i(\vartheta) k_i,$$

dove $w_i(\vartheta) = \int_0^{\vartheta} L_i(x)\, dx, i = 1, \ldots, \nu$. Per quanto riguarda l'ordine di accuratezza uniforme, si ha

$$\max_{t_n \leq t \leq t_n + h} \|y(t) - p(t)\| = \begin{cases} O(h^{\nu}) & \text{se} \quad p = \nu, \\ O(h^{\nu+1}) & \text{se} \quad p > \nu. \end{cases}$$

Poiché tale stima è valida per ogni intervallo, il metodo permette di ottenere una approssimazione uniforme della soluzione su tutto l'intervallo $[t_0,T]$. Citiamo un paio di metodi di collocazione gaussiana, particolarmente utilizzati nelle applicazioni. Si tratta del metodo *midpoint* e del metodo dei trapezi. Il primo si ottiene partendo dalla formula di quadratura del punto medio, ed è definito dai seguenti parametri:

$$\nu = 1, \quad a_{11} = 1/2, \quad c_1 = 1/2, \quad w_1 = 1, \tag{8.29}$$

mentre il secondo si ottiene a partire dalla formula dei trapezi, ottenendo

$$\nu = 2, \quad A = \begin{pmatrix} 0 & 0 \\ 1/2 & 1/2 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \tag{8.30}$$

Come vedremo, questi metodi, oltre ad essere i metodi di ordine massimo, presentano anche peculiari caratteristiche per quanto riguarda le loro proprietà di stabilità (vedi Sezione (8.7) sulla stabilità).

8.5 Controllo automatico del passo

Nella sezione precedente abbiamo approssimato la soluzione di una equazione differenziale suddividendo l'intervallo di integrazione in un certo numero di intervalli di uguale ampiezza. Tale approccio non sempre risulta il più conveniente in termini di rapporto costo/beneficio. Il passo della suddivisione potrebbe infatti risultare eccessivamente ampio in regioni di grande variazione della funzione, oppure inutilmente piccolo in regioni nelle quali la funzione varia molto lentamente. Come si è fatto con la quadratura adattiva, anche qui una strategia più efficace per la risoluzione delle equazioni differenziali consiste nel variare il passo di integrazione in funzione della variabilità della soluzione. Il modo di procedere è simile a quello

utilizzato per la quadratura adattiva. Dapprima facciamo vedere che se imponiamo un vincolo di accuratezza uniforme sull'errore locale di discretizzazione, questo si traduce in una stima sull'errore globale. Dopo vedremo come stimare l'errore locale e come, da questa stima, desumere il passo ottimale.

L'analisi per questo tipo di stima si effettua considerando l'errore locale di troncamento in una forma diversa da quella considerata prima. Più precisamente, sia

$$\sigma_{i+1} \equiv z_i(t_{i+1}) - y_{i+1},$$

dove $z_i(t)$ denota la soluzione esatta della equazione passante per il punto (t_i,y_i) (si noti che questa definizione di σ è diversa da quella utilizzata nello studio della convergenza). Supponiamo che l'intervallo di integrazione $[t_0,T]$ sia suddiviso in N intervalli mediante i punti $t_1,t_2,\ldots,t_N = T$, non necessariamente equispaziati. Se per ogni intervallo i vale la stima

$$\|\sigma_i\| \leq \varepsilon h_i, \quad i = 1,\ldots,N, \tag{8.31}$$

dove ε rappresenta una tolleranza prefissata, allora vale la seguente stima sull'errore globale

$$\|e_N\| \leq \varepsilon \sum_{i=1}^{N} \exp(L(T - t_i))h_i.$$

Il termine di destra rappresenta una somma integrale relativa alla quantità

$$\int_{t_0}^{T} \exp(L(T-t))\,dt = \frac{1}{L}(\exp(L(T-t_0) - 1). \tag{8.32}$$

Pertanto possiamo concludere che se l'errore locale di troncamento soddisfa la disequazione (8.31), allora l'errore globale soddisferà una disuguaglianza del tipo

$$\|e_N\| \leq K\varepsilon, \tag{8.33}$$

dove la costante K è data, in prima approssimazione, dalla espressione (8.32).

La stima appena ricavata, valida in generale per un qualunque sistema di equazioni differenziali, può essere raffinata utilizzando, invece della costante di Lipschitz L della funzione f, la cosiddetta costante di Lipschitz destra del sistema, che si solito è molto inferiore ad L. Tale costante si definisce in questo modo. Supponiamo che esista una costante M tale che il prodotto scalare fra la differenza fra la funzione f calcolata in due punti diversi (ma per lo stesso valore di t) e la differenza fra i due punti soddisfi la disuguaglianza:

$$(f(t,y) - f(t,z),y - z) \leq M(y - z,y - z) = M\|y - z\|_2^2, \tag{8.34}$$

dove, per ogni coppia di vettori di $\mathbb{R}^m$, y e z, intendiamo

$$(y,z) \equiv \sum_{i=1}^{m} y_i z_i.$$

Osserviamo che se la f soddisfa le ipotesi del teorema di Cauchy, allora la costante M esiste sempre ed è maggiorata da L. Infatti $(f(t,y) - f(t,z),y - z) \leq \|f(t,y) - f(t,z)\|\|y - z\| \leq L\|y - z\|^2$.

La costante M è quella che determina la velocità di allontanamento (o di avvicinamento!) di due traiettorie che partono da due punti distinti. Infatti si ha

$$\frac{d}{dt}\|y-z\|_2^2 = 2(f(t,y)-f(t,z),y-z) \leq 2M\|y-z\|_2^2.$$

Da questa disequazione differenziale segue, per il lemma di Gronwall[24, 27]

$$\|y-z\|_2^2 \leq \|y_0-z_0\|_2^2 \exp(2M(t-t_0)),$$

e quindi

$$\|y-z\|_2 \leq \|y_0-z_0\|_2 \exp(M(t-t_0)).$$

I sistemi per i quali vale una disuguaglianza del tipo (8.34) con $M < 0$ sono detti, in analogia con alcuni sistemi termomeccanici, *sistemi dissipativi*. Per essi le traiettorie tendono ad avvicinarsi.

Le tecniche di controllo del passo si basano sull'imporre la relazione (8.31) per ogni intervallo. Si agisce come segue.

Dato un passo h_0 di tentativo, si stima l'errore locale di troncamento relativo a tale passo. Esso potrebbe essere, ad esempio, il passo che è stato utilizzato al tempo precedente.

Per il momento non ci occupiamo di come determinare una stima dell'errore locale di troncamento. Diciamo solo che l'abbiamo calcolata in funzione del passo h_0: $\sigma = \sigma(t_n; h_0)$. Imponiamo che valga adesso la disuguaglianza (8.31). In particolare imponiamo che $\sigma(h_n) = \varepsilon h_n/2$. Ricordando l'espansione asintotica dell'errore di troncamento per un metodo di ordine p e trascurando i termini di ordine superiore, si ottiene per il nuovo passo l'equazione

$$\sigma(h_n) \approx Ch_n^{p+1} = \frac{1}{2}\varepsilon h_n,$$

da cui

$$h_n = \left(\frac{\varepsilon}{2C}\right)^{1/p}. \tag{8.35}$$

La costante C, a sua volta, si determina dalla relazione

$$\sigma(h_0) \approx Ch_0^{p+1},$$

e quindi, sostituendo nella equazione (8.35) si ha

$$h_n = h_0\left(\frac{\varepsilon h_0}{2\sigma(h_0)}\right)^{1/2}.$$

A causa delle approssimazioni effettuate (sulla stima di σ e sull'andamento asintotico di σ per piccoli valori di h) non è detto che il valore determinato in questo modo soddisfi la condizione (8.31). Inoltre non è desiderabile che tale condizione sia soddisfatta con un passo troppo piccolo. In pratica si verifica se il nuovo valore di h soddisfa la condizione

$$\frac{1}{4}\varepsilon h_n \leq \sigma(h_n) \leq \varepsilon h_n. \tag{8.36}$$

Se questa condizione viene soddisfatta il passo h_n è accettato come nuovo passo, altrimenti si pone $h_0 = h_n$ e si ripete il procedimento. Di solito, per evitarle cicli infiniti o troppo lunghi, si pone un vincolo sul massimo numero di cicli effettuati per la determinazione del valore ottimale di h. Osserviamo che se il valore iniziale di h_0 è particolarmente cattivo, la stima del rapporto tra il passo vecchio ed il passo nuovo potrebbe fornire valori eccessivamente alti o bassi. Per evitare che il rapporto h_n/h_0 sia troppo alto o troppo basso si pongono dei vincoli alla massima variabilità di tale rapporto.

Una considerazione a parte va fatta per il primo valore di h. Poiché non si dispone di un valore di tentativo "buono", si prova con un valore fornito dall'utente o con un valore di *default*, ad esempio $h = 0.1$, e si itera il procedimento per la determinazione diverse volte, con vincoli meno stringenti sul rapporto h_n/h_0, fino a determinare un valore di h_n che soddisfi la condizione (8.36).

Un esempio di applicazione della tecnica di controllo del passo su esposta è implementata nella function MATLAB `stepcontrol.m`, disponibile come molti altri script e funzioni al sito Web del libro.

Ci sono funzioni standard di MATLAB per l'integrazione di equazioni differenziali ordinarie che utilizzano tecniche di controllo del passo. In particolare, tali funzioni consentono di fornire due tipi di tolleranza, relativa ed assoluta. Siano ε_r ed ε_a rispettivamente le tolleranze relative ed assolute, e sia $\sigma^{(i)}$ l'errore locale di troncamento sulla componente i. Allora il controllo del passo è effettuato in modo da soddisfare la condizione

$$|\sigma^{(i)}| \leq \max(\varepsilon_r y^{(i)}, \varepsilon_a)h, \tag{8.37}$$

dove $y^{(i)}$ è la componente i del vettore soluzione. Di solito è preferibile imporre vincoli sull'errore relativo, ma in alcuni casi, come ad esempio quello in cui una componente del vettore è molto piccola, il vincolo sull'errore relativo diventa troppo restrittivo.

Torniamo adesso al problema di determinare una stima dell'errore locale di troncamento. Esattamente come nel caso delle stime dell'errore utilizzate per la costruzione di formule di quadratura adattiva, anche in questo caso esistono due approcci possibili per stimare l'errore locale. Un primo approccio consiste nel calcolare, data la soluzione al passo n, y_n, la soluzione numerica al passo $n+1$ mediante due metodi numerici, ad esempio uno di ordine p ed uno di ordine $p+1$. Un secondo approccio consiste nell'ottenere una soluzione numerica al tempo $t_n + h$ mediante un solo passo o mediante più passi (ad esempio due) dello stesso metodo di ordine p, e di utilizzare poi una tecnica di estrapolazione per stimare l'errore. Consideriamo in dettaglio il primo approccio, rimandando per il secondo alla letteratura [27]. Per il primo tipo di tecnica si utilizzano una coppia di metodi Runge-Kutta, di ordine p e $p+1$ che utilizzano le stesse valutazioni di funzione. Questo viene ottenuto costruendo una formula di ordine $p+1$ in modo che i valori delle k_i possano poi essere combinati in modo da ottenere un metodo di ordine p. Coppie di metodi Runge-Kutta con queste caratteristiche sono i metodi di Runge-Kutta-Fehlberg. I più comuni sono quelli che utilizzano metodi di ordine 2 e 3 (RKF 2-3) e quelli che utilizzano metodi di ordine 4 e 5 (RKF 4-5) riportati in Tabella 8.3.

0	0	0	0	0	0	0
$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	0
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$	0	0	0	0
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$	0	0	0
1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	0	0
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	0
RKF4	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$	0
RKF5	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$

Tabella 8.3 Metodi di Runge-Kutta-Fehlberg di ordine 4 e 5.

Posto y_{n+1} e y^*_{n+1}, rispettivamente, la soluzione numerica ottenuta con il metodo di ordine p ed il metodo di ordine $p+1$, con passo h, si pone

$$\sigma(h) = y^*_{n+1} - y_{n+1}.$$

Naturalmente, una volta accettato il passo di integrazione si utilizzerà il valore y^*_{n+1} per la soluzione numerica che è più accurato. Si otterrà quindi un risultato numerico considerevolmente più accurato di quello previsto dalla stima (8.33).

Come esempio di controllo del passo mostriamo, nella Figura (8.6), una soluzione numerica ottenuta con la tecnica riportata nel programma `stepcontrol.m`, che utilizza la tecnica di controllo del passo su esposta, e la soluzione numerica dello stesso problema ottenuta con i due integratori del MATLAB, `ode45` e `ode15s`. Il primo è sempre basato su una coppia di metodi Runge-Kutta, sviluppata da Dormand e Prince [16], ed utilizza una tecnica più sofisticata di controllo del passo, mentre il secondo utilizza un metodo implicito multistep, a passo variabile e ad ordine variabile [51]. Come si vede `ode15s` richiede un numero inferiore di passi di integrazione rispetto agli altri metodi per integrare lo stesso problema. Questo perché il problema è *stiff*, e `ode15s` utilizza un metodo implicito. Il problema della stiffness verrà trattato in dettaglio nella Sezione (8.7).

Le soluzioni ottenute con i tre codici, `stepcontrol`, `ode45`, e `ode15s`, sono sovrapposte nella prima figura e risultano indistinguibili. Tale differenza tra il numero di passi richiesti dai vari metodi risulta eclatante per valori maggiori del parametro μ (vedi Fig. 8.6).

8.6 Metodi multistep

I metodi Runge-Kutta rientrano nella categoria dei metodi ad un passo, in quanto l'approssimazione numerica della funzione al tempo $t+h$ dipende solo dalla soluzione numerica al tempo t.

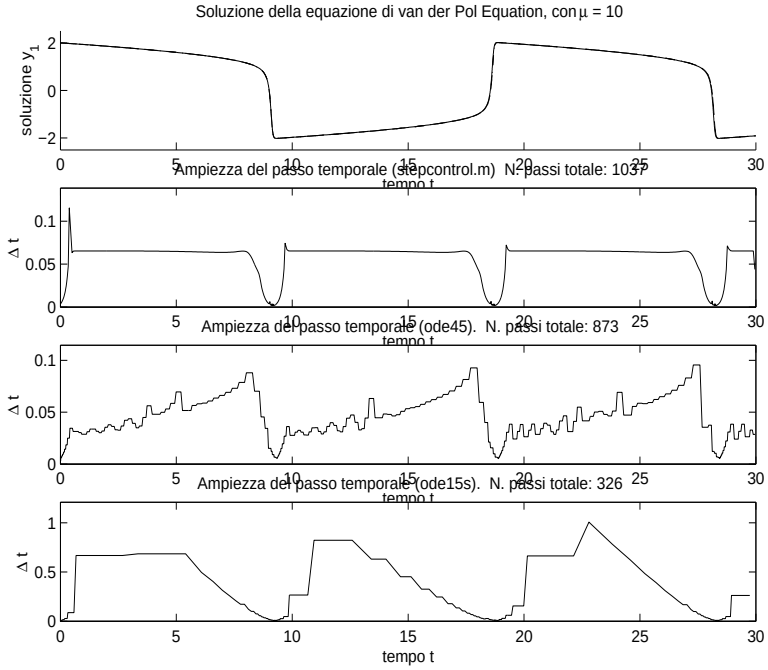

Figura 8.6 Soluzione numerica delle equazioni dell'oscillatore di van der Pol.

È possibile definire altri metodi che invece utilizzano anche valori numerici della funzione y calcolati in tempi precedenti. La deduzione di questa classe di metodi, detti metodi multistep, parte dalle formule di quadratura.

Supponiamo che l'intervallo di integrazione $[t_0,T]$ sia suddiviso in un certo numero di intervalli di ampiezza h. Integrando l'equazione da t_{n-j} a t_{n+k} si ottiene

$$y(t_{n+k}) = y(t_{n-j}) + \int_{t_{n-j}}^{t_{n+k}} f(t,y(t))\,dt.$$

Per trasformare questa relazione esatta in un metodo numerico possiamo applicare una formula di quadratura per il calcolo dell'integrale. Poiché il valore numerico della funzione $y(t)$ è noto nei punti equidistribuiti della griglia, appare naturale utilizzare per il calcolo dell'integrale le formule di Newton-Cotes. Utilizzando allora $q+1$ punti, t_{n-q}, t_{n-q+1}, $\ldots$,t_n, costruiamo il polinomio di Lagrange, ed integriamolo nell'intervallo $[t_{n-j},t_{n+k}]$.

$$p_q(t) = \sum_{i=0}^{q} f(t_{n-i},y_{n-i})L_i(t),$$

dove $L_i(x)$ sono i polinomi elementari di Lagrange definiti nel capitolo sull'inter-

polazione[3],

$$L_i(t) = \prod_{\substack{l=0 \\ l \neq i}}^{q} \frac{t - t_{n-l}}{t_{n-i} - t_{n-l}}.$$

Sostituendo il polinomio nell'integrale, si ottiene la seguente relazione fra i valori delle y_i

$$y_{n+k} = y_{n-j} + h \sum_{i=0}^{q} \beta_{qi} f_{n-i}, \tag{8.38}$$

dove, per compattezza di notazione, si è posto

$$f_l = f(t_l, y_l),$$

e dove i coefficienti β_{qi} si ottengono per integrazione dei polinomi elementari di Lagrange

$$\beta_{qi} \equiv \frac{1}{h} \int_{t_{n-j}}^{t_{n+k}} L_i(t)\, dt = \int_{-j}^{k} \prod_{\substack{l=0 \\ l \neq i}}^{q} \frac{x + l}{-i + l}\, dx.$$

A seconda dei diversi valori di k, j, e q si ottengono differenti metodi multistep. In particolare, per $k = 1$ e $j = 0$ si ottengono i cosiddetti metodi di Adams-Bashforth.

$$y_{n+1} = y_n + h \sum_{i=0}^{q} \beta_{qi} f_{n-i}. \tag{8.39}$$

I coefficienti di questi metodi si possono agevolmente trovare utilizzando la funzione MATLAB `AdamsBashforth`, della quale riportiamo il listato

```
function beta = AdamsBashforth(q)
% Sintassi beta = AdamsBashforth(q)
% Restituisce i valori dei coefficienti
% del metodo di Adams-Bashforth
% di ordine q+1
syms b L x
for i=0:q
    L = [0:i-1,i+1:q];
    b = prod((L+x)./(L-i));
    beta(i+1) = int(b,0,1);
end
```

Tali coefficienti sono utilizzati dalla seguente funzione `AB` che implementa il corrispondente metodo di Adams-Bashforth

```
function [t,y] = AB(fname,t0,y0,N,h);
% Sintassi: [t,y] = AB(fname,t0,y0,N,h)
```

[3] Si noti che è stata usata una convenzione diversa da quella utilizzata nel capitolo 3 sulla interpolazione.

```
%
% Implementa il metodo di Adams-Bashforth
% I parametri del metodo sono memorizzati nel vettore beta
% che e' definito come variabile globale, e puo' essere
% calcolato, ad esempio, con la funzione AdamsBashforth.m
% I dati iniziali sono forniti dai vettori t0 [1,q+1] e y0 [m,q+1].
% h rappresenta l'ampiezza del passo, ed N il numero di passi.
% In uscita la funzione restituisce i vettori [t,y].
% t ha dimensione (N+1,1)
% y ha dimensione (N+1,m) e contiene il vettore delle soluzioni
%
global beta
if length(t0) == length(beta)
  q = length(t0)-1;
  f = fcnchk(fname);
  [n,m] = size(y0);
  t=zeros(N,1);
  y = zeros(N,m);
  for k=1:q+1
    Y = y0(k,:)';
    F(k,:) = feval(f,t0(k),Y)';
    y(k,:) = y0(k,:);
    t(k,:) = t0(k,:);
  end
  for k=q+1:N-1
    y(k+1,:) = y(k,:) + beta*F*h;
    Y = y(k+1,:)';
    FF = feval(f,t(k+1),Y)';
    F = [F(2:q+1,:);FF];
    t(k+1) = t(k) + h;
  end
else
  disp(['Errore nelle dimensioni: length(beta) = '...
   num2str(length(beta)) ', length(t0)  = ' num2str(length(t0))])
end
```

Per $k = 0$ e $j = 1$ si ottengono le formule di Adams-Moulton

$$y_n = y_{n-1} + h \sum_{i=0}^{q} \beta_{qi} f_{n-i}.$$

È più conveniente riscrivere le formule nella forma

$$y_{n+1} = y_n + h \sum_{i=0}^{q} \beta_{qi} f_{n-i+1}. \tag{8.40}$$

I coefficienti del metodo si possono agevolmente determinare utilizzando la funzione MATLAB `AdamsMoulton.m` disponibile al sito Web del libro.

Infine, scegliendo $k = 1$ e $j = 1$ si ottengono i metodi di Nyström

$$y_{n+1} = y_{n-1} + h \sum_{i=0}^{q} \beta_{qi} f_{n-i}. \tag{8.41}$$

I coefficienti β_{qi} si possono determinare con la funzione `Nystrom.m` disponibile al sito Web del libro.

Per $q = 0$ si ottiene il metodo del punto medio

$$y_{n+1} = y_{n-1} + 2hf(t_n,y_n).$$

Osserviamo che i metodi di Adams-Bashforth sono metodi espliciti, mentre i metodi di Adams-Moulton sono impliciti, in quanto il valore della funzione incognita al passo $n + 1$ compare sia alla sinistra che alla destra della equazione di ricorrenza.

Una tecnica che appare naturale utilizzare per la risoluzione della equazione non lineare per y_{n+1} è il metodo della iterazione funzionale.

Data una soluzione di tentativo per il valore di y_{n+1}, chiamiamola $y_{n+1}^{(0)}$, si determina una successione della forma

$$y_{n+1}^{(k+1)} = y_n + h\sum_{i=1}^{q} \beta_{qi} f_{n-i+1} + h\beta_{q0} f(t_{n+1},y_{n+1}^{(k)}), \quad k = 0,1,\ldots \tag{8.42}$$

Per valori sufficientemente piccoli del parametro h la convergenza è garantita, poiché la costante di Lipschitz della funzione di iterazione è hL, dove L è la costante di Lipschitz della funzione f.

Per minimizzare il numero di iterazioni si cerca di scegliere un valore iniziale che sia una buona approssimazione della soluzione. Tale valore si può ottenere mediante un metodo di Adams-Bashforth o da un metodo di Nyström. I metodi così formati vengono chiamati metodi *predictor-corrector*.

Normalmente si effettuano solo uno o due passi del corrector. L'obiettivo principale, infatti, non è quello di guadagnare in accuratezza, ma in stabilità.

I metodi multistep, specialmente nella forma dei metodi predictor-corrector, sono molto utilizzati nella pratica, poiché sono particolarmente efficienti in quanto richiedono poche valutazioni di funzione per avanzare di un passo temporale, e permettono di raggiungere ordini molto elevati.

Risultano tuttavia poco adatti quando i requisiti di stabilità sono preponderanti rispetto a quelli di accuratezza o quando la funzione f non è molto regolare.

L'argomento della stabilità sarà trattato nella sezione successiva.

Metodi multistep lineari, LMM I metodi multistep che abbiamo visto sono stati ottenuti approssimando l'integrale della funzione f mediante una formula di quadratura. Una famiglia di schemi diversi si può ottenere come segue. Supponiamo di conoscere una approssimazione della funzione nei punti y_k, $k = n - q + 1,\ldots,n + 1$. Allora è possibile determinare un polinomio di grado q che interpoli gli $n + 1$ punti dati. Con tale polinomio è possibile calcolare una approssimazione della derivata della funzione in uno dei nodi t_k. A tale scopo basta infatti derivare il polinomio, e calcolare la derivata nel nodo t_k. Se la funzione $y(t)$, supposta sufficientemente regolare, nei nodi è conosciuta con una accuratezza almeno $O(h^{q+1})$ allora la derivata del polinomio approssima la derivata della funzione nel nodo con un errore $O(h^q)$. Un metodo numerico per la equazione differenziale si ottiene imponendo che la derivata del polinomio sia uguale alla funzione f. Più precisamente, sia P il polinomio di interpolazione nei nodi (t_ℓ,y_ℓ),

$\ell = n - q + 1, \ldots, n + 1$; imponiamo

$$P'(t_k) = f(t_k, y_k),$$

dove di solito si prende $k = n$ (ottenendo un metodo esplicito) oppure $k = n + 1$ (ottenendo un metodo implicito). Le derivate del polinomio in un punto sono ottenute mediante una combinazione dei valori della funzione. I coefficienti di tale combinazione sono le derivate dei coefficienti del corrispondente polinomio elementare di Lagrange [50]. Essi assumono dunque la forma

$$\sum_{i=0}^{q} \alpha_i^e \, y_{n-i+1} = h f_n, \tag{8.43}$$

nel caso dei metodi espliciti, e

$$\sum_{i=0}^{q} \alpha_i^i \, y_{n-i+1} = h f_{n+1}, \tag{8.44}$$

nel caso dei metodi impliciti. Questi metodi sono detti BDF (Backward Differentiation Formula). Osserviamo che il metodo BDF esplicito per $q = 1$ coincide con il metodo di Eulero esplicito, mentre quello per $q = 2$ coincide con il metodo del punto medio. Per $q = 3$ si ottiene il metodo

$$\frac{1}{3} y_{n+1} + \frac{1}{2} y_n - y_{n-1} + \frac{1}{6} y_{n-2} = h f_n.$$

Tale metodo risulta tuttavia instabile (vedi sezione successiva), e quindi di scarsa utilità pratica. I coefficienti α^i dei metodi BDF impliciti fino a $q = 6$ sono riportati nella Tabella (8.6). I metodi BDF impliciti con $q > 6$ risultano instabili.

Ordine q	α_0	α_1	α_2	α_3	α_4	α_5	α_6
1	1	-1					
2	3/2	-2	1/2				
3	11/6	-3	3/2	-1/3			
4	25/12	-4	3	-4/3	1/4		
5	137/60	-5	5	-10/3	5/4	-1/5	
6	147/60	-6	15/2	-20/3	15/4	-6/5	1/6

Tabella 8.4 Coefficienti dei metodi BDF impliciti.

Una generalizzazione dei metodi visti precedentemente è costituita dalla famiglia dei metodi multistep lineari (*Linear Multistep Methods* o LMM), che include come caso particolare i metodi di Adams ed i metodi BDF. Il generico metodo LMM ha la forma

$$\sum_{i=0}^{q} \alpha_i y_{n-i} = h \sum_{i=0}^{q} \beta_i f_{n-i}. \tag{8.45}$$

Un metodo LMM è dunque caratterizzato da due polinomi,

$$\varrho(\vartheta) = \sum_{i=0}^{q} \alpha_i \vartheta^{q-i}, \quad S(\vartheta) = \sum_{i=0}^{q} \beta_i \vartheta^{q-i}. \tag{8.46}$$

Un modo compatto per scrivere il metodo multistep è quello di utilizzare l'operatore di shift E, definito da

$$Ey_n \equiv y_{n+1}. \tag{8.47}$$

Osserviamo che $E^2 y_n = EEy_n = Ey_{n+1} = y_{n+2}$ e, in generale, $E^k y_n = y_{n+k}$. Utilizzando i polinomi caratteristici e l'operatore di shift, il generico metodo LMM può essere scritto nella forma

$$\mathcal{L}_h y_n \equiv \varrho(E) y_n - hS(E) f_n = 0. \tag{8.48}$$

Utilizzeremo tali polinomi nello studio della stabilità dei metodi LMM.

Anche per i metodi multistep è possibile definire i concetti di convergenza e di consistenza.

Definizione 8.4 *Un metodo multistep si dice convergente nel punto $t \in [t_0,T]$ se $y_n \to y(t)$ per $n \to \infty$, $h \to 0$, $nh = t - t_0$. Il metodo si dirà convergente in $[t_0,T]$ se risulta convergente $\forall t \in [t_0,T]$.*

Se applichiamo l'operatore discreto che definisce un metodo multistep ad una soluzione esatta della equazione differenziale non otterremo zero, ma otterremo una piccola quantità, che viene chiamata errore locale di troncamento.

Definizione 8.5 *Per un metodo multistep lineare si dice errore locale di troncamento la quantità*

$$\sigma(t_n; h) = \mathcal{L}_h y(t_n) = \varrho(E) y(t_n) - hS(E) f(t_n, y(t_n)).$$

In maniera analoga a quanto fatto per i metodi ad un passo, si può definire l'errore locale di discretizzazione $d(t; h)$ come l'errore locale di troncamento diviso per h. Infine, diamo la definizione di consistenza:

Definizione 8.6 *Un metodo LMM si dirà consistente, di ordine di consistenza p se, quando applicato ad una soluzione della equazione sufficientemente regolare, l'errore di discretizzazione è un infinitesimo di ordine p in h.*

Anche per i metodi multistep lineari è possibile fornire un teorema generale di convergenza.

Naturalmente anche per i codici multistep è possibile disegnare delle strategie di controllo del passo. Il problema è più complesso rispetto al caso dei metodi Runge-Kutta poiché i multistep si basano pesantemente sull'utilizzo della soluzione numerica nei passi precedenti, supposti equispaziati. Un cambiamento di passo quindi richiede la valutazione del campo y in punti intermedi, cosa che si può effettuare ricorrendo, ad esempio, alla interpolazione polinomiale.

8.7 Problemi *stiff* e stabilità

Nelle precedenti sezioni abbiamo visto che esistono metodi espliciti ed impliciti. Questi ultimi sono di solito molto più onerosi da utilizzare per un passo. Eppure

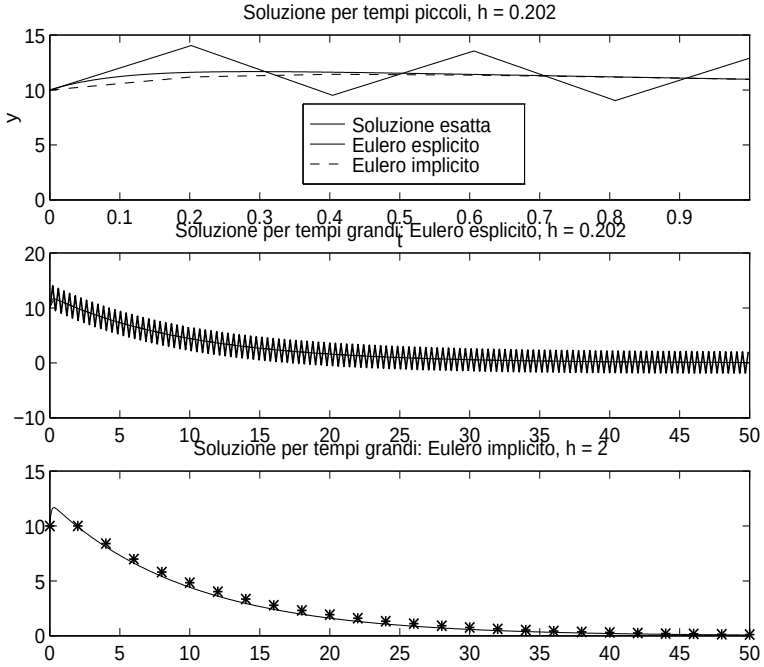

Figura 8.7 Soluzione esatta e numerica dell'equazione (8.49): (a) soluzione per tempi piccoli; (b) Eulero esplicito con passo $h = 0.202$ e soluzione esatta per tempi grandi; (c) soluzione esatta ed Eulero implicito (*) con $h = 2$.

esistono dei casi nei quali metodi impliciti sono decisamente preferibili rispetto a metodi espliciti. Questo si verifica nei cosiddetti sistemi *stiff*. Per comprendere meglio il concetto di stiffness, cominciamo con un semplice esempio. Consideriamo l'equazione

$$y'' + 10y' + y = 0, \quad y(0) = 10, \quad y'(0) = 20. \tag{8.49}$$

La soluzione esatta può essere calcolata esplicitamente. Essa è data da

$$y(t) = C_1 \exp(\lambda_1 t) + C_2 \exp(\lambda_2 t),$$

dove λ_1 e λ_2 sono le radici della equazione caratteristica $\lambda^2 + 10\lambda + 1 = 0$, e le costanti C_1 e C_2 sono determinate imponendo le condizioni iniziali.

Proviamo a risolvere l'equazione con il metodo di Eulero esplicito e con il metodo di Eulero implicito. Le soluzioni (esatta e numeriche) sono riportate nella Figura (8.7)

Si vede che mentre il metodo di Eulero implicito è in grado di catturare le caratteristiche fondamentali della soluzione, il metodo di Eulero esplicito fornisce una soluzione oscillante. Se si prova ad aumentare il passo si osservano oscillazioni di ampiezza crescente, mentre per passi inferiori si vede che le oscillazioni decrescono, e per passi sufficientemente piccoli svaniscono del tutto.

Cosa sta succedendo? La soluzione esatta è composta dalla somma di due termini esponenziali con costanti molto diverse, infatti $\lambda_1 \approx -9.9$ e $\lambda_2 \approx -0.1$. Il termine relativo a λ_1 decade molto rapidamente, e la soluzione esatta dell'equazione, dopo un tempo assai breve, è data dall'esponenziale che decade lentamente. Per ottenere una soluzione accurata ci si aspetta che sia necessario prendere un passo temporale piccolo rispetto ai tempi caratteristici in cui varia la soluzione. A parte un breve transitorio iniziale, dunque, dovrebbe essere sufficiente usare dei passi temporali piccoli rispetto a $1/|\lambda_2| \approx 10$. In effetti questo è quanto si osserva utilizzando il metodo di Eulero implicito. Per capire perché il metodo esplicito non funziona, riscriviamo l'equazione come sistema del primo ordine

$$Y' = AY, \quad Y = \begin{pmatrix} y \\ y' \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ -1 & -10 \end{pmatrix}. \tag{8.50}$$

Il metodo di Eulero esplicito applicato a tale sistema diventa

$$Y_{n+1} = Y_n + hAY_n = (I + hA)Y_n = (I + hA)^{n+1}Y_0.$$

Mentre la soluzione esatta del sistema (8.50) tende a zero per $t \to \infty$, la soluzione numerica Y_n tende a zero o a infinito, a seconda del raggio spettrale della matrice $I + hA$. Gli autovalori di tale matrice si calcolano a partire dagli autovalori della matrice A. Infatti, gli autovalori di hA sono $h\lambda_1$ e $h\lambda_2$, e gli autovalori di $I + hA$ sono $1+h\lambda_1$ e $1+h\lambda_2$. La condizione affinché il raggio spettrale sia minore di uno è dunque $h < -2/\lambda_1$. Si vede quindi che, affinché la soluzione numerica ottenuta con il metodo di Eulero esplicito decada, occorre che il passo temporale sia minore di una quantità proporzionale alla più piccola costante di tempo presente nel sistema.

Applicando il metodo di Eulero implicito si ottiene invece

$$(I - hA)Y_{n+1} = Y_n, \Rightarrow Y_{n+1} = (I - hA)^{-1}Y_n = (I - hA)^{-(n+1)}Y_0.$$

Gli autovalori della matrice $(I - hA)^{-1}$ sono uguali a $1/(1 - h\lambda_1)$ e $1/(1 - h\lambda_2)$ e sono entrambi minori di 1 (anche in valore assoluto, poiché sono positivi) e quindi la soluzione numerica decade per qualunque valore di h.

Sistemi nei quali la soluzione generica varia su tempi scala molto diversi, ed i termini corrispondenti alle scale temporali piccole decadono, sono detti sistemi *stiff*.

Ci possono essere altre definizioni di *stiffness*, ma quella qui utilizzata è forse la più comune.

Il risultato che abbiamo visto per l'esempio particolare ha carattere abbastanza generale. Quando ci si trova davanti a equazioni di tipo *stiff*, i metodi espliciti soffrono di restrizioni sul passo temporale dovute alla stabilità che sono molto più stringenti dei requisiti richiesti da obiettivi di accuratezza. In questi casi è di solito preferibile ricorrere a schemi impliciti, che sono utilizzati proprio in virtù delle loro migliori proprietà di stabilità.

Per studiare la stabilità dei metodi si considerano delle equazioni modello, e si studia il comportamento delle soluzioni dei metodi numerici su tali equazioni confrontandone il comportamento con quello delle soluzioni esatte.

Consideriamo due definizioni di stabilità: la zero stabilità e la A-stabilità. Prima di addentrarci nello studio della stabilità premettiamo la definizione di metodo numerico lineare. Consideriamo un sistema di equazioni differenziali lineari, $Y' = AY$, e denotiamo con $\hat{Y}_n(Y_0)$ la soluzione numerica ottenuta con un certo

metodo, corrispondente alla condizione iniziale Y_0. Diremo che il metodo è lineare se

$$\hat{Y}_n(\alpha Y_0 + \beta Z_0) = \alpha \hat{Y}_n(Y_0) + \beta \hat{Y}_n(Z_0).$$

In pratica, tutti i metodi che abbiamo considerato finora sono lineari.

Definizione 8.7 *Un metodo numerico lineare si dice* zero-stabile *se la soluzione numerica della equazione*

$$y' = 0, \tag{8.51}$$

è stabile, cioè se per ogni dato iniziale esiste una costante C tale che $|y_n| < C \; \forall n$

Tutti i metodi Runge-Kutta risultano banalmente zero-stabili, in quanto la soluzione numerica della equazione (8.51) è $y_n = y_0$, e quindi una eventuale perturbazione del dato iniziale non viene amplificata.

Per quanto riguarda i metodi multistep lineari, le proprietà di zero-stabilità dipendono dalle proprietà di stabilità della equazione alle differenze

$$\sum_{i=0}^{q} \alpha_i y_{n-i} = 0. \tag{8.52}$$

Tale equazione è detta equazione alle differenze lineare omogenea di ordine q, a coefficienti costanti. La teoria delle equazioni alle differenze è analoga alla teoria delle equazioni differenziali. In particolare, una soluzione di tale equazione si ottiene cercando una soluzione del tipo $y_n = z^n$. Inserendo tale *ansatz* nella equazione (8.52) si ottiene l'equazione algebrica

$$\sum_{i=0}^{q} \alpha_i z^{n-i} = 0. \tag{8.53}$$

Pertanto z^n è soluzione della equazione alle differenze (8.52) se z è una radice del polinomio caratteristico $\varrho(z)$. Se q radici di tale polinomio sono tutte distinte, allora la soluzione generale della equazione alle differenze (8.52) si scrive

$$y_n = \sum_{i=1}^{q} C_i z_i^n.$$

Nel caso in cui la radice z_i abbia molteplicità m_i, si può dimostrare che oltre a z_i^n, anche nz_i^n, $n^2 z_i^n$, $\ldots$,$n^{m_i-1} z_i^n$ sono soluzioni della equazione alle differenze (8.52). Pertanto, la soluzione generale di una equazione alle differenze omogenea a coefficienti costanti del tipo (8.52) è data da

$$y_n = \sum_{i=1}^{k} u_i(n) z_i^n,$$

dove le k radici z_i,$i = 1,\ldots,k$ hanno molteplicità m_i, con $\sum_{i=1}^{k} m_i = q$, e $u_i(n)$ sono polinomi di grado $m_i - 1$.

Le proprietà di 0 stabilità del metodo dipendono quindi dalle radici del polinomio $\varrho(z)$. Se tutte le radici hanno modulo minore di 1, la soluzione generale di tale equazione tende a zero per $n \to \infty$, e quindi il metodo numerico è sicuramente zero-stabile. Se qualche radice ha modulo unitario, allora occorre considerare anche

la sua molteplicità. Se questa è 1 allora la soluzione della equazione alle differenze si mantiene limitata ed il metodo è ancora stabile. Se, viceversa, c'è almeno una radice con modulo maggiore di 1, o se c'è almeno una radice multipla con modulo unitario, allora il metodo numerico non è zero stabile.

La condizione di 0 stabilità è una condizione necessaria per la convergenza. Un metodo che non sia almeno 0 stabile risulta di scarsa utilità pratica. Tutti i metodi multistep che abbiamo considerato in questa sezione risultano zero stabili.

Citiamo un teorema generale sulla convergenza dei metodi multistep:

Teorema 8.1 *Condizione necessaria e sufficiente affinché un metodo multistep lineare sia convergente è che sia consistente e zero stabile.*

8.7.1 A-stabilità

Un'altra definizione di stabilità molto importante nella pratica è quella di A-stabilità. In questo caso si considera come equazione modello la singola equazione lineare

$$y' = \lambda y, \quad y(0) = 1, \tag{8.54}$$

con $y : \mathbb{R} \to \mathbb{C}$, $\lambda \in \mathbb{C}$, $\mathbf{Re}\,\lambda < 0$. La soluzione esatta di questa equazione è $y = \exp(\lambda t)$. Se la parte reale di λ è negativa, il modulo della y decresce esponenzialmente nel tempo. Nello studio della stabilità si cerca di vedere se anche la soluzione numerica decresce nel tempo.

Lo studio del comportamento di un metodo numerico sulla singola equazione complessa fornisce informazioni più generali di quanto possa apparire a prima vista. In particolare, si ottengono informazioni sul comportamento del metodo per sistemi di equazioni lineari. Consideriamo infatti un sistema lineare della forma

$$Y' = AY, \tag{8.55}$$

dove $Y \in \mathbb{R}^m, A \in \mathbb{R}^{m\times m}$. Se la matrice A è diagonalizzabile, il sistema può essere ridotto ad un sistema diagonale mediante una trasformazione ortogonale. Detta Q la matrice ortogonale avente per colonne gli m autovettori di A, si ha infatti

$$\Lambda = Q^{-1}AQ, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m),$$

dove i λ_i sono gli autovalori di A. Posto dunque $Y = QC$, il sistema (8.55) si riduce ad un sistema diagonale per le incognite $C = (c_1, \ldots, c_m)^T$. Il sistema si può infatti riscrivere come

$$QC' = Q\Lambda C.$$

Le soluzioni per le c_i si scrivono

$$c_i = c_i^{(0)} \exp(\lambda_i t), \quad i = 1, \ldots, m,$$

o, in forma vettoriale,

$$C = \exp(\Lambda t)C_0 = \exp(\Lambda t)Q^{-1}Y_0,$$

dove abbiamo indicato simbolicamente con $\exp(\Lambda t)$ una matrice diagonale che ha per elementi $\exp(\lambda_i t), i = 1, \ldots, m$. Utilizzando questa espressione per C, la

soluzione del sistema lineare di partenza può essere scritta come

$$Y = Q\exp(-\Lambda t)Q^{-1}Y_0.$$

Se applichiamo un metodo lineare al sistema lineare (8.55), la soluzione numerica che si ottiene è la stessa che si otterrebbe applicando il metodo alla equazione nella forma diagonale, ed operando poi la trasformazione $Y = QC$. Scriviamo questa relazione formalmente per i metodi ad un passo. Un metodo ad un passo lineare, applicato ad un sistema lineare ha la forma

$$Y_1 = Y_0 + h\mathcal{L}_h AY_0 = Y_0 + h\mathcal{L}_h Q\Lambda Q^{-1}Y_0 = Q(C_0 + h\mathcal{L}_h \Lambda C_0),$$

quindi lo studio del comportamento del metodo per equazioni lineari complesse permette di stabilire le proprietà di stabilità del metodo anche per sistemi lineari[4].

Analisi della A-stabilità Applichiamo il generico metodo di Runge-Kutta alla equazione (8.54). Si ottiene la soluzione numerica dopo un passo

$$\begin{aligned} k_i &= \lambda(1 + h\sum_{j=1}^{\nu} a_{ij}k_j), \\ y_1 &= 1 + h\sum_{i=1}^{\nu} w_i k_i, \end{aligned}$$

La soluzione può essere più convenientemente scritta in forma vettoriale introducendo il vettore $e = (1,1,\ldots,1)^T \in \mathbb{R}^\nu$, ed il vettore $K = (k_1,\ldots,k_\nu)^T$

$$\begin{aligned} K &= \lambda(e + hAK), \\ y_1 &= 1 + hw^T K. \end{aligned}$$

Risolvendo per hK si ha

$$hK = z(I - zA)^{-1}e,$$

dove $z = h\lambda$. Sostituendo nella espressione per y_1 si ottiene

$$y_1 = R(z) = 1 + zw^T(I - zA)^{-1}e. \tag{8.56}$$

Si può dimostrare che tale espressione rappresenta una funzione razionale in z, e precisamente che

$$R(z) = \frac{\det(I - zA + zew^T)}{\det(I - zA)}. \tag{8.57}$$

La dimostrazione dell'espressione per $R(z)$ si ottiene facilmente mediante una diversa rappresentazione dei metodi Runge-Kutta. Questi si possono scrivere infatti nella forma

$$Y_i = y_0 + h\sum_{j=1}^{\nu} a_{ij}f(Y_j),$$

[4]Una analisi più accurata mostra che lo studio della stabilità per equazioni scalari caratterizza il comportamento delle soluzioni di sistemi di equazioni differenziali lineari anche nel caso in cui la matrice A non sia diagonalizzabile

$$y_1 = y_0 + h\sum_{i=1}^{\nu} w_i f(Y_i).$$

Applicando il metodo in questa forma alla equazione (8.54) si ottiene, con notazione vettoriale,

$$\begin{aligned} Y &= e + zAY, \\ y_1 &= 1 + zw^T Y. \end{aligned}$$

Queste equazioni rappresentano un sistema lineare di $\nu+1$ equazioni nelle incognite $(Y; y_1)$, precisamente

$$\begin{pmatrix} I - zA & 0 \\ -zw^T & 1 \end{pmatrix} \begin{pmatrix} Y \\ y_1 \end{pmatrix} = \begin{pmatrix} e \\ 1 \end{pmatrix}.$$

Utilizzando la regola di Cramer, è possibile risolvere il sistema per l'ultima incognita y_1 ottenendo

$$y_1 = \frac{\begin{vmatrix} I - zA & e \\ -zw^T & 1 \end{vmatrix}}{\begin{vmatrix} I - zA & 0 \\ -zw^T & 1 \end{vmatrix}}.$$

Per calcolare il determinante del numeratore della ultima espressione, sottraiamo a ciascuna delle prime ν righe l'ultima riga. Otteniamo così l'espressione

$$\begin{vmatrix} I - zA & e \\ -zw^T & 1 \end{vmatrix} = \begin{vmatrix} I - zA + zew^T & 0 \\ -zw^T & 1 \end{vmatrix} = |I - zA + zew^T|.$$

Si ottiene pertanto per y_1 l'espressione

$$y_1 = R(z) = \frac{\det(I - zA + zew^T)}{\det(I - zA)},$$

che rappresenta evidentemente una funzione razionale in z.

Tale funzione razionale viene detta *funzione di assoluta stabilità del metodo.* La soluzione numerica del metodo, dopo n passi, è data da

$$y_n = R(z)^n,$$

pertanto se $|R(z)| < 1$ si ha $y_n \to 0$, mentre se $|R(z)| > 1$ allora $y_n \to \infty$. La funzione $R(z)$ dunque caratterizza le zone in cui la soluzione numerica della equazione (8.54) decresce. I punti del piano complesso per i quali $|R(z)| < 1$ costituiscono la *regione di assoluta stabilità.*

Definizione 8.8 *Si dice regione si Assoluta stabilità la regione* $\{S_A \subset \mathbb{C} : \exists (I - zA)^{-1}$, e $|R(z)| \leq 1\}$.

Tanto più ampia sarà la regione di stabilità, tanto più stabile sarà il metodo numerico. Un metodo che fornisce una soluzione numerica che decresce quando la soluzione esatta decresce è detto A-stabile. Più precisamente

Definizione 8.9 *Un metodo numerico viene detto A-stabile se la regione di stabilità comprende il semipiano* $\mathbb{C}_- \equiv \{z \in \mathbb{C} : \mathbf{Re}\, z \leq 0\}$.

Osserviamo immediatamente che un metodo di Runge-Kutta esplicito non può essere A-stabile. Per un metodo esplicito, infatti, la matrice dei coefficienti A è una matrice triangolare inferiore con elementi nulli sulla diagonale. Il denominatore della espressione (8.57) vale dunque 1 e la funzione $R(z)$ è un polinomio. Poiché un polinomio diverge per $z \to \infty$, la regione in cui $|R(z)| < 1$ è necessariamente limitata.

Osserviamo inoltre che, se un metodo è di ordine p, allora la funzione di assoluta stabilità ha in comune con la soluzione esatta della equazione (8.54), $y(h) = e^{h\lambda} = e^z$ i primi p termini dello sviluppo di Taylor, e quindi

$$R(z) = 1 + z + \frac{z^2}{2} + \ldots + \frac{z^p}{p!} + o(z^p), \quad z \to 0.$$

Se poi consideriamo un metodo di Runge-Kutta esplicito a ν livelli con $p = \nu$, allora la sua funzione di assoluta stabilità è data da

$$R(z) = 1 + z + \frac{z^2}{2} + \ldots + \frac{z^p}{p!},$$

indipendentemente dal metodo. Studiamo adesso le forme delle regioni di assoluta stabilità di alcuni metodi di Runge-Kutta.

Per il metodo di Eulero esplicito si ha

$$R(z) = 1 + z.$$

La regione di assoluta stabilità è data da $S_A = \{z \in \mathbb{C} : |z + 1| < 1\}$. Il bordo di tale regione è il luogo dei punti z complessi per cui $|z - (-1)| = 1$, e quindi è la circonferenza di centro -1 e raggio 1. La regione di stabilità è dunque l'insieme dei punti del cerchio di centro -1 e raggio 1.

Per il metodo di Eulero implicito si ha

$$R(z) = \frac{1}{1 - z}.$$

È facile dimostrare che la regione di stabilità corrispondente è l'insieme dei punti del piano complesso *esterni* al cerchio di centro 1 e raggio 1. Tale regione di stabilità comprende il semipiano $\mathbb{C}_-$ e quindi il metodo di Eulero implicito è A-stabile.

Consideriamo adesso le regioni di stabilità del metodo dei trapezi e del metodo *midpoint*.

Per entrambi i metodi si ha (vedi Esercizio (8.4))

$$R(z) = \frac{2 + z}{2 - z}.$$

Il bordo della regione di stabilità è dato dal luogo geometrico dei punti per i quali

$$\frac{|z + 2|}{|z - 2|} = 1,$$

che risulta essere esattamente l'asse immaginario (in quanto luogo geometrico dei punti equidistanti dai due punti $+2$ e -2). La regione si assoluta stabilità coincide

dunque con $\mathbb{C}_-$, e dunque il metodo dei trapezi ed il metodo *midpoint* sono A-stabili. Tali metodi hanno la peculiare caratteristica di avere le stesse proprietà di stabilità della soluzione esatta. Essi risultano particolarmente indicati per alcune classi di problemi, come ad esempio sistemi lineari con matrici antisimmetriche. Tali problemi infatti presentano una proprietà di conservazione, che viene mantenuta anche a livello discreto da questi integratori. Integratori che mantengano a livello discreto certe proprietà di conservazione del sistema continuo sono detti integratori simplettici, e rappresentano attualmente un attivo campo di ricerca nel settore dei metodi numerici per sistemi di equazioni differenziali ordinarie. Tali metodi risultano anche molto adatti alla risoluzione del problema degli N corpi, poiché mantengono a livello discreto alcune proprietà di conservazione del problema degli N corpi. Si veda il libro di Sanz-Serna per una rassegna sugli integratori simplettici [49].

Il diagramma del bordo delle regioni di stabilità per i metodi Runge-Kutta espliciti con $p = \nu$, per $p = 1,2,3,4$ è facilmente ottenibile con MATLAB mediante il seguente script

```
% RKstabil.m
% Disegno regioni stabilita'
%
x = -3:0.1:3;
y = -3:0.1:3;
[X,Y] = meshgrid(x,y);
Z = X+i*Y;
R1 = 1 + Z;
R2 = R1 + Z.^2/2;
R3 = R2 + Z.^3/6;
R4 = R3 + Z.^4/24;
clf
hold on
contour(X,Y,abs(R1),[1 1])
contour(X,Y,abs(R2),[1 1])
contour(X,Y,abs(R3),[1 1])
contour(X,Y,abs(R4),[1 1])
axis('square')
axis([-3 3 -3 3])
hold off
```

Il grafico è mostrato nella Figura (8.8). Si osservi come le regioni di stabilità dei metodi espliciti del terzo e quarto ordine, oltre ad essere più ampie di quelle dei metodi di ordine inferiore, contengano al loro interno anche un segmento dell'asse immaginario.

8.8 Integrazione di equazioni differenziali con MATLAB

MATLAB possiede diverse funzioni per la integrazione di sistemi di equazioni differenziali ordinarie. La scelta della funzione dipenderà dal problema che si vuole risolvere. La sintassi comune a tutte le funzioni è una delle seguenti

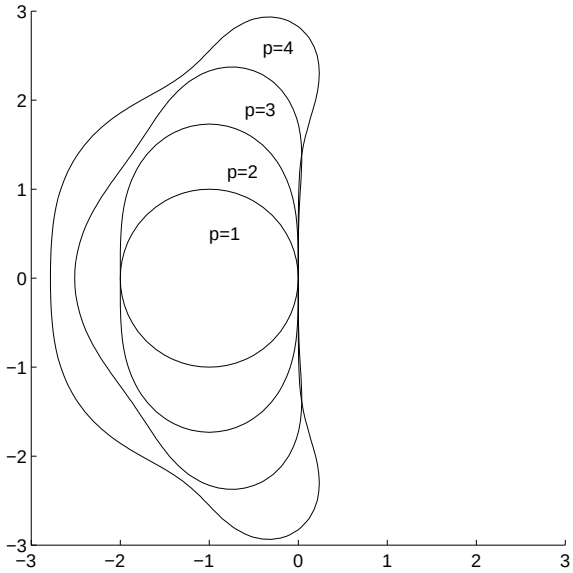

Figura 8.8 Bordi delle regioni di stabilità per i metodi Runge-Kutta espliciti con $p = \nu$, per $p = 1,2,3,4$

```
[T,Y] = solutore(odefun,tspan,y0),
[T,Y] = solutore(odefun,tspan,y0,options),
```

dove `solutore` è una delle seguenti funzioni MATLAB `ode45`, `ode23`, `ode113`, `ode15s`, `ode23s`, `ode23t`, `ode23tb`.

Queste funzioni permettono di risolvere problemi ai valori iniziali del tipo

$$y' = f(t,y), \quad y(t_0) = y_0.$$

Gli argomenti passati alla function sono i seguenti

`odefun` Rappresenta il nome di una funzione che valuta la $f(t,y)$. Tale funzione ha due argomenti, t scalare, ed y vettore colonna, e restituisce un vettore delle stesse dimensioni del vettore di ingresso y, che rappresenta il valore di f.

`tspan` Specifica l'intervallo di integrazione, $[t_0,T]$. In questo modo i tempi in uscita T saranno determinati dalla routine di controllo del passo. Se si vogliono specificare particolari valori dei tempi in uscita basta assegnare a `tspan` i tempi desiderati (in ordine crescente): `tspan = [t0, t1, ..., T]`.

`y0` È un vettore contenente le condizioni iniziali del sistema

`options` È un vettore che contiene opzioni sulla integrazione numerica, quali tolleranze, passo iniziale, passo massimo, ecc. Tale vettore viene creato dalla funzione `odeset`. Per maggiori dettagli si veda `help odeset`.

La function restituisce in uscita un vettore colonna T ed una matrice Y. T contiene i tempi ai quali la soluzione numerica è stata valutata. La Y è una matrice. Il numero di righe è uguale alla dimensione di T, il numero di colonne è uguale al numero di componenti del sistema. Ogni riga della matrice di uscita Y contiene i valori della soluzione al corrispondente tempo definito nel vettore T.

Diamo qui una breve descrizione dei metodi utilizzati dalle diverse funzioni.

`ode45` È basata su una coppia di metodi Runge-Kutta espliciti (4,5), la coppia di Dormand e Prince. È la funzione che si raccomanda di utilizzare come primo tentativo per la risoluzione di un nuovo problema.

`ode23` È una implementazione della coppia di metodi Runge-Kutta (2,3) di Bogacki e Shampine. Può essere più efficiente di `ode45` per tolleranze lasche ed in presenza di una moderata stiffness.

`ode113` È basato su una formula *predictor-corrector* di tipo Adams-Moulton, di ordine variabile. Può essere più efficiente di `ode45` per tolleranze restrittive e quando la funzione f risulta particolarmente costosa. È un solutore multistep, e pertanto richiede la soluzione numerica per i primi passi. La funzione provvede automaticamente a calcolare i primi passi necessari all'innesco del metodo multistep.

Gli algoritmi descritti finora sono indicati per problemi non-stiff. Se essi risultano particolarmente lenti, molto probabilmente siamo di fronte a problemi di tipo stiff, ed allora è opportuno provare ad utilizzare una delle funzioni seguenti

`ode15s` È un solutore ad ordine variabile basato su metodi multistep lineari impliciti. Opzionalmente può utilizzare metodi di tipo BDF. È consigliato quando `ode45` risulta poco efficiente o quando si sospetta che il problema sia stiff.

`ode23s` È basato su una formula di Rosenbrock di ordine 2. Può essere più efficiente di `ode15s` per tolleranze lasche. I metodi di Rosenbrock sono metodi di Runge-Kutta impliciti nei quali l'equazione non-lineare per la valutazione della soluzione al nuovo passo temporale viene ottenuta tramite un passo del metodo di Newton.

`ode23t` Utilizza la formula dei trapezi. È indicato per metodi moderatamente stiff, e quando non si vogliano introdurre smorzamenti di tipo numerico sulla soluzione.

`ode23tb` Utilizza una combinazione di metodo dei trapezi e metodo BDF di ordine 2.

Alcune delle opzioni di uso più comune Se si vogliono cambiare le opzioni di *default* occorre usare il programma `odeset`. Le opzioni di uso più comune riguardano le tolleranze sull'errore locale di troncamento. Esse vengono stabilite con una istruzione del tipo

```
options = odeset('reltol',epsr,'abstol',epsa),
```

dove `epsr` è uno scalare, e `epsa` è uno scalare o un vettore della stessa dimensione m del sistema. La condizione sull'errore locale di troncamento è del tipo espresso dalla (8.37), che qui riportiamo. Per ogni componente i si impone che

$$|\sigma^{(i)}| \leq \max(\varepsilon_r y^{(i)}, \varepsilon_{a\,i})h. \tag{8.58}$$

È anche possibile imporre tolleranze sulle norme, invece che sulle singole componenti del vettore soluzione.

Un altro parametro di uso comune, quando si risolvono problemi stiff, è l'opzione `jacobian`, che permette di specificare lo jacobiano del sistema $\partial f/\partial y$. Lo jacobiano è utilizzato dai solutori impliciti per risolvere, in genere mediante il metodo di Newton, le equazioni non lineari per la determinazione dell'incognita al nuovo passo temporale.

8.9 Applicazioni

In questa sezione consideriamo alcune applicazioni dei metodi studiati. In particolare, considereremo il problema degli N corpi ed il moto di un punto vincolato ad una superfice.

8.9.1 Sole-Terra-Luna

Un problema affascinante è quello dello studio del comportamento del sistema solare. Tale studio può essere effettuato a diversi livelli. Dal punto di vista matematico ci si può porre, ad esempio, il problema della stabilità del sistema solare, visto come problema di N corpi soggetti alla mutua interazione gravitazionale di tipo Newtoniano. Il sistema solare è stabile, cioè per tutti i tempi le traiettorie dei pianeti continueranno a rimanere limitate, o può succedere che qualche pianeta, a causa delle piccole ma cumulative interazioni con gli altri pianeti, prima o poi parta "per la tangente" ed abbandoni per sempre il sistema solare? Malgrado la fondamentale importanza e l'approfondito studio che tale problema ha ricevuto, esso rimane ancora un problema aperto. Esistono infatti dei teoremi generali che permettono di dare condizioni sufficienti sulla stabilità di sistemi dinamici cosiddetti Hamiltoniani, che includono il sistema degli N corpi come caso particolare, tuttavia il sistema solare non rientra nelle condizioni di applicabilità dei teoremi noti fino ad ora. Una simulazione numerica del sistema solare non potrebbe mai dare una dimostrazione della stabilità del sistema solare, poiché questo richiederebbe una simulazione accurata per un tempo arbitrariamente grande. Viceversa, una accurata simulazione può essere interessante di per se, per fornire altri tipi di informazioni. Il settore che si occupa della dinamica di sistemi autogravitanti prende il nome di Meccanica Celeste. In essa si studia non solo il comportamento del sistema solare, ma anche altri fenomeni, quali le orbite dei satelliti dei pianeti maggiori, il calcolo e le tecniche di controllo delle orbite di satelliti artificiali, e così via.

In questa sede ci limitiamo a studiare il moto del sistema Sole-Terra-Luna. La prima cosa da fare è riscrivere il sistema di N punti come un sistema di $6N$ equazioni differenziali del primo ordine. Poniamo $U = (P; V)$, dove $P = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_N, y_N, z_N)^T$, e V è il corrispondente vettore, di dimensione $3N$, contenete le velocità delle particelle. Poniamo $F = (V; A)$, dove A è il vettore contenente le accelerazioni delle particelle, che esprime la legge di gravitazione di Newton (in pratica la i-esima equazione vettoriale è fornita dalla equazione (8.2) diviso per la massa della particella i). A questo punto si tratta di costruire una `function` MATLAB che, dato il vettore U, calcoli il vettore F. Tale funzione può poi essere utilizzata dalle funzioni MATLAB per l'integrazione delle equazioni differenziali. La funzione che calcola F è qui riportata

```
function F = fnbody(t,U)
% Sintassi  F = fnbody(t,U)
% Implementa la funzione del problema degli N corpi in
%  forma vettoriale
% U = (P;V);
% X = (x1,y1,z1,x2,y2,z2,...,xn,yn,zn)

global M gn    % Contiene le masse e la costante di gravitazione

N = length(U)/6;               % Numero di corpi
ix = 1:3:3*N-2;                % indice delle coordinate x
x = U(ix);
y = U(ix+1);
z = U(ix+2);
DX = x*ones(1,N)-ones(N,1)*x'; % matrice differenze delle ascisse
DY = y*ones(1,N)-ones(N,1)*y';
DZ = z*ones(1,N)-ones(N,1)*z';
D3 = (M*ones(1,N))./(DX.^2+DY.^2+DZ.^2+eye(N)).^(3/2);
ax = sum(DX.*D3);
ay = sum(DY.*D3);
az = sum(DZ.*D3);
F(1:3*N) = U(3*N+1:6*N); % La prima meta' di F contiene le velocita'
F(3*N+ix) = gn*ax;
F(3*N+ix+1) = gn*ay;
F(3*N+ix+2) = gn*az;
F=F(:);                        % F e' un vettore colonna
```

La funzione realizza il calcolo in modo vettoriale. Il codice così risulta più compatto ed efficiente. La relazione che fornisce una delle componenti delle accelerazioni delle particelle, diciamo la componente x, si scrive

$$a_{xj} = \gamma \sum_i A_{ij},$$

dove

$$A_{ij} = \frac{x_i - x_j}{[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2]^{3/2}} m_i.$$

Nella funzione dapprima si calcola la matrice $N \times N$ A_{ij}, e poi, tramite il comando `sum`, si calcolano le somme degli elementi delle colonne di A. Notiamo che nel calcolo di `D3` al denominatore vengono aggiunti gli elementi della matrice identità.

Questo viene effettuato per evitare di incorrere in una forma indeterminata del tipo $0/0$ che produrrebbe un `NaN` quando si effettua una divisione fra gli elementi diagonali della matrice DX e $D3$. In pratica questo artificio evita di dover tener conto che la somma degli elementi di A deve essere effettuata per $i \neq j$. Gli elementi si sommano tutti, e l'elemento per $i = j$ risulta automaticamente nullo, poiché le matrici `DX`, `DY`, e `DZ` sono nulle sulla diagonale.

La cosa più difficile nella impostazione del problema degli N corpi è fornire i dati iniziali. Tali dati si possono desumere da tabelle di dati astronomici. Citiamo ad esempio il classico testo di Archie Roy di meccanica celeste [48].

Poiché è molto difficile fornire le condizioni iniziali in un generico punto dell'orbita, le forniremo supponendo che la Terra si trovi all'afelio (massima distanza dal Sole) e la Luna si trovi all'apogeo (massima distanza dalla Terra). In questo modo possiamo sfruttare i dati orbitali riguardanti le distanze massime e le velocità minime. Inoltre sappiamo che le velocità iniziali saranno ortogonali al raggio vettore (rispettivamente Terra-Sole e Luna-Terra).

Un ulteriore accorgimento, di grande importanza in molte applicazioni in fisica, riguarda la scelta di unità di misura adeguate. È sempre opportuno scegliere unità di misura nelle quali le grandezze in gioco siano il più possibile vicine all'unità. Questo per diversi motivi. Innanzitutto per evitare che il calcolatore possa incorrere in problemi di *overflow* o *underflow*, che si possono verificare se il computer si trova a maneggiare numeri troppo grandi o troppo piccoli. Un secondo motivo è che se le grandezze in gioco non sono ben scalate si rischia di perdere accuratezza per problemi di arrotondamento e cancellazioni numeriche. Si pensi semplicemente alla scarsa efficacia della tecnica di pivoting parziale quando le equazioni di un sistema lineare algebrico sono mal scalate. Infine un altro motivo riguarda le scelta delle tolleranze che imponiamo all'integratore numerico. Normalmente si impone una tolleranza relativa ed una tolleranza assoluta. La tolleranza relativa, per definizione, non dipende dal riscalamento delle variabili. Per la scelta di una tolleranza assoluta che abbia senso, viceversa, è necessario conoscere il *range* di variazione delle variabili in gioco. Se le variabili, ad esempio, sono distanze astronomiche espresse in metri, per il sistema solare esse sono dell'ordine di 10^{11} e quindi imporre una tolleranza assoluta dell'ordine di 10^{-3} risulterebbe senz'altro troppo restrittivo. Inoltre, anche ammesso che l'utilizzatore sappia bene quali siano gli ordini di grandezza delle variabili in gioco, se le grandezze non sono ben scalate occorrerebbe utilizzare tolleranze diverse a seconda delle quantità fisiche che si considerano, cosa evidentemente poco pratico.

La soluzione a tutti questi problemi è quella di porre le equazioni in forma adimensionale, riscalando le variabili in gioco mediante grandezze fisiche caratteristiche del problema. Ad esempio, per il problema del sistema Sole-Terra-Luna, una scelta ragionevole potrebbe essere quella di esprimere le lunghezze in termini della distanza media Terra-Sole, i tempi in termini di anni, e le masse in termini di massa della Terra. Così facendo si restringe la variabilità delle grandezze al minimo indispensabile, e si possono agevolmente fissare le tolleranze.[5]

[5] Vi sono molti altri vantaggi nella scrittura di un problema in forma adimensionale, soprattutto per sistemi continui complessi. In molti casi l'analisi dimensionale permette di individuare immediatamente alcune delle dipendenze funzionali delle diverse grandezze in gioco, semplificando la trattazione del problema. La riduzione nella forma dimensionale permette di individuare i parametri adimensionali da cui dipende un certo sistema fisico. Tali parametri individuano i

Il programma `SSolar.m` realizza la simulazione del sistema Sole-Terra-Luna.

```
% SSolar.m
% Implementa la simulazione del moto dei pianeti
% del sistema solare
% Le equazioni sono scritte nella forma non dimensionale
%
global M gn  % masse degli astri e costante di gravitazione
global beta  % contiene i coefficienti del metodo di Adams-Bashforth
gn = 6.67e-11; % Costante di gravitazione universale in MKS

% Definizione delle masse e delle grandezze orbitali
% 1: Sole
% 2: Terra
% 3: Luna
MT = 5.98e24;            % Massa della Terra in Kg
LTS = 149600000000;      % Distanza media Terra-Sole in m
Year = 86400*365.25;     % Anno in s
alphaL = 23*pi/180;      % Inclinaz. orbita lunare risp. all'eclittica
V0 = LTS/Year;           % Unita' di lunghezza per le velocita'
gn = gn*MT/LTS*(Year/LTS)^2; % costante gravitazionale scalata

% Dati scalati
MS = 333000;
ML = 1/81;
LTL = 406720000/LTS; % massima distanza Terra-Luna
vt0 = 29295/V0;
          % velocita' orbi. min. della Terra (risp. al Sole)
vl0 = 3456/3.6/V0;
         % velocita' orbit. min. della Luna (risp. alla Terra)
% Condizioni iniziali
% Il Sole al centro fermo
PS = [0;0;0];
VS = [0;0;0];
% La Terra all'afelio
PT = [152.1/149.6;0;0];
% con velocita; orbitale minima
VT = [0;vt0;0];
% La Luna al perigeo, inclinata sull'eclittica
PL = PT + [LTL*cos(alphaL);0;LTL*sin(alphaL)];
% Con velocita' orbitale minima
VL = VT + [0;vl0;0];
% Vettore delle masse
M = [MS;1;ML];
% Vettore delle condizioni iniziali
P = [PS;PT;PL;VS;VT;VL];

options = odeset('abstol',1e-8,'reltol',1e-8);
```

"regimi" nei quali operano i sistemi. Due sistemi fisici retti dalle stesse equazioni, anche molto diversi fra loro, ma caratterizzati dagli stessi parametri adimensionali mostrano essenzialmente il medesimo comportamento. Questo comportamento viene sfruttato (e molto di più fino a pochi decenni or sono) nei modelli meccanici che riproducono un sistema che si vuole costruire in scala ridotta, come ad esempio un modellino di aereo nella galleria del vento.

```
clf
hold on
E0 = energia(P)
tspan = [0 1];  % Integra per un anno
[T,X] = ode45('fnbody',tspan,P,options);
% Visualizza le orbite,
% amplificando la distanza Terra-Luna di amp volte
amp = 20;
figure(1)
plot3(X(:,4),X(:,5),X(:,6),...
    X(:,4)+amp*(X(:,7)-X(:,4)),X(:,5)+amp*(X(:,8)-X(:,5)),X(:,6)+...
    amp*(X(:,9)-X(:,6)))
axis('equal')
axis(1.2*[-1 1 -1 1]);
N = length(T);
E1 = energia(X(N,:))
% Integrazione effettuata utilizzando Adams-Bashforth a passo fisso
% con lo stesso numero di passi usato da ode45
h = tspan(2)/N;
q = 5;  % Scegli l'ordine del metodo di Adams-Bashforth
T0 = h*(0:q);
% Calcola le condizioni iniziali del metodo di AB utilizzando ode45
options = odeset('abstol',1e-13,'reltol',1e-13);
[T0,X0] = ode45('fnbody',T0,P,options);
% Calcola i coefficienti del metodo di AB
beta = double(fliplr(AdamsBashforth(q)));
[T,X] = AB('fnbody',T0,X0,N,h);
figure(1)
plot3(X(:,4),X(:,5),X(:,6),'r:',...
    X(:,4)+amp*(X(:,7)-X(:,4)),X(:,5)+amp*(X(:,8)-X(:,5)),X(:,6)+...
    amp*(X(:,9)-X(:,6)),':g')
axis('equal')
axis(1.2*[-1 1 -1 1]);
E2 = energia(X(N,:))
```

Per confronto, lo stesso problema viene risolto dapprima con il codice `ode45` del MATLAB e poi con il codice `AB.m`, che utilizza un metodo di Adams-Bashforth del sesto ordine. È possibile verificare che, a parità di numero di passi, il codice basato sul metodo di Adams-Bashforth è più accurato e più veloce. Questo perché il problema è molto regolare, e quindi ci sono vantaggi ad utilizzare metodi di ordine molto elevato. Inoltre il metodo multistep è più economico del Runge-Kutta poiché deve effettuare solo una valutazione di funzione per avanzare di un passo. Un controllo sulla accuratezza del calcolo potrebbe essere effettuato monitorando l'andamento della energia totale, che è un invariante del moto. L'energia totale è data dalla somma di energia cinetica ed energia potenziale

$$E_{\text{tot}} = E_{\text{cin}} + E_{\text{pot}},$$

dove

$$E_{\text{cin}} = \sum_{i=1}^{N} \frac{1}{2} m_i |v_i|^2,$$

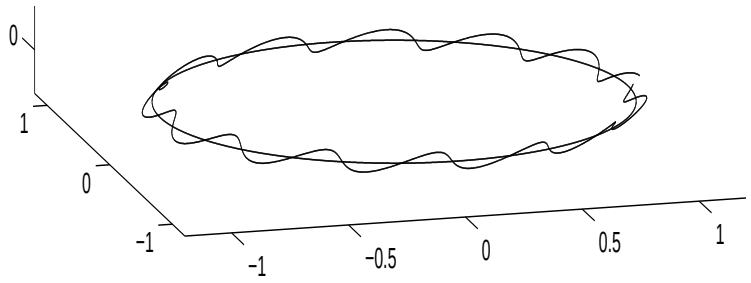

Figura 8.9 Orbite della Terra (ellisse) e della Luna. La distanza Terra-Luna è stata amplificata di un fattore 20 per evidenziare le differenze fra le traiettorie del nostro pianeta e del nostro satellite.

$$E_{\text{pot}} = -\frac{\gamma}{2} \sum_{ij} \frac{m_i m_j}{|P_i - P_j|}.$$

Le traiettorie della Terra e della Luna ottenute con i due metodi (praticamente indistinguibili) sono illustrate nella Figura (8.9). Per evidenziare la traiettoria della Luna attorno alla Terra, la distanza Terra-Luna è stata amplificata (solo nella fase finale della rappresentazione grafica!) di un fattore 20.

8.9.2 Moto di un punto vincolato ad una superficie liscia

Vedremo in questa sezione come calcolare le traiettorie di un punto vincolato a stare su una superficie liscia, e soggetto ad un campo di accelerazioni A.

Le equazioni del moto sono illustrate nella Sezione 8.1, equazione ((8.7)).

Consideriamo qui il problema di come implementare la funzione che definisce il moto. Scriviamo il sistema nella forma

$$U' = F(U),$$

dove $U = (P; V)$, $F = (V; A_s)$. Il campo di accelerazioni A_s si scrive in maniera molto compatta utilizzando la notazione vettoriale del MATLAB. Posto $G = \nabla\varphi$ (vettore colonna) e $H = \nabla\nabla\varphi$ (matrice hessiana), le equazioni del moto si scrivono semplicemente

$$P' = V, \quad V' = \left(I - \frac{GG^T}{G^T G}\right) A - \frac{V^T H V}{G^T G} G,$$

dove l'apice denota la trasposizione. Come esempi consideriamo il moto di un punto su una sfera, soggetto all'azione di una accelerazione esterna costante diretta verso il basso, il moto inerziale si un punto sulla superficie di un toro, il moto inerziale di un punto su una superficie a forma di cilindro a sezione variabile, ed

il moto di un punto vincolato a stare su un paraboloide a sella, e attirato verso l'origine da una molla elastica.

Le immagini delle traiettorie sono riportate nelle Figure 8.10, 8.11 e 8.12.

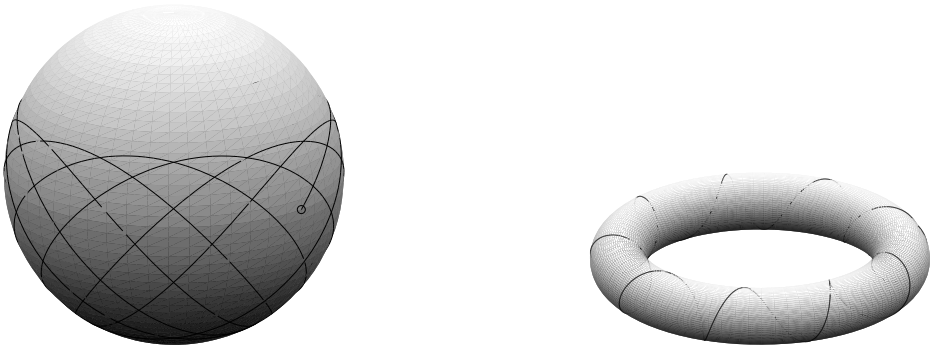

Figura 8.10 Traiettoria di un punto vincolato ad una sfera e soggetto ad un campo di forze esterno costante diretto verso il basso (sinistra) e traiettoria di un punto vincolato alla superficie di un toro (destra).

Per avere una maggiore impressione della tridimensionalità del problema, nei primi tre esempi è stata visualizzata anche la superficie. La rappresentazione della superficie è stata effettuata con MATLAB. Il codice MATLAB `superfici.m` per il calcolo delle traiettorie e per la rappresentazione delle superfici è disponibile in rete al sito del libro.

Nell'ultimo esempio sono state rappresentate le proiezioni delle traiettorie di un esempio nei piani x,y e x,z. Come si vede, la semplice forza di richiamo di una molla produce delle traiettorie complesse per un punto vincolato ad un paraboloide iperbolico.

8.10 Esercizi

Esercizio 8.1 Si dimostri che la soluzione del problema ai valori iniziali

$$y' = y^2, \quad y(0) = y_0,$$

è data da

$$y(t) = \frac{y_0}{1 - y_0 t},$$

e si discuta il problema della esistenza della soluzione per ogni $t \geq 0$, in dipendenza del dato iniziale. Ci sono dei dati iniziali per i quali la soluzione non esiste $\forall t \geq 0$? Se si, quale ipotesi del teorema di Cauchy è venuta a mancare?

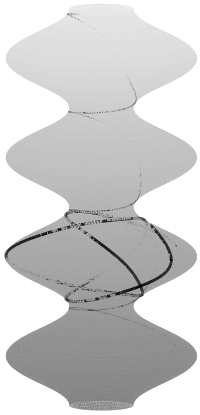

Figura 8.11 Traiettoria di un punto vincolato al cilindro a sezione variabile ottenuto facendo ruotare una sinusoide sfalsata attorno all'asse z. Per alcune condizioni iniziali il punto passa da un rigonfiamento al successivo (traiettoria continua), mentre per altre condizioni iniziali il punto è intrappolato attorno ad un rigonfiamento.

Esercizio 8.2 Dimostrare che se $A \in \mathbb{R}^{\nu \times \nu}$ è una matrice triangolare superiore con elementi diagonali nulli allora A^2 è triangolare superiore con elementi sulla diagonale principale e sulla prima sottodiagonale nulli, A^k ha le prime k diagonali nulle, e, in particolare, $A^\nu = 0$.

Esercizio 8.3 Sfruttare il risultato dell'esercizio precedente per dimostrare che

$$(I - zA)^{-1} = I + zA + z^2A^2 + \ldots + z^{\nu-1}A^{\nu-1},$$

ed utilizzando questo risultato, dimostrare direttamente, senza far uso della espressione (8.57), che la funzione $R(z)$ espressa nell'equazione ((8.56)) è un polinomio in z.

Esercizio 8.4 Dimostrare che la funzione di assoluta stabilità del metodo dei trapezi e del punto medio è data da $R(z) = (2 + z)/(2 - z)$.

Esercizio 8.5 Determinare i parametri del metodo di collocazione gaussiana basato sulla formula di quadratura di Gauss-Legendre con due punti

Esercizio 8.6 Risolvere utilizzando i metodi di Runge-Kutta espliciti di ordine $p = 1, \ldots, 4$ ed i metodi di Adams-Bashforth di pari ordine le equazioni di Lotka-Volterra del *modello preda-predatore*

$$\begin{aligned} y' &= \alpha y + \beta yz, \\ z' &= \gamma z + \delta yz, \end{aligned}$$

dove $y(t)$ rappresenta il numero di prede e $z(t)$ il numero di predatori, corrispondenti ai seguenti valori dei parametri, $\alpha = 2$, $\gamma = -1$, $\beta = -\delta = -0.001$, (volpi-conigli) e per i seguenti dati iniziali

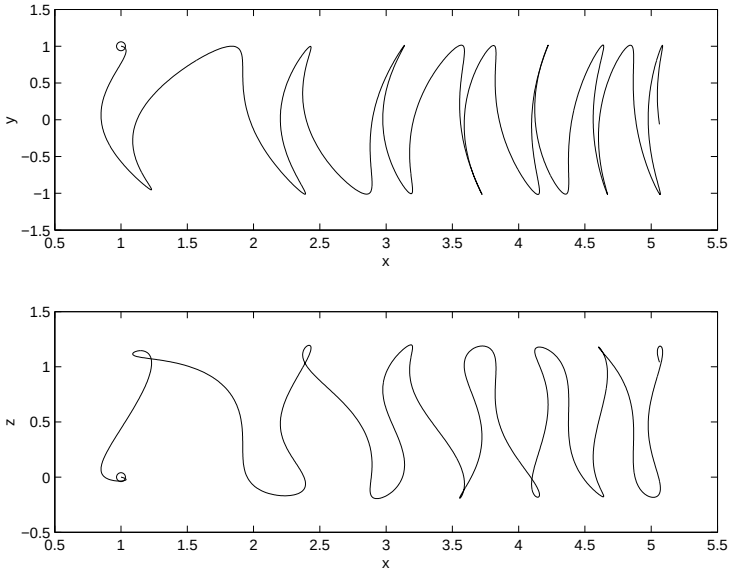

Figura 8.12 Proiezioni nei piani x,y e x,z delle traiettorie di un punto vincolato a stare su un paraboloide iperbolico di equazioni $z = y^2 - x^2$, che si muove sotto l'azione di una molla che lo attira verso l'origine degli assi coordinati.

a) $y(0) = 300$, $z(0) = 150$,

b) $y(0) = 15$, $z(0) = 22$.

In input si deve assegnare il tempo di calcolo finale $t_{\max}$ ed il numero di punti utilizzati N. In output si riporteranno le tre curve di errore (E_p,t) ottenute confrontando la soluzione di ordine p con quella di ordine $p+1$, $p = 1, \ldots, 3$, in scala logaritmica ed i quattro grafici della soluzione relativi alle prede (y, t), ai predatori (z, t), e dei predatori rispetto alle prede (y, z).

Esercizio 8.7 Risolvere utilizzando le diverse funzioni di libreria di MATLAB le equazioni relative al problema dei due corpi

$$\begin{aligned} x'' &= -\frac{x}{(x^2+y^2)^{3/2}}, \\ y'' &= -\frac{y}{(x^2+y^2)^{3/2}}, \end{aligned}$$

e mostrare che, al variare di t, $x(t)$ e $y(t)$ descrivono una ellisse. Si consideri come dato iniziale

$$x(0) = 0.4,\ x'(0) = 0,\ y(0) = 0,\ y'(0) = 2.$$

Realizzare i due grafici della soluzione (x, y), e il diagramma di fase (x', y') e confrontare l'efficienza dei diversi metodi per una fissata tolleranza.

Esercizio 8.8 Si consideri il seguente sistema di equazioni differenziali nonlineari del primo ordine (*modello di Lorenz*)

$$\begin{aligned} x' &= \sigma(y-x), \\ y' &= rx - y - xz, \\ z' &= -bz + xy, \end{aligned}$$

dove $x = x(t)$, $y = y(t)$, $z = z(t)$.

Calcolarne le soluzioni (con un metodo opportuno) nei casi

a) $\sigma = 10, b = 8/3, r = 28, \qquad x(0) = 10, y(0) = 0, z(0) = 20,$

b) $\sigma = 10, b = 8/3, r = 28, \qquad x(0) = 11, y(0) = 0, z(0) = 20,$

per un tempo $tmax$ assegnato in input. Si realizzino i grafici di (x,t), (y,t), (z,t) ed il grafico di (x,y,z) utilizzando l'istruzione `plot3(x,y,z)`. Osservare che pur essendo i dati iniziali molto vicini il comportamento della soluzione per tempi grandi risulta completamente differente.