Dispense del corso Laboratorio di Metodi Numerici per le Equazioni Differenziali

Dott. Marco Caliari

a.a. 2010/11

La versione più aggiornata (al 10 agosto 2011) si trova all'indirizzo
http://profs.sci.univr.it/∼caliari/aa1011/equazioni_differenziali/dispense.pdf

Questi appunti non hanno nessuna pretesa di completezza. Sono solo alcune note ed esercizi che affiancano il corso di Metodi Numerici per le Equazioni Differenziali. Sono inoltre da considerarsi in perenne "under revision" e pertanto possono contenere discrepanze, inesattezze o errori.

Indice

Parte 0

Preliminari

Capitolo 1

Interpolazione polinomiale a tratti

Data una funzione $f\colon [a,b] \to \mathbb{R}$ e un'insieme $\{x_i\}_{i=1}^m \subset [a,b]$ di nodi ordinati ($x_{i-1} < x_i$), consideriamo l'interpolante polinomiale a tratti $L^{\mathrm{c}}_{k-1} f$ di grado $k-1$. Su ogni intervallo $[x_i, x_{i+1}]$ di lunghezza $h_i = x_{i+1} - x_i$ essa è il polinomio di grado $k-1$

$$a_{i,1}(x-x_i)^{k-1} + a_{i,2}(x-x_i)^{k-2} + \ldots + a_{i,k-1}(x-x_i) + a_{i,k}\ . \tag{1.1}$$

Dunque, l'interpolante polinomiale a tratti è completamente nota una volta noti i nodi e i coefficienti di ogni polinomio.

In GNU Octave, l'interpolante polinomiale a tratti è definita mediante una struttura solitamente chiamata `pp` *(piecewise polynomial)* che si costruisce con il comando `mkpp(x,P)`, ove `x` è il vettore di nodi e `P` è la matrice, con riferimento a (1.1),

$$P_{ij} = a_{i,j}\ .$$

Nota una struttura `pp`, è possibile valutare il valore dell'interpolante in un generico target $\bar{x}$ con il comando `ppval(pp,xbar)`.

1.1 Interpolazione lineare a tratti

Dati i vettori $[x_1, \ldots, x_m]^{\mathrm{T}}$ e $[f_1, \ldots, f_m]^{\mathrm{T}}$, nell'intervallo $[x_i, x_{i+1}]$ l'interpolante lineare a tratti coincide con il polinomio di grado uno

$$\frac{f_{i+1} - f_i}{h_i}(x - x_i) + f_i$$

Pertanto, si costruisce la corrispondente struttura `pp` con il comando

```
> pp = mkpp(x,[(f(2:m)-f(1:m-1))./h,f(1:m-1)])
```

Capitolo 2

Formule di quadratura gaussiana

Dato un intervallo (a, b) (eventualmente anche non limitato) e una funzione peso $w(x)$ non negativa su (a, b), si considera il prodotto scalare

$$(f, g) = \int_a^b f(x)g(x)w(x)\mathrm{d}x$$

con l'ipotesi

$$\int_a^b |x|^k w(x)\mathrm{d}x < \infty, \quad k \geq 0$$

Allora, esiste un'unica famiglia $\{p_j(x)\}_j$, $p_j(x)$ polinomio di grado j, *ortonormale* rispetto al prodotto scalare

$$\int_a^b p_j(x)p_i(x)w(x)\mathrm{d}x = \delta_{ij}$$

Gli zeri $\{x_n\}_{n=1}^m$ del polinomio $p_m(x)$ sono interni all'intervallo (a, b) e assieme ai pesi

$$w_n = \int_a^b L_n(x)w(x)\mathrm{d}x, \quad 1 \leq n \leq m$$

ove $L_n(x)$ è il polinomio di Lagrange che vale 1 in x_n e zero in tutti gli altri nodi, costituiscono una formula di quadratura *gaussiana* esatta fino al grado polinomiale $2m - 1$, cioè

$$\int_a^b p_j(x)w(x)\mathrm{d}x = \sum_{n=1}^m p_j(x_n)w_n, \quad 0 \leq j \leq 2m - 1$$

In particolare

$$\delta_{ij} = \int_a^b p_j(x)p_i(x)w(x)\mathrm{d}x = \sum_{n=1}^{m} p_j(x_n)p_i(x_n)w_n, \quad 0 \le i,j \le m-1$$

Nel caso in cui (a,b) sia limitato, esiste un'unica formula di quadratura esatta fino al grado polinomiale $2m-3$ che usa come nodi $\bar{x}_1 = a$, $\bar{x}_m = b$ e gli zeri $\{\bar{x}_n\}_{n=2}^{m-1}$ del polinomio di grado $m-2$ della famiglia di polinomi ortogonali rispetto alla funzione peso $w(x)(x-a)(b-x)$. In questo caso si ha, in particolare,

$$\delta_{ij} = \int_a^b p_j(x)p_i(x)w(x)\mathrm{d}x = \sum_{n=1}^{m} p_j(\bar{x}_n)p_i(\bar{x}_n)\bar{w}_n, \quad \begin{array}{l} 0 \le i \le m-3 \\ 0 \le j \le m-1 \end{array}$$

La famiglia $\{\phi_j(x)\}_{j=1}^{m}$, ove $\phi_j(x) = p_{j-1}(x)\sqrt{w(x)}$ è ovviamente ortonormale rispetto al prodotto scalare

$$(f,g) = \int_a^b f(x)g(x)\mathrm{d}x$$

e per essa valgono le osservazioni fatte sopra riguardo al calcolo degli integrali.

2.1 Quadratura gaussiana di Chebyshev e di Chebyshev–Lobatto

Per integrali del tipo

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\mathrm{d}x$$

i polinomi ortogonali da considerare sono quelli di Chebyshev

$$p_j(x) = T_j(x) = \cos(j \arccos(x))$$

che soddisfano la relazione di ricorrenza

$$\begin{aligned} T_0(x) &= 1, \quad T_1(x) = 1 \\ T_{j+1}(x) &= 2xT_j(x) - T_{j-1}(x), \quad j \ge 1 \end{aligned}$$

Gli zeri del polinomio di grado m soddisfano

$$m \arccos(x) = \frac{\pi}{2} + n\pi$$

da cui

$$x_n = \cos\left(\frac{\frac{\pi}{2} + (n-1)\pi}{m}\right) = \cos\left(\frac{(2n-1)\pi}{2m}\right), \quad 1 \le n \le m$$

e i corrispondenti pesi di quadratura sono costanti e valgono

$$w_n = \frac{\pi}{m}, \quad 1 \le n \le m$$

I nodi di Gauss–Chebyshev–Lobatto sono invece

$$\bar{x}_n = \cos\left(\frac{(n-1)\pi}{m-1}\right), \quad 1 \le n \le m$$

e i corrispondenti pesi

$$\bar{w}_n = \begin{cases} \dfrac{\pi}{2(m-1)} & \text{per } n = 1 \text{ o } n = m \\ \dfrac{\pi}{m-1} & \text{per } 2 \le n \le m-1 \end{cases}$$

Capitolo 3

Metodi iterativi per sistemi di equazioni lineari

I metodi iterativi per la soluzione del sistema lineare

$$Ax = b \tag{3.1}$$

si basano sull'idea di calcolare la soluzione come limite di una successione di vettori

$$x = \lim_{l \to \infty} x^{(l)} .$$

Una strategia generale per costruire la successione $\{x^{(l)}\}_l$ è basata sullo splitting $A = P - M$, ove P è non singolare. Assegnato $x^{(1)}$, il termine $x^{(l+1)}$ è calcolato ricorsivamente come

$$Px^{(l+1)} = Mx^{(l)} + b, \quad l \geq 1 \tag{3.2}$$

Posto $e^{(l)} = x - x^{(l)}$, si ha

$$e^{(l)} = Be^{l-1}, \quad B = P^{-1}M = I - P^{-1}A ,$$

ove B è chiamata *matrice di iterazione*.

Lemma 1. *Si ha* $\lim_{l\to\infty} e^{(l)} = 0$ *per ogni* $e^{(1)}$ *se e solo se* $\lim_{l\to\infty} B^l = 0$, *cioè se e solo se* $\rho(B) < 1$.

3.1 Metodi di Richardson

Indicato con $r^{(l)}$ il *residuo*

$$r^{(l)} = b - Ax^{(l)} = Ax - Ax^{(l)} = A(x - x^{(l)}) = Ae^{(l)} ,$$

il metodo (3.2) può essere riscritto come

$$P(x^{(l+1)} - x^{(l)}) = r^{(l)} \ . \tag{3.3}$$

In questo contesto, P viene chiamata *matrice di precondizionamento* o *precondizionatore* di A e viene scelta in modo che la matrice di iterazione $B = I - P^{-1}A$ abbia un raggio spettrale minore di 1 e la risoluzione di (3.3) sia "facile".

Una generalizzazione dello schema (3.3) è il *metodo di Richardson*: dato $x^{(1)}$, $x^{(l+1)}$ è calcolato ricorsivamente come

$$P(x^{(l+1)} - x^{(l)}) = \alpha r^{(l)} \ ,$$

ove α è un opportuno parametro di accelerazione. Dati $x^{(1)}$ e $r^{(1)} = b - Ax^{(1)}$, l'algoritmo per calcolare $x^{(l+1)}$ è

$$\begin{aligned} Pz^{(l)} &= r^{(l)} \\ x^{(l+1)} &= x^{(l)} + \alpha z^{(l)} \\ r^{(l+1)} &= r^{(l)} - \alpha A z^{(l)} \end{aligned} \tag{3.4}$$

Il costo di un'iterazione è dato essenzialmente dalla risoluzione di un sistema lineare $Pz^{(l)} = r^{(l)}$ facile e dal prodotto matrice-vettore $Az^{(l)}$. Tali metodi risulteranno particolarmente vantaggiosi per matrici *sparse*, in cui il numero di elementi diversi da zero è $\mathcal{O}(N)$ piuttosto che $\mathcal{O}(N^2)$ (e dunque il costo di un prodotto matrice-vettore è $\mathcal{O}(N)$), se l'ordine della matrice è N.

Il calcolo del residuo $r^{(l+1)} = r^{(l)} - \alpha A z^{(l)}$ (invece di $r^{(l+1)} = b - Ax^{(l+1)}$) permette di ridurre la propagazione, attraverso il prodotto matrice-vettore, degli errori, in quanto il vettore $z^{(l)}$, contrariamente a $x^{(l+1)}$, diminuisce in modulo al crescere di l.

3.1.1 Metodo del gradiente precondizionato

Siano A e P simmetriche e definite positive. Il metodo di Richardson può essere generalizzato con una scelta dinamica del parametro di accelerazione, prendendo $\alpha = \alpha_l$ in modo tale che

$$\|x - x^{(l+1)}\|_A, \quad \|y\|_A = \sqrt{y^{\mathrm{T}} A y}$$

sia minima. Si ha

$$\begin{aligned} \|x - x^{(l+1)}\|_A^2 &= (x - x^{(l)} - \alpha_l z^{(l)})^{\mathrm{T}} A (x - x^{(l)} - \alpha_l z^{(l)}) = \\ &= \alpha_l^2 {z^{(l)}}^{\mathrm{T}} A z^{(l)} - 2\alpha_l {z^{(l)}}^{\mathrm{T}} A (x - x^{(l)}) + (x - x^{(l)})^{\mathrm{T}} A (x - x^{(l)}) \end{aligned}$$

e dunque il minimo è dato dalla scelta

$$\alpha_l = \frac{{z^{(l)}}^{\mathrm{T}} r^{(l)}}{{z^{(l)}}^{\mathrm{T}} A z^{(l)}} .$$

Il metodo ottenuto si chiama *metodo del gradiente precondizionato*. Dati $x^{(1)}$ e $r^{(1)}$, l'algoritmo per calcolare $x^{(l+1)}$ è

$$\begin{aligned} Pz^{(l)} &= r^{(l)} \\ \alpha_l &= \frac{{z^{(l)}}^{\mathrm{T}} r^{(l)}}{{z^{(l)}}^{\mathrm{T}} A z^{(l)}} \\ x^{(l+1)} &= x^{(l)} + \alpha_l z^{(l)} \\ r^{(l+1)} &= r^{(l)} - \alpha_l A z^{(l)} \end{aligned} \tag{3.5}$$

Nel caso si scelga $P = I$, si ottiene il *metodo del gradiente* (noto anche come *steepest descent*).

3.1.2 Metodo del gradiente coniugato precondizionato

Siano A e P simmetriche e definite positive. Il *metodo del gradiente coniugato precondizionato* è una generalizzazione del metodo di Richardson in cui

$$x^{(l+1)} = x^{(l)} + \alpha_l p^{(l)}$$

ove i $\{p^{(l)}\}_l$ sono *coniugati*, cioè soddisfano

$${p^{(i)}}^{\mathrm{T}} A p^{(j)} = 0, \quad i \neq j$$

Per soddisfare questa proprietà è necessaria l'introduzione di un ulteriore parametro β_l. Dati $x^{(1)}$, $r^{(1)}$, $Pz^{(1)} = r^{(1)}$ e $p^{(1)} = z^{(1)}$, l'algoritmo per calcolare $x^{(l+1)}$ è

$$\begin{aligned} \alpha_l &= \frac{{z^{(l)}}^{\mathrm{T}} r^{(l)}}{{p^{(l)}}^{\mathrm{T}} A p^{(l)}} \\ x^{(l+1)} &= x^{(l)} + \alpha_l p^{(l)} \\ r^{(l+1)} &= r^{(l)} - \alpha_l A p^{(l)} \\ Pz^{(l+1)} &= r^{(l+1)} \\ \beta_{l+1} &= \frac{{z^{(l+1)}}^{\mathrm{T}} r^{(l+1)}}{{z^{(l)}}^{\mathrm{T}} r^{(l)}} \\ p^{(l+1)} &= z^{(l+1)} + \beta_{l+1} p^{(l)} \end{aligned} \tag{3.6}$$

Teorema 1. *Il metodo del gradiente coniugato applicato ad una matrice di ordine N converge in al più N iterazioni (in aritmetica esatta).*

Dimostrazione. La dimostrazione (omessa) si basa essenzialmente sul fatto che $p^{(1)}, \ldots, p^{(N)}$ sono vettori linearmente indipendenti e non ce ne possono essere più di N. □

Per questo motivo, tale metodo è detto *semiiterativo*.

Stima dell'errore

Vale la seguente stima dell'errore:

$$\|e^{(l)}\|_A \leq 2 \left(\frac{\sqrt{\text{cond}_2(P^{-1}A)} - 1}{\sqrt{\text{cond}_2(P^{-1}A)} + 1} \right)^{l-1} \|e^{(1)}\|_A$$

dalle quale si osserva che

- la stima d'errore decresce in ogni caso, poiché il numeratore è più piccolo del denominatore;
- in particolare, nel caso $P = I$;
- tanto più è piccolo il numero di condizionamento di $P^{-1}A$, tanto più il metodo ha convergenza veloce;
- nel caso limite di $P = A$, si ha $\|e^{(l)}\|_A \leq 0$.

3.1.3 Test d'arresto

Un primo stimatore è costituito dal residuo: si arresta cioè il metodo iterativo quando

$$\|r^{(l)}\| \leq \text{tol} \cdot \|b\|$$

Infatti, dalla precedente si ricava

$$\frac{\|e^{(l)}\|}{\|x\|} \leq \text{tol} \cdot \text{cond}(A)$$

Una modifica consiste in

$$\|r^{(l)}\| \leq \text{tol} \cdot \|r^{(1)}\| \tag{3.7}$$

che coincide con il precedente nel caso in cui come $x^{(1)}$ venga scelto il vettore di zeri.

Capitolo 4

Memorizzazione di matrici sparse

Sia A una matrice sparsa di ordine N con m elementi diversi da zero. Esistono molti formati di memorizzazione di matrici sparse. Quello usato da GNU Octave è il Compressed Column Storage (CCS). Consiste di tre array: un primo, `data`, di lunghezza m contenente gli elementi diversi da zero della matrice, ordinati prima per colonna e poi per riga; un secondo, `ridx`, di lunghezza m contenente gli indici di riga degli elementi di `data`; ed un terzo, `cidx`, di lunghezza $N+1$, il cui primo elemento è 0 e l'elemento $i+1$-esimo è il numero totale di elementi diversi da zero nelle prime i colonne della matrice. Per esempio, alla matrice

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 \\ 4 & 0 & 5 & 6 \\ 0 & 0 & 0 & 7 \end{pmatrix}$$

corrispondono i vettori

$$\texttt{data} = [1, 4, 2, 3, 5, 6, 7]$$
$$\texttt{ridx} = [1, 3, 2, 2, 3, 3, 4]$$
$$\texttt{cidx} = [0, 2, 3, 5, 7]$$

In GNU Octave, il formato CCS e l'implementazione del prodotto matrice-vettore sono automaticamente usati dalla function `sparse` e dall'operatore `*`, rispettivamente.

4.1 Alcuni comandi per matrici sparse

- Il comando `speye(N)` genera la matrice identità di ordine N.

- Il comando `spdiags(v,0,N,N)`, ove v è un vettore colonna, genera la matrice diagonale di ordine n avente v in diagonale. Se la dimensione di v è minore di n, la diagonale viene riempita con zeri posti dopo il vettore v. Se invece la dimensione di v è maggiore di N, vengono usate solo le prime N componenti di v.

  Sia V la matrice

$$V = \begin{pmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ \vdots & \vdots & \vdots \\ v_{N1} & v_{N2} & v_{N3} \end{pmatrix}$$

  Il comando `spdiags(V,-1:1,N,N)` genera la matrice

$$\begin{pmatrix} v_{12} & v_{23} & 0 & 0 & \dots & 0 \\ v_{11} & v_{22} & v_{33} & 0 & \dots & 0 \\ 0 & v_{21} & v_{32} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & v_{N-2\,1} & v_{N-1\,2} & v_{N3} \\ 0 & \dots & \dots & 0 & v_{N-1\,1} & v_{N2} \end{pmatrix}$$

Capitolo 5

Sistemi tridiagonali

La risoluzione di sistemi tridiagonali

$$Ax = b$$

con

$$A = \begin{bmatrix} a_1 & c_1 & 0 & \dots & \dots & 0 \\ b_1 & a_2 & c_2 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & b_{n-2} & a_{n-1} & c_{n-1} \\ 0 & \dots & \dots & 0 & b_{n-1} & a_n \end{bmatrix}$$

risulta particolarmente economica. Infatti, nel caso non sia necessario il pivoting, si ha $A = LU$, ove

$$L = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ \beta_1 & 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \beta_{n-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} \alpha_1 & c_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & c_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \alpha_{n-1} & c_{n-1} \\ 0 & \dots & \dots & 0 & \alpha_n \end{bmatrix}$$

con

$$\begin{cases} \alpha_1 = a_1 \\ \beta_{k-1} = b_{k-1}/\alpha_{k-1}, \quad \alpha_k = a_k - \beta_{k-1}c_{k-1}, \quad k = 2, 3, \dots, n \end{cases}$$

e dunque la fattorizzazione LU costa $\mathcal{O}(2n)$ flops. A questo punto si risolvono i due sistemi $Ly = b$ e $Ux = y$, mediante

$$\begin{cases} y_1 = b_1 \\ y_k = b_k - \beta_{k-1}y_{k-1}, \quad k = 2, 3, \dots n \end{cases}$$

e

$$\begin{cases} x_n = y_n/\alpha_n \\ x_k = (y_k - c_k x_{k+1})/\alpha_k, \quad k = n-1, n-2, \dots 1 \end{cases}$$

con un ulteriore costo $\mathcal{O}(2n)$ flops. GNU Octave usa automaticamente questo algoritmo per le matrici tridiagonali.

Capitolo 6

Metodo di Newton per sistemi di equazioni non lineari

Consideriamo il sistema di equazioni non lineari

$$\begin{cases} f_1(x_1, x_2, \ldots, x_N) = 0 \\ f_2(x_1, x_2, \ldots, x_N) = 0 \\ \vdots \\ f_N(x_1, x_2, \ldots, x_N) = 0 \end{cases}$$

che può essere riscritto, in forma compatta,

$$f(x) = 0 \, .$$

Dato $x^{(1)}$, il metodo di Newton per calcolare $x^{(l+1)}$ è

$$\begin{aligned} J^{(l)} \delta x^{(l)} &= -f(x^{(l)}) \\ x^{(l+1)} &= x^{(l)} + \delta x^{(l)} \end{aligned} \tag{6.1}$$

ove $J^{(l)}$ è la matrice Jacobiana, definita da

$$J_{ij}^{(l)} = \frac{\partial f_i(x^{(l)})}{\partial x_j^{(l)}} \, . \tag{6.2}$$

Il criterio d'arresto solitamente usato è

$$\|\delta x^{(l)}\| \leq \text{tol}$$

6.1 Metodo di Newton modificato

Il metodo di Newton (6.1) richiede il calcolo della matrice Jacobiana e la sua "inversione" ad ogni passo k. Questo potrebbe essere troppo oneroso. Una strategia per ridurre il costo computazionale è usare sempre la stessa matrice Jacobiana $J^{(1)}$, oppure aggiornarla solo dopo un certo numero di iterazioni. In tal modo, per esempio, è possibile usare la stessa fattorizzazione $L^{(l)}U^{(l)}$ per più iterazioni successive.

Capitolo 7

Esponenziale di matrice

Data una matrice quadrata $A \in \mathbb{R}^{N\times N}$, si definisce

$$\exp(A) = \sum_{j=0}^{\infty} \frac{A^j}{j!}$$

Tale serie converge per qualunque matrice A, essendo A un operatore lineare tra spazi di Banach e avendo la serie esponenziale raggio di convergenza ∞. Se A e B sono *permutabili* (cioè $AB = BA$), allora

$$\exp(A+B) = \exp(A)\exp(B)$$

7.1 Formula delle *variazioni delle costanti*

Data l'equazione differenziale

$$\begin{cases} y'(t) = ay(t) + b(y(t)), \quad t > 0 \\ y(t_0) = y_0 \end{cases} \tag{7.1}$$

$y(t) \in \mathbb{R}$, la soluzione può essere scritta analiticamente mediante la formula delle *variazioni delle costanti*

$$y(t) = e^{(t-t_0)a}y_0 + \int_{t_0}^{t} e^{(t-\tau)a}b(y(\tau))\mathrm{d}\tau \tag{7.2}$$

Infatti, si ha

$$y'(t) = ae^{(t-t_0)a}y_0 + a\int_{t_0}^{t} e^{(t-\tau)a}b(y(\tau))\mathrm{d}\tau + e^{(t-t)a}b(y(t)) = ay(t) + b(y(t))$$

Si osservi che

$$\begin{aligned}\int_{t_0}^{t} e^{(t-\tau)a}\mathrm{d}\tau &= -\frac{1}{a}\int_{t_0}^{t} -ae^{(t-\tau)a}\mathrm{d}\tau = -\frac{1}{a}e^{(t-\tau)a}\Big|_{t_0}^{t} = \\ &= -\frac{1}{a}\left(1-e^{(t-t_0)a}\right) = (t-t_0)\frac{e^{(t-t_0)a}-1}{(t-t_0)a} = \\ &= (t-t_0)\varphi_1((t-t_0)a)\ ,\end{aligned}$$

ove

$$\varphi_1(z) = \frac{e^z-1}{z} = \sum_{j=0}^{\infty}\frac{z^j}{(j+1)!} \tag{7.3}$$

e, analogamente,

$$\int_{t_0}^{t} e^{(t-\tau)a}(\tau-t_0)\mathrm{d}\tau = (t-t_0)^2\varphi_2((t-t_0)a)$$

ove

$$\varphi_2(z) = \frac{e^z-1-z}{z^2} = \sum_{j=0}^{\infty}\frac{z^j}{(j+2)!} \tag{7.4}$$

Consideriamo ora un sistema differenziale

$$\begin{cases} \boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{b}(\boldsymbol{y}(t)), \quad t>0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

Ancora, la soluzione esplicita può essere scritta come

$$\boldsymbol{y}(t) = \exp((t-t_0)A)\boldsymbol{y}_0 + \int_{t_0}^{t}\exp((t-\tau)A)\boldsymbol{b}(\boldsymbol{y}(\tau))\mathrm{d}\tau$$

7.2 Calcolo di $\exp(A)$

Come per la risoluzione di sistemi lineari, non esiste *il* modo per calcolare $\exp(A)$, ma diversi modi, ognuno adatto a particolari situazioni.

7.2.1 Matrici piene, di modeste dimensioni

Questi metodi si applicano, in pratica, a quelle matrici per le quali si usano i metodi diretti per la risoluzione di sistemi lineari.

Decomposizione spettrale Se la matrice è diagonalizzabile, cioè $A = VDV^{-1}$, allora $\exp(A) = V\exp(D)V^{-1}$, ove $\exp(D)$ è la matrice diagonale con elementi $e^{d_1}, e^{d_2}, \ldots, e^{d_N}$. Basta infatti osservare che

$$A^2 = (VDV^{-1})^2 = (VDV^{-1})(VDV^{-1}) = VD^2V^{-1}$$

e scrivere $\exp(A)$ come serie di Taylor. La decomposizione spettrale di una matrice costa, in generale, $\mathcal{O}(N^3)$. Si ottiene in GNU Octave con il comando `eig`.

Approssimazione razionale di Padé Si considera un'approssimazione razionale della funzione esponenziale

$$e^z \approx \frac{a_1 z^{p-1} + a_2 z^{p-2} + \ldots + a_p}{b_1 z^{q-1} + b_2 z^{q-2} + \ldots + b_q} , \tag{7.5}$$

ove $b_q = 1$ per convenzione. Essa è chiamata *diagonale* quando $p = q$. Si può dimostrare che le approssimazioni diagonali sono le più efficienti. Fissato il grado di approssimazione, si sviluppa in serie di Taylor la funzione esponenziale e si fanno coincidere quanti più coefficienti possibile. Per esempio, fissiamo $p = q = 2$. Si ha allora

$$\left(1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \ldots\right)(b_1 z + 1) = a_1 z + a_2$$

$$b_1 z + 1 + b_1 z^2 + z + \frac{z^2}{2} + o(z^2) = a_1 z + a_2$$

da cui

$$\begin{cases} 1 = a_2 \\ b_1 + 1 = a_1 \\ b_1 + \dfrac{1}{2} = 0 \end{cases}$$

L'approssimazione di Padé si estende banalmente al caso matriciale. Considerando sempre il caso $p = q = 2$, si ha

$$\exp(A) \approx B = (b_1 A + I)^{-1}(a_1 A + a_2 I) ,$$

cioè B è soluzione del sistema lineare $(b_1 A + I)B = a_1 A + a_2 I$.

L'approssimazione di Padé è accurata solo quando $|z| < 1/2$ (o, nel caso matriciale, $\|A\|_2 < 1/2$). Per la funzione esponenziale esiste una tecnica, chiamata *scaling and squaring* che permette di aggirare il problema. Si usa infatti la proprietà

$$e^z = \left(e^{z/2}\right)^2 = \left(e^{z/2^j}\right)^{2^j}$$

Se $|z| > 1/2$, allora $|z|/2^j < 1/2$ per $j > \log_2(|z|) + 1$. Si calcola dunque l'approssimazione di Padé di $e^{z/2^j}$ e poi si eleva al quadrato j volte. Per la funzione φ_1 vale

$$\varphi_1(z) = \frac{1}{2}(e^{z/2} + 1)\varphi_1\left(\frac{z}{2}\right)$$

Anche l'approssimazione di Padé matriciale ha costo $\mathcal{O}(N^3)$. In GNU Octave si usa una variante di questa tecnica nel comando `expm`.

7.2.2 Matrici sparse, di grandi dimensioni

I metodi visti nel paragrafo precedente ignorano l'eventuale sparsità delle matrici. Inoltre, negli integratori esponenziali, non è mai richiesto di calcolare esplicitamente funzioni di matrice, ma solo funzioni di matrice applicate a vettori, cioè $\exp(A)v$ (è l'analoga differenza tra calcolare A^{-1} e $A^{-1}v$). Si possono allora usare dei metodi *iterativi*.

Metodo di Krylov Mediante la *tecnica di Arnoldi* è possibile, tramite prodotti matrice-vettore, decomporre A in $A \approx V_m H_m V_m^{\mathrm{T}}$, ove $V_m \in \mathbb{R}^{n\times m}$, $V_m^{\mathrm{T}} V_m = I$, $V_m e_1 = v$ e H_m è matrice di Hessenberg di ordine m (con $m \ll N$). Allora $AV_m \approx V_m H_m$ e

$$\exp(A)v \approx V_m \exp(H_m) e_1 \ .$$

Il calcolo di $\exp(H_m)$ è fatto mediante l'approssimazione di Padé. Il costo della tecnica di Arnoldi è $\mathcal{O}(Nm^2)$ se A è matrice sparsa. È necessario inoltre memorizzare la matrice V_m.

Interpolazione su nodi di Leja Se il polinomio $p_m(z)$ interpola e^z nei nodi $\xi_0, \xi_1, \ldots, \xi_m$, allora $p_m(A)v$ è una approssimazione di $\exp(A)v$. È una *buona* approssimazione se i nodi sono buoni (*non* equispaziati, per esempio) e se sono contenuti nell'involucro convesso dello spettro di A. È difficile stimare a priori il grado di interpolazione m necessario. È conveniente usare la formula di interpolazione di Newton

$$p_{m-1}(z) = d_1 + d_2(z-\xi_1) + d_3(z-\xi_1)(z-\xi_2) + \cdot\ldots\cdot + d_m(z-\xi_1)\cdots(z-\xi_{m-1})$$

ove $\{d_i\}_i$ sono le differenze divise. Tale formula si può scrivere, nel caso matriciale,

$$p_{m-1}(A)v = p_{m-2}v + d_m w_m, \quad w_m = \left(\prod_{i=1}^{m-1}(A - \xi_i I)\right) v = (A - \xi_{m-1})w_{m-1}$$

Dunque, la complessità è $\mathcal{O}(Nm)$ è richiesta la memorizzazione di un solo vettore w.

Quali nodi usare? I nodi di Chebyshev, molto buoni per l'interpolazione, non possono essere usati, in quanto non permettono un uso efficiente della formula di interpolazione di Newton (cambiano tutti al cambiare del grado). I *nodi di Leja* sono distribuiti asintoticamente come i nodi di Chebyshev e, dati i primi $m-1$, ξ_m è il nodo per cui

$$\prod_{i=1}^{m-1} |\xi_m - \xi_i| = \max_{\xi \in [a,b]} \prod_{i=1}^{m-1} |\xi - \xi_i| \ ,$$

ove l'intervallo $[a,b]$ è in relazione con lo spettro di A, per esempio $[a,b] = \sigma(A) \cap \{y = 0\}$. Il primo nodo coincide, solitamente, con l'estremo dell'intervallo $[a,b]$ di modulo massimo. È chiaro che l'insieme dei primi m nodi di Leja coincide con l'unione di $\{\xi_m\}$ con l'insieme dei primi $m-1$ nodi di Leja.

Capitolo 8

Esercizi

1. Implemetare le functions `[data,ridx,cidx] = full2ccs(A)` e `[A] = ccs2full(data,ridx,cidx)` e le functions che, dati `data`, `ridx` e `cidx`, implementano i prodotti matrice vettore Ax e $A^{\mathrm{T}}x$.

2. Si risolvano 6 sistemi lineari con le matrici di Hilbert di ordine $N = 4, 6, 8, 10, 12, 14$ (`hilb(N)`) e termine noto scelto in modo che la soluzione esatta sia il vettore $[1, 1, \ldots, 1]^{\mathrm{T}}$ usando il comando \ di GNU Octave, il metodo del gradiente precondizionato e il metodo del gradiente coniugato precondizionato. Per questi ultimi due, si usi una tolleranza pari a 10^{-6}, un numero massimo di iterazioni pari a 2000, il precondizionatore diagonale e un vettore iniziale $x^{(1)}$ di zeri. Si riporti, per ogni N, il numero di condizionamento della matrice, l'errore in norma infinito rispetto alla soluzione esatta e il numero di iterazioni dei metodi iterativi.

3. Risolvere il sistema non lineare

$$\begin{cases} f_1(x_1, x_2) = x_1^2 + x_2^2 - 1 = 0 \\ f_2(x_1, x_2) = \sin(\pi x_1/2) + x_2^3 = 0 \end{cases}$$

   con il metodo di Newton (6.1). Si usi una tolleranza pari a 10^{-6}, un numero massimo di iterazioni pari a 150 e un vettore iniziale $x^{(1)} = [1, 1]^{\mathrm{T}}$. Si risolva lo stesso sistema non lineare usando sempre la matrice Jacobiana relativa al primo passo e aggiornando la matrice Jacobiana ogni r iterazioni, ove r è il più piccolo numero di iterazioni che permette di ottenere la stessa soluzione con la tolleranza richiesta calcolando solo due volte la matrice Jacobiana.

4. Si implementi una function `[a,b] = padeexp(p)` che restituisce i coef-

ficienti dell'approssimazione razionale di Padé (7.5) (con $p = q$) per la funzione esponenziale.

Parte 1

BVPs
(Problemi ai limiti)

Capitolo 9

Introduzione

Consideriamo il seguente *problema ai limiti (boundary value problem)*

$$\begin{cases} u''(x) = f(x, u(x), u'(x)), \quad x \in (a, b) \\ u(a) = u_a \\ u(b) = u_b \end{cases} \tag{9.1}$$

ove $u(x) \in \mathbb{R}$. Le condizioni ai bordi sono di *Dirichlet* quando viene prescritto il valore della soluzione $u(x)$ e di *Neumann* quando viene prescritto il valore della derivata della soluzione $u'(x)$. Si possono avere anche condizioni *miste*, ad esempio

$$\begin{cases} u''(x) = f(x, u(x), u'(x)), \quad x \in (a, b) \\ u(a) = u_a \\ u'(b) = u'_b \end{cases}$$

Quando i valori prescritti sono nulli, si parla di condizioni *omogenee*.

Capitolo 10

Differenze finite

10.1 Differenze finite centrate del secondo ordine

Sia $u \in \mathcal{C}^3([a,b])$ e $x_i = a + (i-1)h$, $1 \leq i \leq m$, $h = (b-a)/(m-1)$. Sviluppando in serie, si ha

$$u(x_{i+1}) = u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i) + \frac{h^3}{6}u^{(3)}(\hat{x}_i)$$
$$u(x_{i-1}) = u(x_i) - hu'(x_i) + \frac{h^2}{2}u''(x_i) - \frac{h^3}{6}u^{(3)}(\tilde{x}_i)$$

da cui

$$u'(x_i) = \Delta u_i - \tau_i^{(1)} = \frac{u(x_{i+1}) - u(x_{i-1})}{2h} - \tau_i^{(1)}$$

ove $\tau_i^{(1)} = \frac{h^2}{6}u^{(3)}(\bar{x}_i)$ è l'*errore locale* ($u^{(3)}(\hat{x}_i) + u^{(3)}(\tilde{x}_i) = 2u^{(3)}(\bar{x}_i)$, per un opportuno $\bar{x}_i$, per il teorema dei valori intermedi). Analogamente, sia $u \in \mathcal{C}^4([a,b])$. Si ha

$$u(x_{i+1}) = u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i) + \frac{h^3}{6}u^{(3)}(x_i) + \frac{h^4}{24}u^{(4)}(\hat{x}_i)$$
$$u(x_{i-1}) = u(x_i) - hu'(x_i) + \frac{h^2}{2}u''(x_i) - \frac{h^3}{6}u^{(3)}(x_i) + \frac{h^4}{24}u^{(4)}(\tilde{x}_i)$$

da cui

$$u''(x_i) = \Delta^2 u_i - \tau_i^{(2)} = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} - \tau_i^{(2)}$$

ove $\tau_i^{(2)} = \frac{h^2}{12}u^{(4)}(\bar{x}_i)$. Queste approssimazioni della derivata prima e seconda di chiamano *differenze finite centrate del secondo ordine*. Il termine "centrate" si riferisce al fatto che i punti x_i sono equispaziati e si usano i valori

della funzione $u(x)$ in uno stesso numero di punti a sinistra e a destra di x_i per ricavare un'approssimazione delle derivate. Il termine "secondo ordine" si riferisce al fatto che l'errore locale è proporzionale alla seconda potenza del *passo di discretizzazione* h. Ovviamente sono possibili altri tipi di approssimazione, basati su nodi non equispaziati, non centrate e di ordine diverso.

Una volta scelto il tipo di discretizzazione, invece del problema originale (9.1) si risolve il problema discretizzato

$$\begin{cases} \Delta^2 u_i = f(x_i, u_i, \Delta u_i), \quad 2 \leq i \leq m-1 \\ u_1 = u_a \\ u_m = u_b \end{cases}$$

nell'incognita $u_h = [u_1, u_2, \ldots, u_{m-1}, u_m]^{\mathrm{T}}$.

10.2 Convergenza per un problema modello

Consideriamo il seguente problema modello *(elasticità della trave)*

$$\begin{cases} -u''(x) + q(x)u(x) = g(x), \quad x \in (a,b) \\ u(a) = u_a \\ u(b) = u_b \end{cases} \tag{10.1}$$

con $q, f \in \mathcal{C}^0([a,b])$, $q(x) \geq 0$ per $x \in [a,b]$. La funzione $q(x)$ dipende dal materiale di cui è fatta la trave e $f(x)$ è la densità di carico trasversale. La soluzione $u(x)$ rappresenta il momento flettente. Vogliamo studiare l'esistenza, l'unicità e la regolarità della soluzione analitica.

10.2.1 Unicità

Se $u_1(x)$ e $u_2(x)$ sono due soluzioni di (10.1), allora $z(x) = u_1(x) - u_2(x)$ soddisfa il problema *omogeneo*

$$\begin{cases} -z''(x) + q(x)z(x) = 0, \quad x \in (a,b) \\ z(a) = 0 \\ z(b) = 0 \end{cases} \tag{10.2}$$

Proposizione 1. *Se $z(x)$ è soluzione di (10.2), allora $z(x) \equiv 0$.*

Dimostrazione (metodo dell'energia). Moltiplicando l'equazione per $z(x)$ ed integrando si ha

$$\begin{aligned} 0 &= \int_a^b -z''(x)z(x)\mathrm{d}x + \int_a^b q(x)z(x)^2\mathrm{d}x = \\ &= [-z'(x)z(x)]_a^b + \int_a^b z'(x)^2\mathrm{d}x + \int_a^b q(x)z(x)^2\mathrm{d}x = \\ &= \int_a^b z'(x)^2\mathrm{d}x + \int_a^b q(x)z(x)^2\mathrm{d}x \end{aligned}$$

Poiché le funzioni integrande sono non negative, si ha che deve essere necessariamente $z(x) \equiv 0$. □

Dunque, $u_1(x) \equiv u_2(x)$.

10.2.2 Esistenza

Sia $z(x) = c_1 z_1(x) + c_2 z_2(x)$ la soluzione generale di $-z''(x) + q(x)z(x) = 0$. La soluzione di (10.2) si ottiene imponendo

$$\begin{cases} c_1 z_1(a) + c_2 z_2(a) = 0 \\ c_1 z_1(b) + c_2 z_2(b) = 0 \end{cases}$$

Poiché sappiamo che $z(x) \equiv 0$ è l'unica soluzione, si ha che la matrice

$$\begin{bmatrix} z_1(a) & z_2(a) \\ z_1(b) & z_2(b) \end{bmatrix}$$

è non singolare.

La soluzione generale di $-u''(x) + q(x)u(x) = f(x)$ è $u(x) = c_1 z_1(x) + c_2 z_2(x) + s(x)$ ($s(x)$ soluzione particolare). La soluzione di (10.1) si ottiene imponendo

$$\begin{cases} c_1 z_1(a) + c_2 z_2(a) = u_a - s(a) \\ c_1 z_1(b) + c_2 z_2(b) = u_b - s(b) \end{cases}$$

cioè risolvendo un sistema lineare non singolare che ammette dunque (unica) soluzione.

Si è costretti a ridursi ad un problema modello perché problemi ai limiti anche molto semplici possono non avere soluzione: si consideri, per esempio,

$$\begin{cases} u''(x) + u(x) = 0 \\ u(0) = 0 \\ u(\pi) = 1 \end{cases}$$

La soluzione generale è $c_1\cos(x) + c_2\sin(x)$, ma non è possibile imporre le condizioni al bordo.

10.2.3 Regolarità

Proposizione 2. *Se $q, f \in \mathcal{C}^k([a,b])$, allora $u \in \mathcal{C}^{k+2}([a,b])$.*

Dimostrazione. Se $q, f \in \mathcal{C}^0([a,b])$, poiché la soluzione u esiste, u'' è definita in ogni punto $x \in [a,b]$, e dunque u' esiste (ed è derivabile). Quindi $u \in \mathcal{C}^0([a,b])$ e quindi $u'' \in \mathcal{C}^0([a,b])$. Dunque $u \in \mathcal{C}^2([a,b])$. Sia vero adesso l'enunciato per k e siano $q, g \in \mathcal{C}^{k+1}([a,b])$: poiché anche $u \in \mathcal{C}^{k+1}([a,b])$, si ha $u'' \in \mathcal{C}^{k+1}([a,b])$ da cui $u \in \mathcal{C}^{k+3}([a,b])$. □

Ci occupiamo adesso di analizzare la convergenza del problema discretizzato mediante differenze finite centrate del secondo ordine, che si scrive

$$\begin{cases} -\dfrac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + q_i u_i = f_i, \quad 2 \le i \le m-1 \\ u_1 = u_a \\ u_m = u_b \end{cases}$$

ove $q_i = q(x_i)$ e $f_i = f(x_i)$.

10.2.4 Consistenza

Se si sostituisce u_i con la soluzione analitica $u(x_i)$, si ottiene

$$\begin{cases} -\dfrac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} + q(x_i)u(x_i) - f(x_i) = -\tau_i^{(2)}, \ 2 \le i \le m-1 \\ u(x_1) = u_a \\ u(x_m) = u_b \end{cases}$$

da cui si deduce che il metodo numerico è *consistente* di ordine 2. Il sistema lineare da risolvere per trovare $u_h = [u_1, u_2, \ldots, u_{m-1}, u_m]^{\mathrm{T}}$ è

$$\frac{1}{h^2}\begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ -1 & 2+q_2h^2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2+q_3h^2 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 2+q_{m-1}h^2 & -1 \\ 0 & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix} = \begin{bmatrix} u_a/h^2 \\ f_2 \\ f_3 \\ \vdots \\ f_{m-1} \\ u_b/h^2 \end{bmatrix}$$

e può essere semplificato in

$$\frac{1}{h^2}\begin{bmatrix} 2+q_2h^2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2+q_3h^2 & -1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -1 & 2+q_{m-2}h^2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2+q_{m-1}h^2 \end{bmatrix} \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ \vdots \\ u_{m-2} \\ u_{m-1} \end{bmatrix} = \begin{bmatrix} f_2+u_a/h^2 \\ f_3 \\ \vdots \\ \vdots \\ f_{m-2} \\ g_{m-1}+u_b/h^2 \end{bmatrix}$$

cioè

$$A_h u_h = f_h \tag{10.3}$$

10.2.5 Esistenza ed unicità

Proposizione 3. *Il sistema lineare* (10.3) *è non singolare e dunque ammette un'unica soluzione.*

Dimostrazione (metodo dell'energia discreto). Dato $z = [z_2, z_3, \dots, z_{m-1}]^{\mathrm{T}}$, consideriamo il prodotto $z^{\mathrm{T}} A_h z$. Si ha

$$\begin{aligned} z^{\mathrm{T}} A_h z = \frac{1}{h^2}[&(2+q_2h^2)z_2^2 - z_2z_3 - z_3z_2 + (2+q_3h^2)z_3^2 - z_3z_4 + \dots + \\ &+ \dots - z_{m-1}z_{m-2} + (2+q_{m-1}h^2)z_{m-1}^2] = \\ &= \frac{1}{h^2}[z_2^2 + (z_2-z_3)^2 + (z_3-z_4)^2 + \dots + (z_{m-2}-z_{m-1})^2 + z_{m-1}^2] + \\ &+ \sum_{i=2}^{m-1} q_i z_i^2 \geq 0 \end{aligned}$$

Poiché si ha una somma di elementi non negativi, l'uguaglianza a 0 si può avere solo quando tutti gli elementi sono nulli e quindi per solo per z nullo. Dunque la matrice A_h è definita positiva e quindi non singolare. □

10.2.6 Proprietà di A_h

A_h è una matrice simmetrica e diagonalmente dominante. È possibile usare i metodi iterativi, semi-iterativi e diretti *senza* pivoting per la soluzione del sistema lineare. Inoltre, è una M-matrice, cioè i suoi elementi extra-diagonali sono non positivi e la sua inversa ha elementi non negativi.

10.2.7 Stabilità

Consideriamo due soluzioni relative a dati perturbati $\tilde{f}_h$ e $\hat{f}_h$. Si ha

$$A_h\tilde{u}_h = \tilde{f}_h$$
$$A_h\hat{u}_h = \hat{f}_h$$

da cui

$$(\tilde{u}_h - \hat{u}_h) = A_h^{-1}(\tilde{f}_h - \hat{f}_h)$$

Se si vuole che le perturbazioni sui dati non si ripercuotano in maniera distruttiva sulle soluzioni, occorre che la matrice A_h^{-1} sia limitata in norma *indipendentemente* da h, in particolare per $h \to 0$. Consideriamo la matrice $A_{h,q=0}$ corrispondente alla stessa discretizzazione nel caso $q(x) \equiv 0$. Si ha $A_h - A_{h,q=0} = \text{diag}(q_2, \ldots, q_{m-1}) \geq 0$. Allora

$$A_{h,q=0}^{-1} - A_h^{-1} = A_{h,q=0}^{-1}(A_h - A_{h,q=0})A_h^{-1} \geq 0$$

perché $A_{h,q=0}$ e A_h sono M-matrici. Allora $A_h^{-1} \leq A_{h,q=0}^{-1}$. Osserviamo poi che $A_0^{-1}[1, \ldots, 1]^{\mathrm{T}}$ è la soluzione discreta (approssimata) di

$$\begin{cases} -v''(x) = 1 \\ v(a) = 0 \\ v(b) = 0 \end{cases}$$

la cui soluzione analitica è $v(x) = (x-a)(b-x)/2$. Poiché $v^{(3)}(x) \equiv 0$, così è per $v^{(4)}$ e dunque l'errore locale, per questo problema, è 0. Dunque

$$\|A_{h,q=0}^{-1}\|_\infty = \|A_{h,q=0}^{-1}[1, \ldots, 1]^{\mathrm{T}}\|_\infty = \max_{i=2,\ldots,m-1} v_i =$$
$$= \max_{i=2,\ldots,m-1} v(x_i) \leq \max_{x\in[a,b]} v(x) \leq \frac{(b-a)^2}{8}$$

e poiché $\|A_h^{-1}\|_\infty \leq \|A_{h,q=0}^{-1}\|_\infty$, si ha la maggiorazione richiesta.

10.2.8 Convergenza

Definiamo $e_h = [e_{2,h}, \ldots, e_{m-1,h}]^{\mathrm{T}} = [u_{2,h} - u(x_2), \ldots, u_{m-1,h} - u(x_{m-1})]^{\mathrm{T}}$. Poiché

$$A_h[u_{2,h}, \ldots, u_{m-1,h}]^{\mathrm{T}} = f_h$$
$$A_h[u(x_2), \ldots, u(x_{m-1})]^{\mathrm{T}} = f_h - \tau_h^{(2)}$$

ove $\tau_h^{(2)} = [\tau_{2,h}^{(2)}, \ldots, \tau_{m-1,h}^{(2)}]^{\mathrm{T}}$, si deduce $\boldsymbol{e}_h = A_h^{-1}\tau_h$. Combinando i risultati di consistenza e stabilità, si ottiene, per il problema (10.1) discretizzato mediante differenze finite centrate del secondo ordine,

$$\|e_h\|_\infty \leq \frac{(b-a)^2}{8}\frac{h^2}{12}\|u^{(4)}\|_\infty$$

e dunque l'errore è proporzionale a h^2, posto che $u \in \mathcal{C}^4([a,b])$.

10.3 Differenze finite non equispaziate

Dati tre nodi x_{i-1}, x_i, x_{i+1}, con $h_{i-1} = x_i - x_{i-1}$ e $h_i = x_{i+1} - x_i$, si ha

$$u(x_{i+1}) = u(x_i) + h_i u'(x_i) + \frac{h_i^2}{2}u''(x_i) + \frac{h_i^3}{6}u^{(3)}(x_i) + \mathcal{O}(h_i^4)$$
$$u(x_{i-1}) = u(x_i) - h_{i-1}u'(x_i) + \frac{h_{i-1}^2}{2}u''(x_i) - \frac{h_{i-1}^3}{6}u^{(3)}(x_i) + \mathcal{O}(h_{i-1}^4)$$

da cui

$$u'(x_i) = \frac{u(x_{i+1}) - u(x_{i-1})}{h_{i-1}+h_i} - \frac{h_i^2 - h_{i-1}^2}{h_{i-1}+h_i}u''(x_i) - \frac{h_{i-1}^3 + h_i^3}{h_{i-1}+h_i}u^{(3)}(x_i) + \\ + \mathcal{O}(\max\{h_{i-1}^4, h_i^4\})$$

Se h_{i-1} e h_i non differiscono troppo (precisamente, se la loro differenza è $\mathcal{O}(\max\{h_{i-1}^2, h_i^2\})$), allora l'approssimazione con il rapporto incrementale centrato è di ordine $\mathcal{O}(\max\{h_{i-1}^2, h_i^2\})$. Analogamente, si può costruire un'approssimazione della derivata seconda

$$u''(x_i) \approx \frac{\frac{u(x_{i+1})-u(x_i)}{h_i} - \frac{u(x_i)-u(x_{i-1})}{h_{i-1}}}{\frac{h_{i-1}+h_i}{2}}$$

La matrice corrispondente all'approssimazione mediante differenze finite di ordine due della derivata prima con griglia *non* equispaziata è (senza tener condo delle condizioni ai bordi)

$$\begin{bmatrix} u'(x_1) \\ u'(x_2) \\ u'(x_3) \\ \vdots \\ u'(x_{m-1}) \\ u'(x_m) \end{bmatrix} \approx \begin{bmatrix} * & * & * & * & * & * \\ \frac{-1}{h_1+h_2} & 0 & \frac{1}{h_1+h_2} & 0 & \ldots & 0 \\ 0 & \frac{-1}{h_2+h_3} & 0 & \frac{1}{h_2+h_3} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \frac{-1}{h_{m-2}+h_{m-1}} & 0 & \frac{1}{h_{m-2}+h_{m-1}} \\ * & * & * & * & * & * \end{bmatrix} \begin{bmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \\ \vdots \\ u(x_{m-1}) \\ u(x_m) \end{bmatrix}$$

Dati i nodi `x` (vettore colonna di lunghezza m), è possibile costruire il vettore $[h_1, h_2, \ldots, h_{m-1}]^{\mathrm{T}}$ con il comando `h=diff(x)`. Allora la matrice, a meno della prima e dell'ultima riga, può essere costruita, direttamente in formato sparso, con i comandi

```
> d = 1./(h(1:m-2)+h(2:m-1));
> spdiags([[-d;0;0],[0;0;d]],[-1,1],m,m)
```

La costruzione della matrice relativa alla derivata seconda è analoga. Nel caso di griglia equispaziata, di passo h, le matrici relative alle approssimazione della derivata prima e seconda possono essere costruite con i comandi

```
> toeplitz(sparse(1,2,-1/(2*h),1,m),sparse(1,2,1/(2*h),1,m));
```

e

```
> toeplitz(sparse([1,1],[1,2],[-2/h^2,1/h^2],1,m));
```

rispettivamente.

10.4 Condizioni di Dirichlet

Se vengono prescritti i valori $u(a) = u_a$ o $u(b) = u_b$, conviene discretizzare, in un *primo momento*, il problema ai limiti senza tener conto delle condizioni al bordo. Per esempio, la discretizzazione del problema ai limiti

$$\begin{cases} u''(x) = 1, \quad x \in (a, b) \\ u(a) = u_a \\ u(b) = u_b \end{cases}$$

diventa

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \\ 1 \end{bmatrix}$$

Poi, si correggono le equazioni relative ai nodi al bordo

$$\frac{1}{h^2}\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix} = \frac{1}{h^2}\begin{bmatrix} u_a \\ 1 \\ \vdots \\ \vdots \\ 1 \\ u_b \end{bmatrix}$$

In questo modo, però, la simmetria della matrice viene persa. Pertanto, non è più possibile applicare gli appositi metodi per la risoluzione di sistemi lineari simmetrici. Un metodo *numericamente* equivalente è quello di modificare i soli elementi diagonali della prima e dell'ultima riga inserendo un numero molto grande

$$\frac{1}{h^2}\begin{bmatrix} M & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 1 & M \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix} = \begin{bmatrix} Mu_a \\ 1 \\ \vdots \\ \vdots \\ 1 \\ Mu_b \end{bmatrix}$$

10.5 Condizioni di Neumann

Se vengono prescritti i valori della derivata prima $u'(a) = u'_a$ o $u'(b) = u'_b$, è necessario approssimare la derivata prima con uno stencil non simmetrico, eventualmente con ordine di approssimazione minore. Per esempio, la discretizzazione del problema ai limiti

$$\begin{cases} u''(x) = 1, & x \in (a, b) \\ u'(a) = u'_a \\ u(b) = u_b \end{cases}$$

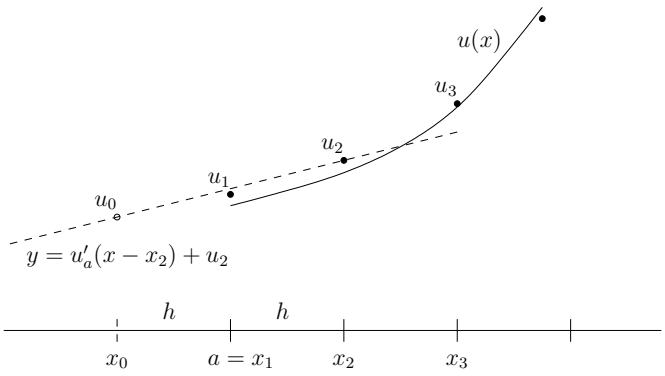
potrebbe essere

$$\frac{1}{h^2}\begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix} = \begin{bmatrix} u'_a/h \\ 10 \\ \vdots \\ \vdots \\ 1 \\ u_b/h^2 \end{bmatrix}$$

Volendo miglior accuratezza, si può usare

$$\frac{1}{h^2}\begin{bmatrix} -\frac{3}{2} & 2 & -\frac{1}{2} & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix} = \begin{bmatrix} u'_a/h \\ 0 \\ \vdots \\ \vdots \\ 0 \\ u_b/h^2 \end{bmatrix}$$

Infatti lo stencil in avanti $[-3, 4, -1]/(2h)$ produce una approssimazione del secondo ordine in h della derivata prima.



Un altro modo per avere ordine di accuratezza due (e che si può usare anche con i problemi parabolici, vedi paragrado 22.2.1) è quella di introdurre una variabile fittizia $u_0 \approx u(a-h)$ da porre uguale a $u_2 - 2hu'_a \approx u(x_2) - 2hu'_a$ (in modo che $(u_2 - u_0)/(2h) = u'_a$). In tal modo, la discretizzazione della derivata seconda nel primo nodo diventa

$$u''(a) \approx \frac{u_0 - 2u_1 + u_2}{h^2} = \frac{u_2 - 2hu'_a - 2u_1 + u_2}{h^2} = \frac{2u_2 - 2u_1 - 2hu'_a}{h^2}$$

e la prima riga del sistema $u''(a) = 1$ viene discretizzata da

$$\frac{2u_2 - 2u_1}{h^2} = 1 + \frac{2u'_a}{h}$$

10.6 Un esempio: l'equazione della catenaria

Consideriamo l'equazione della *catenaria*

$$\begin{cases} u''(x) = a\sqrt{1 + u'(x)^2}, & x \in (-1, 1) \\ u(-1) = 1 \\ u(1) = 1 \end{cases}$$

La discretizzazione mediante differenze finite centrate del secondo ordine è

$$A \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{m-1} \\ u_m \end{bmatrix} - a \begin{bmatrix} 1 \\ \sqrt{1 + \left(\frac{u_3 - u_1}{2h}\right)^2} \\ \vdots \\ \sqrt{1 + \left(\frac{u_m - u_{m-2}}{2h}\right)^2} \\ 1 \end{bmatrix} = \boldsymbol{b}$$

Si tratta dunque di risolvere il sistema non lineare

$$F(\boldsymbol{u}) = A\boldsymbol{u} - a\sqrt{1 + (B\boldsymbol{u})^2} - \boldsymbol{b} = 0$$

ove

$$A = \frac{1}{h^2}\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \frac{1}{2h}\begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 0 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix}$$

e $\boldsymbol{b} = [1/h^2 - a, 0, \ldots, 0, 1/h^2 - a]^{\mathrm{T}}$. Lo jacobiano di $F(\boldsymbol{u})$ è

$$J(\boldsymbol{u}) = A - aD(\boldsymbol{u})B, \quad D = (d_{ij}(\boldsymbol{u})), \quad d_{ij}(\boldsymbol{u}) = \begin{cases} \dfrac{(B\boldsymbol{u})_i}{\sqrt{1 + (B\boldsymbol{u})_i^2}}, & i = j \\ 0, & i \neq j \end{cases}$$

In generale,

$$JF(\boldsymbol{u}, B\boldsymbol{u}) = A\boldsymbol{u} - f(\boldsymbol{u}, B\boldsymbol{u}) = A - \mathrm{diag}\{f_{\boldsymbol{u}}(\boldsymbol{u}, B\boldsymbol{u})\} - \mathrm{diag}\{f_{B\boldsymbol{u}}(\boldsymbol{u}, B\boldsymbol{u})\}B$$

10.7 Norme

Data una funzione $u(x)$ e due diverse discretizzazioni su nodi equispaziati $[\tilde{u}_1, \ldots, \tilde{u}_m] \approx [u(\tilde{x}_1), \ldots, u(\tilde{x}_m)]$ e $[\hat{u}_1, \ldots, \hat{u}_l] \approx [u(\hat{x}_1), \ldots, u(\hat{x}_l)]$, $\{\tilde{x}_i\}_i \subset [a, b]$, $\{\hat{x}_i\}_i \subset [a, b]$, non ha molto senso confrontare gli errori $\|[u(\tilde{x}_1) - \tilde{u}_1, u(\tilde{x}_2) - \tilde{u}_2, \ldots, u(\tilde{x}_m) - \tilde{u}_m]\|_2$ e $\|[u(\hat{x}_1) - \hat{u}_1, y(\hat{x}_2) - \hat{u}_2, \ldots, u(\hat{x}_l) - \hat{u}_l]\|_2$. Si preferisce usare la norma infinito, oppure la norma $\|v\|_2\sqrt{\frac{b-a}{m}}$, che risulta essere una approssimazione mediante quadratura con formula dei rettangoli della norma in L^2 di $u(x)$.

Se si devono invece confrontare tra loro le due discretizzazioni, occorre che i nodi siano "intercalati".

Capitolo 11

Metodo di shooting

È possibile trasformare il problema (9.1) in un sistema differenziale del primo ordine

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad t \in (a, b]$$

tramite il cambiamento di variabili $t = x$, $y_1(t) = u(x)$, $y_2(t) = u'(x)$, $\boldsymbol{f}(t, \boldsymbol{y}(t)) = [y_2(t), f(t, y_1(t), y_2(t)]^{\mathrm{T}}$. Per quanto riguarda le condizioni iniziali, mentre quella per $y_1(t)$ è $y_1(a) = u_a$, quella per $y_2(t)$ non è definita. Si può allora introdurre un parametro $s \in \mathbb{R}$ e considerare la seguente famiglia di problemi ai valori iniziali

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad t \in (a, b] \\ y_1(a) = u_a \\ y_2(a) = s \end{cases} \tag{11.1}$$

Dato s, il sistema sopra può essere risolto con un opportuno metodo per problemi ai valori iniziali. Poiché s è il valore della derivata prima di $u(x)$, tale metodo di risoluzione prende il nome di *shooting*. Chiamiamo $y_1(t \mid y_2(a) = s)$ la prima componente della soluzione. Si dovrà ovviamente trovare $\bar{s}$ tale che $y_1(t \mid y_2(a) = \bar{s}) = u(x)$, $t = x \in [a, b]$. In particolare, dovrà essere $y_1(b \mid y_2(a) = \bar{s}) = u_b$. Introduciamo allora la funzione

$$F(s) = y_1(b \mid y_2(a) = s) - u_b$$

Si tratta di risolvere l'equazione (in genere non lineare) $F(s) = 0$.

11.1 Metodo di bisezione

Dati due valori s_1 e s_2 per cui $F(s_1)F(s_2) < 0$, è possibile applicare il metodo di bisezione per trovare lo zero di $F(s)$. Poiché la soluzione di (11.1) è

approssimata a meno di un errore dipendente dal passo di discretizzazione temporale, la tolleranza richiesta per il metodo di bisezione dovrà essere (leggermente) inferiore a tale errore.

11.2 Metodo di Newton

Per applicare il metodo di Newton, è necessario calcolare $F'(s)$. Definiamo a tal scopo

$$v(x) = \frac{\partial}{\partial s} u(x \mid u'(a) = s) = \frac{\partial}{\partial s} y_1(t \mid y_2(a) = s)$$

Derivando rispetto a s nel problema ai limiti

$$\begin{cases} u''(x) = f(x, u(x), u'(x)), & x \in (a, b) \\ u(a) = u_a \\ u'(a) = s \end{cases}$$

si ha

$$\frac{\partial}{\partial s} u''(x) = \frac{\partial}{\partial s} f(x, u(x), u'(x))$$

da cui, scambiando l'ordine di derivazione

$$v''(x) = f_u(x, u(x), u'(x))v(x) + f_{u'}(x, u(x), u'(x))v'(x), \quad x \in (a, b)$$

Per quanto riguarda le condizioni iniziali per $v(x)$, si ha

$$\begin{aligned} v(a) &= \frac{\partial}{\partial s} u(a \mid u'(a) = s) = \lim_{h \to 0} \frac{u(a \mid u'(a) = s + h) - u(a \mid u'(a) = s)}{h} = \\ &= \lim_{h \to 0} \frac{u_a - u_a}{h} = 0 \\ v'(a) &= \frac{\partial}{\partial s} u'(a \mid u'(a) = s) = \lim_{h \to 0} \frac{u'(a \mid u'(a) = s + h) - u'(a \mid u'(a) = s)}{h} = \\ &= \lim_{h \to 0} \frac{s + h - s}{h} = 1 \end{aligned}$$

Dunque, per calcolare $F'(s) = v(b)$ occorre risolvere il *sistema variazionale* (lineare in $v(x)$)

$$\begin{cases} v''(x) = f_u(x, u(x), u'(x))v(x) + f_{u'}(x, u(x), u'(x))v'(x), & x \in (a, b) \\ v(a) = 0 \\ v'(a) = 1 \end{cases}$$

In conclusione, per calcolare la coppia $F(s)$ e $F'(s)$ in un generico punto s, occorre risolvere il sistema differenziale del primo ordine ai dati iniziali

$$\begin{cases} y_1'(t) = y_2(t) \\ y_2'(t) = f(t, y_1(t), y_2(t)) \\ y_3'(t) = y_4(t) \\ y_4'(t) = f_{y_1}(t, y_1(t), y_2(t))y_3(t) + f_{y_2}(t, y_1(t), y_2(t))y_4(t) \\ y_1(a) = u_a \\ y_2(a) = s \\ y_3(a) = 0 \\ y_4(a) = 1 \end{cases}$$

fino al tempo $t = b$. Quindi $F(s) = y_1(b)$ e $F'(s) = y_3(b)$. Poiché le equazioni per $y_1'(t)$ e $y_2'(t)$ non dipendono da $y_3(t)$ e $y_4(t)$, è possibile disaccoppiare le prime due componenti dalle seconde due.

Una semplificazione del metodo di Newton che non richiede il calcolo di $F'(s)$ è il metodo delle secanti.

11.3 Problema ai limiti con frontiera libera

Un caso particolarmente interessante per l'applicazione del metodo di shooting è quello a frontiera libera *(free boundary)*

$$\begin{cases} u''(x) = f(x, u(x), u'(x)), \quad x \in (s, b) \\ u(s) = \alpha \\ u'(s) = \beta \\ u(b) = u_b \end{cases} \tag{11.2}$$

ove i valori di u e di u' sono assegnati in un punto incognito s, $s < b$. La funzione di cui si deve trovare lo zero è, in questo caso,

$$F(s) = u(b \mid u(s) = \alpha, u'(s) = \beta) - u_b$$

(scriveremo $F(s) = u(b \mid s) - u_b$ per brevità). Dati due punti s_1 e s_2 tali che $F(s_1)F(s_2) < 0$, l'applicazione del metodo di bisezione non presenta difficoltà. Per quanto riguarda il metodo di Newton, il sistema variazionale per

$$v(x) = \frac{\partial}{\partial s} u(x \mid s) = \lim_{h \to 0} \frac{u(x \mid s + h) - u(x \mid s)}{h}$$

è analogo al caso precedente. L'unica diversità è data dalle condizioni iniziali (in s). Si ha

$$v(s) = \lim_{h \to 0} \frac{u(s \mid s+h) - u(s \mid s)}{h}$$

Ora, $u(s \mid s) = \alpha$. Poi

$$u(s \mid s+h) = u(s+h \mid s+h) - hu'(s+h \mid s+h) + \mathcal{O}(h^2) = \alpha - h\beta + \mathcal{O}(h^2)$$

Dunque, $v(s) = -\beta$. In maniera analoga

$$v'(s) = \lim_{h \to 0} \frac{u'(s \mid s+h) - u'(s \mid s)}{h} = -u''(s)$$

ove il valore $u''(s)$ si ricava dal problema (11.2).

Capitolo 12

Equazione di Poisson

Di particolare interesse è l'equazione di Poisson

$$-\nabla^2 u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^d$$

ove ∇^2 è l'operatore *laplaciano* definito da

$$\nabla^2 = \sum_{k=1}^{d} \frac{\partial^2}{\partial x_k^2}$$

L'equazione è solitamente accompagnata da condizioni al bordo di Dirichlet o di Neumann.

12.1 Equazione di Poisson bidimensionale

Analizziamo numericamente in dettaglio il caso $d = 2$ ($\boldsymbol{x} = (x, y)$) e $\Omega = [a, b] \times [c, d]$.

12.1.1 Condizioni al bordo di Dirichlet

Consideriamo dapprima il caso con condizioni al bordo di Dirichlet. Dunque

$$\begin{cases} -\nabla^2 u(x, y) = f(x, y), \quad (x, y) \in [a, b] \times [c, d] \subset \mathbb{R}^2 \\ u(a, y) = D_a(y) \\ u(b, y) = D_b(y) \\ u(x, c) = D_c(x) \\ u(x, d) = D_d(x) \end{cases}$$

con le necessarie condizioni di compatibilità ai vertici. Introduciamo una discretizzazione $x_i = a + (i - 1)h_x$, $i = 1, 2, \ldots, m_x$, $h_x = (b - a)/(m_x - 1)$

e $y_j = c + (j-1)h_y$, $j = 1, 2, \ldots, m_y$, $h_y = (d-c)/(m_y - 1)$. Introduciamo infine la discretizzazione di $u(x, y)$ definita da

$$u_k \approx u(x_i, y_j), \quad k = (j-1)m_x + i$$

di cui si vede un esempio in Figura 12.1. La matrice di discretizzazio-

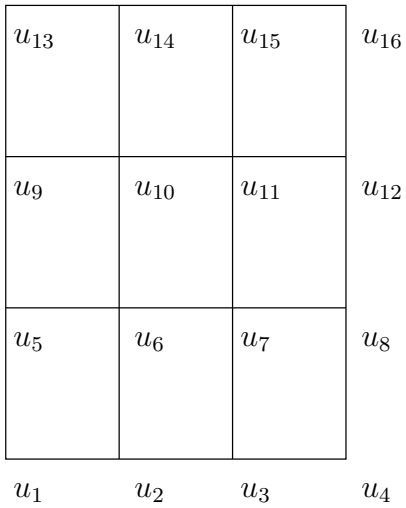


Figura 12.1: Numerazione di una griglia bidimensionale

ne alle differenze finite centrate del secondo ordine, *senza* tener conto delle condizioni al bordo, è data da

$$A = I_{m_y} \otimes A_x + A_y \otimes I_{m_x}$$

ove $\otimes$ indica il prodotto di Kronecker e

$$A_x = \frac{1}{h_x^2} \begin{bmatrix} 2 & -1 & 0 & \ldots & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ldots & 0 & -1 & 2 & -1 \\ 0 & \ldots & \ldots & 0 & -1 & 2 \end{bmatrix}, \quad A_y = \frac{1}{h_y^2} \begin{bmatrix} 2 & -1 & 0 & \ldots & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ldots & 0 & -1 & 2 & -1 \\ 0 & \ldots & \ldots & 0 & -1 & 2 \end{bmatrix}$$

ove $A_x \in \mathbb{R}^{m_x \times m_x}$ e $A_y \in \mathbb{R}^{m_y \times m_y}$. Poi, le righe di indice, diciamo k, corrispondente ad un nodo al bordo vanno sostituite con il vettore della base

canonica e_k, diviso per $h_x^2 + h_y^2$. Il termine noto è $[b_1, b_2, \ldots, b_{m_x m_y}]^{\mathrm{T}}$, ove

$$b_k = \begin{cases} f(x_i, y_j) & \text{se } (x_i, y_j) \text{ è un nodo interno, } k = (j-1)m_x + i \\ \frac{D_a(y_j)}{h_x^2+h_y^2} & \text{se } x_i = a,\ k = (j-1)m_x + i \\ \frac{D_b(y_j)}{h_x^2+h_y^2} & \text{se } x_i = b,\ k = (j-1)m_x + i \\ \frac{D_c(x_i)}{h_x^2+h_y^2} & \text{se } y_j = c,\ k = (j-1)m_x + i \\ \frac{D_d(x_i)}{h_x^2+h_y^2} & \text{se } y_j = d,\ k = (j-1)m_x + i \end{cases}$$

Alternativamente, si può sostituire il solo termine diagonale delle righe corrispondenti ad un nodo al bordo con un coefficiente $M/(h_x^2 + h_y^2)$, $M \gg 1$ e moltiplicare per M il corrispondente elemento nel termine noto. Questa procedura permette di assegnare, di fatto, le condizioni al bordo di Dirichlet, mantenendo la matrice A *simmetrica*.

In GNU Octave, la corretta numerazione dei nodi avviene con i comandi

```
> x = linspace(a,b,mx);
> y = linspace(c,d,my);
> [X,Y] = ndgrid(x,y);
```

e la costruzione della matrice A tramite il comando `kron`.

12.1.2 Condizioni al bordo miste

L'equazione di Poisson non può essere accompagnata solo da condizioni al bordo di Neumann, altrimenti la soluzione è indeterminata. Consideriamo allora il seguente problema con condizioni al bordo miste

$$\begin{cases} -\nabla^2 u(x,y) = f(x,y), \quad (x,y) \in [a,b] \times [c,d] \subset \mathbb{R}^2 \\ u(b,y) = D_b(y) \\ u(x,c) = D_c(x), \quad D_c(b) = D_b(c) \\ -\dfrac{\partial u}{\partial x}(x,y) = N_a(y), \quad x = a,\ c < y < d \\ \dfrac{\partial u}{\partial y}(x,y) = N_d(x), \quad y = d,\ x < b \end{cases}$$

La matrice di discretizzazione alle differenze finite centrate del secondo ordine è data da

$$A = I_{m_y} \otimes A_x + A_y \otimes I_{m_x}$$

ove

$$A_x = \frac{1}{h_x^2}\begin{bmatrix} 2 & -2 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -1 & 2 \end{bmatrix}, \quad A_y = \frac{1}{h_y^2}\begin{bmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & 0 & -2 & 2 \end{bmatrix}$$

Poi, le righe di indice, diciamo k, corrispondente ad un nodo al bordo su cui sono prescritte condizioni di Dirichlet vanno sostituite con il vettore della base canonica e_k, diviso per $h_x^2 + h_y^2$. La riga di indice m_y, corrispondente al nodo di bordo (a, c), va sostituita con

$$[0, \dots, 0, 1] \otimes \frac{1}{h_x^2}[-2, 5, -4, 1, 0, \dots, 0] + \frac{1}{h_y^2}[0, \dots, 0, -2, 2] \otimes [1, 0, \dots, 0]$$

(si può verificare che lo stencil $[2, -5, 4, -1]/h_x^2$ è un'approssimazione al secondo ordine della derivata seconda). Il termine noto è $[b_1, b_2, \dots, b_{m_x m_y}]^{\mathrm{T}}$, ove

$$b_k = \begin{cases} f(x_i, y_j) & \text{se } (x_i, y_j) \text{ è un nodo interno, } k = (j-1)m_x + i \\ \frac{D_b(y_j)}{h_x^2+h_y^2} & \text{se } x_i = b,\ k = (j-1)m_x + i \\ \frac{D_c(x_i)}{h_x^2+h_y^2} & \text{se } y_j = c,\ k = (j-1)m_x + i \\ f(x_i, y_j) + \frac{2N_a(y_i)}{h_x} & \text{se } x_i = a,\ k = (j-1)m_x + i,\ j \neq 1,\ j \neq m_y \\ f(x_i, y_j) + \frac{2N_d(x_i)}{h_y} & \text{se } y_j = d,\ k = (j-1)m_y + i,\ i \neq m_x \end{cases}$$

Capitolo 13

Metodi variazionali

13.1 Formulazione variazionale di un problema modello

Consideriamo il seguente problema ai limiti (equazione di Poisson)

$$\begin{cases} -u''(x) = f(x), & x \in (0,1) \\ u(0) = u(1) = 0 \end{cases} \tag{13.1}$$

con $f \in \mathcal{C}^0([0,1])$. Introduciamo il seguente spazio lineare:

$$V = \{v\colon\ v \in \mathcal{C}^0([0,1]),\ v' \text{ continua a tratti e limitata},\ v(0) = v(1) = 0\}$$

e il funzionale lineare $J\colon V \to \mathbb{R}$ dato da

$$J(v) = \frac{1}{2}(v', v') - (f, v)$$

ove

$$(v, w) = \int_0^1 v(x)w(x)\mathrm{d}x$$

Teorema 2 (Formulazione variazionale)**.** *Se u è la soluzione del problema (13.1), allora*

$$(u', v') = (f, v), \quad \forall v \in V \tag{13.2}$$

e, equivalentemente,

$$J(u) \leq J(v), \quad \forall v \in V \tag{13.3}$$

Dimostrazione. Sia u soluzione di (13.1). Allora, per ogni $v \in V$,

$$\int_0^1 -u''(x)v(x)\mathrm{d}x = \int_0^1 f(x)v(x)\mathrm{d}x = (f, v)$$

Integrando per parti,

$$\int_0^1 -u''(x)v(x)\mathrm{d}x = -u'(x)v(x)\Big|_0^1 + \int_0^1 u'(x)v'(x)\mathrm{d}x = (u', v')$$

poiché $v(0) = v(1) = 0$.

Sia adesso $u \in V$ soluzione di (13.2) e $w = v - u$, per $v \in V$. Allora $w \in V$ e

$$\begin{aligned} J(v) = J(u+w) &= \frac{1}{2}(u'+w', u'+w') - (f, u+w) = \\ &= \frac{1}{2}(u', u') + (u', w') + \frac{1}{2}(w', w') - (f, u) - (f, w) \geq J(u) \end{aligned}$$

perché $(u', w') - (f, w) = 0$ e $(w', w') \geq 0$. Dunque $J(u) \leq J(v)$.

Sia infine $u \in V$ soluzione di (13.3). Allora

$$J(u) \leq J(u + \varepsilon v), \quad \forall \varepsilon,\ \forall v \in V$$

Allora $\psi(\varepsilon) = J(u + \varepsilon v)$ ha un minimo in $\varepsilon = 0$ e dunque $\psi'(0) = 0$. Poiché

$$\psi(\varepsilon) = \frac{1}{2}(u', u') + \varepsilon(u', v') + \frac{\varepsilon^2}{2}(v', v') - (f, u) - \varepsilon(f, v)$$

si ha

$$\begin{aligned} 0 = \psi'(0) &= \lim_{\varepsilon \to 0} \frac{\psi(\varepsilon) - \psi(0)}{\varepsilon} = \lim_{\varepsilon \to 0} \left[(u', v') + \frac{\varepsilon}{2}(v', v') - (f, v) \right] = \\ &= (u', v') - (f, v) \end{aligned}$$

□

Abbiamo dunque dimostrato le seguenti implicazioni

$$(13.1) \Rightarrow (13.2) \Leftrightarrow (13.3)$$

Per quanto visto per il problema modello (10.1), la soluzione di (13.1) esiste ed è unica. Dunque, essa è soluzione anche di (13.2) e (13.3). Ci sono per caso altre soluzioni di (13.2)? No: se u_1 e u_2 sono due soluzioni, allora

$$(u_1' - u_2', v') = 0, \quad \forall v \in V$$

e in particolare per $v = u_1 - u_2$. Dunque

$$\int_0^1 (u_1'(x) - u_2'(x))^2 \mathrm{d}x = 0$$

e quindi $u_1'(x) - u_2'(x) = (u_1(x) - u_2(x))' = 0$. Pertanto $u_1 - u_2$ è costante e siccome $u_1(0) - u_2(0) = 0$, allora $u_1(x) - u_2(x) = 0$.

La soluzione di (13.1) si chiama *soluzione forte* del problema (13.1), mentre la soluzione di (13.2) (o, equivalentemente, di (13.3)) si chiama *soluzione debole* del problema (13.1). Perché soluzione debole? Se u è soluzione di (13.2) *e* $u \in \mathcal{C}^2([0,1])$, allora $0 = (u' - f, v) = (-u'' - f, v)$ per ogni $v \in V$. Poiché $u'' + f$ è continua, si deduce $-u''(x) = f(x)$ per $0 < x < 1$.

Le formulazioni variazionali (13.2) e (13.3) del problema (13.1) sono in realtà le più "fisiche": pensando al problema della trave, esse permettono di descrivere anche il caso in cui il carico $f(x)$ non sia continuo (ma, per esempio, applicato in un solo punto). Basta infatti che sia possibile calcolare (f, v), $v \in V$ e dunque basta, per esempio, che f sia continua a tratti. Quindi, in generale, è possibile come modello per un fenomeno fisico la sola formulazione debole.

13.1.1 Metodo di approssimazione variazionale

Prendiamo un sottospazio V_h di V di dimensione finita. Si cerca allora $u_h \in V_h$ tale che

$$(u_h, v_h)' = (f, v_h), \quad \forall v_h \in V_h \tag{13.4}$$

(metodo di Galerkin) ove, per brevità, $(u_h, v_h)' = (u_h', v_h')$, o, equivalentemente

$$J(u_h) = \inf_{v_h \in V_h} J(v_h)$$

(metodo di Ritz).

Teorema 3. *Il problema* (13.4) *ha un'unica soluzione.*

Dimostrazione. Sia $\{\phi_j\}_{j=1}^m$ una base di V_h. Allora

$$u_h(x) = \sum_{j=1}^m u_j \phi_j(x)$$

e il problema (13.4) si riscrive, per $i = 1, 2, \ldots, m$,

$$\int_0^1 u_h'(x)\phi_i'(x)\mathrm{d}x = \left(\sum_{j=1}^m u_j\phi_j, \phi_i\right)' = \sum_{j=1}^m (\phi_j, \phi_i)' u_j = Au_h = (f, \phi_i)$$

ove $A = (a_{ij}) = (\phi_j, \phi_i)'$. Calcoliamo ora $u_h^{\mathrm{T}} A u_h$. Si ha

$$u_h^{\mathrm{T}} A u_h = \sum_{i=1}^m u_i \left(\sum_{j=1}^m (\phi_i, \phi_j)' u_j\right)$$

da cui, per la linearità del prodotto scalare,

$$u_h^{\mathrm{T}} A u_h = \left(\sum_{i=1}^{m} u_i \phi_i(x), \sum_{j=1}^{m} u_j \phi_j(x) \right)' = \int_0^1 \left(\sum_{j=1}^{m} u_j \phi_j'(x) \right)^2 \mathrm{d}x \geq 0$$

e l'unica possibilità per avere 0 è che $u_h(x)$ sia nullo. Dunque, A è definita positiva. □

La matrice A, che risulta essere simmetrica e definita positiva, si chiama matrice di rigidezza *(stiffness matrix)* e il vettore (f, ϕ_i) vettore di carico *(load vector)*. Vale poi il seguente risultato:

Teorema 4. *Se u è soluzione di (13.2) e u_h di (13.4), allora*

$$\|u - u_h\|' \leq \inf_{v_h \in V_h} \|u - v_h\|'$$

ove $\|\cdot\|' = \sqrt{(\cdot,\cdot)'}$.

Dimostrazione. Dalle uguaglianze

$$\begin{aligned} (u, v)' &= (f, v) \quad \forall v \in V \text{ e, dunque, } \forall v \in V_h \\ (u_h, v_h)' &= (f, v_h) \quad \forall v_h \in V_h \end{aligned}$$

si ricava $((u - u_h), w_h)' = 0$ per ogni $w_h \in V_h$. Dunque

$$\begin{aligned} (u - u_h, u - u_h)' = (u - u_h, u - v_h + v_h - u_h)' = (u - u_h, u - v_h)' \leq \\ \leq \|u - u_h\|' \|u - v_h\|' \end{aligned}$$

(per la disuguaglianza di Cauchy–Schwartz) da cui

$$\|u - u_h\|' \leq \|u - v_h\|', \quad \forall v_h \in V_h$$

e quindi la tesi. □

Per definizione, u_h è allora la proiezione ortogonale della soluzione esatta u sul sottospazio V_h, tramite il prodotto scalare $(\cdot,\cdot)'$.

La scelta di V_h caratterizza il metodo. Da un lato bisogna considerare la regolarità della soluzione richiesta. Dall'altro la difficoltà di *assemblare* la matrice di rigidezza e di risolvere il sistema lineare.

Stabilità e consistenza

La consistenza del metodo di Galerkin discende da

$$(u, v_h)' = (f, v_h), \quad \forall v_h \in V_h$$

(il metodo si dice *fortemente* consistente).

Per quanto riguarda la stabilità, cominciamo ad osservare che se u_h soddisfa (13.4), allora

$$\left| \int_0^1 2x u_h(x) u_h'(x) \mathrm{d}x \right| \leq 2 \left| \int_0^1 u_h(x) u_h'(x) \mathrm{d}x \right| \leq 2\sqrt{(u_h, u_h)}\sqrt{(u_h', u_h')}$$

per la monotonia degli integrali ($x \leq 1$ in $[0,1]$) e la disuguaglianza di Cauchy–Schwartz e

$$\int_0^1 2x u_h(x) u_h'(x) \mathrm{d}x = \int_0^1 x u_h^2(x)' \mathrm{d}x = u_h^2(x) x \Big|_0^1 - \int_0^1 u_h^2(x) \mathrm{d}x$$

da cui

$$(u_h, u_h) \leq 2\sqrt{(u_h, u_h)}\sqrt{(u_h', u_h')} = 2\sqrt{(u_h, u_h)}\sqrt{(u_h, u_h)'}$$

cioè

$$\sqrt{(u_h, u_h)} \leq 2\|u_h\|'$$

Poiché u_h soddisfa, in particolare,

$$(u_h, u_h)' = (f, u_h)$$

si ricava, *suppondendo* f a quadrato sommabile,

$$\|u_h\|'^2 \leq \sqrt{(f,f)}\sqrt{(u_h, u_h)} \leq 2\sqrt{(f,f)}\|u_h\|'$$

da cui

$$\|u_h\|' \leq 2\sqrt{(f,f)}$$

Si conclude osservando che $\tilde{u}_h - \hat{u}_h$ soddisfa

$$(\tilde{u}_h - \hat{u}_h, v_h)' = (\tilde{f} - \hat{f}, v_h), \quad \forall v_h \in V_h$$

ove

$$(\tilde{u}_h, v_h)' = (\tilde{f}, v_h), \quad \forall v_h \in V_h$$
$$(\hat{u}_h, v_h)' = (\hat{f}, v_h), \quad \forall v_h \in V_h$$

e pertanto

$$\|\tilde{u}_h - \hat{u}_h\|' \leq 2\sqrt{(\tilde{f} - \hat{f}, \tilde{f} - \hat{f})}$$

e cioè che piccole variazioni sui dati producono piccole variazioni sulle soluzioni.

Metodo degli elementi finiti (FEM)

Introduciamo una discretizzazione dell'intervallo $[0,1]$ a passo *variabile*, come in Figura 13.1 Lo spazio V_h è generato dalle funzioni di base $\{\phi_j\}_{j=2}^{m-1}$, le quali

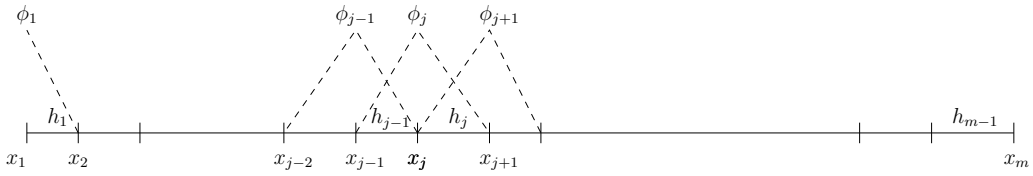


Figura 13.1: Hat functions

sono definite da

$$\phi_j(x) = \begin{cases} \dfrac{x - x_{j-1}}{h_{j-1}}, & x_{j-1} \le x \le x_j \\ \dfrac{x_{j+1} - x}{h_j}, & x_j \le x \le x_{j+1} \\ 0, & \text{altrimenti} \end{cases}$$

e

$$\phi_j'(x) = \begin{cases} \dfrac{1}{h_{j-1}}, & x_{j-1} < x < x_j \\ -\dfrac{1}{h_j}, & x_j < x < x_{j+1} \\ 0, & \text{altrimenti} \end{cases}$$

Tuttavia, per permettere la trattazione di problemi con differenti condizioni al bordo, consideriamo anche

$$\phi_1(x) = \begin{cases} \dfrac{x_2 - x}{h_1}, & x_1 \le x \le x_2 \\ 0, & \text{altrimenti} \end{cases}$$

e

$$\phi_1'(x) = \begin{cases} -\dfrac{1}{h_1}, & x_1 < x < x_2 \\ 0, & \text{altrimenti} \end{cases}$$

e

$$\phi_m(x) = \begin{cases} \dfrac{x - x_{m-1}}{h_{m-1}}, & x_{m-1} \le x \le x_m \\ 0, & \text{altrimenti} \end{cases}$$

e

$$\phi'_m(x) = \begin{cases} \dfrac{1}{h_{m-1}}, & x_{m-1} < x < x_m \\ 0, & \text{altrimenti} \end{cases}$$

Dunque, nell'approssimazione

$$u_h(x) = \sum_{j=1}^{m} u_j \phi_j(x)$$

i coefficienti u_j sono i valori di u_h nei nodi x_j. Il problema (13.4) si riscrive

$$\int_0^1 u'_h(x)\phi'_i(x)\mathrm{d}x = \sum_{j=1}^{m} u_j \int_0^1 \phi'_j(x)\phi'_i(x)\mathrm{d}x = \sum_{j=1}^{m} u_j \int_{x_i-h_{i-1}}^{x_i+h_i} \phi'_j(x)\phi'_i(x)\mathrm{d}x =$$
$$= \sum_{j=1}^{m} u_j a_{ij} = \int_{x_i-h_{i-1}}^{x_i+h_i} f(x)\phi_i(x)\mathrm{d}x$$

Siccome il supporto di $\phi_j(x)$ è $[x_{j-1}, x_{j+1}]$, gli unici elementi non nulli a_{ij} sono a_{ii}, $a_{i\,i-1}$ e $a_{i+1\,i} = a_{i\,i+1}$. Per $1 < i < m$,

$$a_{ii} = (\phi_i, \phi_i)' = \int_{x_i-h_{i-1}}^{x_i} \left(\frac{1}{h_{i-1}}\right)^2 \mathrm{d}x + \int_{x_i}^{x_i+h_i} \left(-\frac{1}{h_i}\right)^2 \mathrm{d}x = \frac{1}{h_{i-1}} + \frac{1}{h_i}$$
$$a_{i\,i-1} = (\phi_{i-1}, \phi_i)' = \int_{x_i-h_{i-1}}^{x_i} -\frac{1}{h_{i-1}} \cdot \frac{1}{h_{i-1}}\mathrm{d}x = -\frac{1}{h_{i-1}} = a_{i-1\,i}$$

Per $i = 1$ e $i = m$, si ha invece

$$a_{11} = \int_{x_1}^{x_1+h_1} \left(-\frac{1}{h_1}\right)^2 \mathrm{d}x = \frac{1}{h_1}$$
$$a_{21} = \int_{x_2-h_1}^{x_2} -\frac{1}{h_1} \cdot \frac{1}{h_1}\mathrm{d}x = -\frac{1}{h_1} = a_{12}$$
$$a_{m\,m-1} = \int_{x_m-h_{m-1}}^{x_m} -\frac{1}{h_{m-1}} \cdot \frac{1}{h_{m-1}}\mathrm{d}x = -\frac{1}{h_{m-1}} = a_{m-1\,m}$$
$$a_{mm} = \int_{x_m-h_{m-1}}^{x_m} \left(-\frac{1}{h_{m-1}}\right)^2 \mathrm{d}x = \frac{1}{h_{m-1}}$$

Per quanto riguarda il calcolo di (f, ϕ_i) si può ricorrere alla formula di quadratura del trapezio che risulta essere sufficientemente accurata. Si ha dunque, per $1 < i < m$,

$$f_i = (f, \phi_i) = \int_{x_i-h_{i-1}}^{x_i} f(x)\frac{x - x_{i-1}}{h_{i-1}}\mathrm{d}x + \int_{x_i}^{x_i+h_i} f(x)\frac{x_{i+1} - x}{h_i}\mathrm{d}x \approx$$
$$\approx f(x_i)\frac{h_{i-1}}{2} + f(x_i)\frac{h_i}{2} = f(x_i)\frac{h_{i-1} + h_i}{2}$$

Per $i = 1$ e $i = m$ si ha invece

$$f_1 = (f, \phi_1) = \int_{x_1}^{x_1+h_1} f(x)\frac{x_2 - x}{h_1}\mathrm{d}x \approx f(x_1)\frac{h_1}{2}$$
$$f_m = (f, \phi_m) = \int_{x_m-h_{m-1}}^{x_m} f(x)\frac{x - x_{m-1}}{h_{m-1}}\mathrm{d}x \approx f(x_m)\frac{h_{m-1}}{2}$$

Più precisa (anche se dello stesso costo) risulta essere la formula del punto medio: per $1 < i < m$ è

$$f_i = (f, \phi_i) = \int_{x_{i-1}}^{x_i} f(x)\frac{x - x_{i-1}}{h_{i-1}}\mathrm{d}x + \int_{x_i}^{x_{i+1}} f(x)\frac{x_{i+1} - x}{h_i}\mathrm{d}x \approx$$
$$\approx f\left(\frac{x_{i-1} + x_i}{2}\right)\frac{h_{i-1}}{2} + f\left(\frac{x_i + x_{i+1}}{2}\right)\frac{h_i}{2}$$

Per $i = 1$ e $i = m$ si ha invece

$$f_1 = (f, \phi_1) = \int_{x_1}^{x_2} f(x)\frac{x_2 - x}{h_1}\mathrm{d}x \approx f\left(\frac{x_1 + x_2}{2}\right)\frac{h_1}{2}$$
$$f_m = (f, \phi_m) = \int_{x_{m-1}}^{x_m} f(x)\frac{x - x_{m-1}}{h_{m-1}}\mathrm{d}x \approx f\left(\frac{x_{m-1} + x_m}{2}\right)\frac{h_{m-1}}{2}$$

Siccome

$$f\left(\frac{x_{i-1} + x_i}{2}\right) = \frac{f(x_{i-1}) + f(x_i)}{2} + \mathcal{O}(h_{i-1}^2)$$

e

$$\int_{x_{i-1}}^{x_i} \phi_i(x)\mathrm{d}x = \phi_i\left(\frac{x_{i-1} + x_i}{2}\right)h_{i-1} = \frac{h_{i-1}}{2}$$

la formula del punto medio viene di solito sostituita da

$$f_i = (f, \phi_i) \approx \frac{f(x_{i-1}) + f(x_i)}{2}\int_{x_i-h_{i-1}}^{x_i} \phi_i(x)\mathrm{d}x + \frac{f(x_i) + f(x_{i+1})}{2}\int_{x_i}^{x_i+h_i} \phi_i(x)\mathrm{d}x$$

per $1 < i < m$ e da

$$f_1 = (f, \phi_1) = \frac{f(x_1) + f(x_2)}{2}\int_{x_1}^{x_1+h_1} \phi_1(x)\mathrm{d}x$$
$$f_m = (f, \phi_m) = \frac{f(x_{m-1}) + f(x_m)}{2}\int_{x_m-h_{m-1}}^{x_m} \phi_m(x)\mathrm{d}x$$

La riga i-esima del sistema lineare risulta dunque essere (nel caso di quadratura trapezoidale)

$$\begin{bmatrix} 0 & \dots & 0 & -\frac{1}{h_{i-1}} & \left(\frac{1}{h_{i-1}}+\frac{1}{h_i}\right) & -\frac{1}{h_i} & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \vdots \\ u_{j-1} \\ u_j \\ u_{j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ f(x_i)\frac{h_{i-1}+h_i}{2} \\ \vdots \end{bmatrix}$$

e dunque *uguale* (per questo semplice problema modello) a quella della discretizzazione con differenze finite del secondo ordine. Pertanto, è naturale aspettarsi, sotto opportune ipotesi di regolarità, che l'errore rispetto alla soluzione analitica tenda a zero come h^2, $h = \max_j h_j$.

A questo punto si risolve il sistema lineare, dopo aver opportunamente modificato la matrice e il termine noto per imporre le condizioni al bordo di Dirichlet. Nel caso di condizioni di Neumann omogenee, non è necessaria alcuna modifica. Infatti, la forma debole del problema è

$$-u'(x)\phi_i(x)\Big|_0^1 + \int_0^1 u'(x)\phi_i'(x)\mathrm{d}x = \int_0^1 f(x)\phi_i(x)\mathrm{d}x, \quad i = 1, \dots, m$$

Il primo termine è naturalmente zero per $1 < i < m$. Non considerarlo, cioè porlo a zero, neanche per $i = 1$ ($i = m$) significa richiedere $u'(x_1) = 0$ ($u'(x_m) = 0$) visto che $\phi_1(x_1) = 1$ ($\phi_m(x_m) = 1$). Nel caso di condizioni di Neumann non omogenee (per esempio in 0), basta modificare la prima riga del sistema secondo l'equazione

$$\int_0^1 u'(x)\phi_1'(x)\mathrm{d}x = -u'(0) + \int_0^1 f(x)\phi_1(x)\mathrm{d}x$$

13.1.2 Estensione al caso bidimensionale

Tutto quanto detto si estende, in particolare, al caso bidimensionale. Si deve usare la formula di Green

$$\int_\Omega \nabla^2 u(\boldsymbol{x})v(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = -\int_\Omega \nabla u(\boldsymbol{x})\cdot\nabla v(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} v(s)\nabla u(s)\cdot\nu(s)\mathrm{d}s$$

ove $\nu(s)$ è il versore esterno normale a $\partial\Omega$.

13.2 Metodi spettrali

Sia

$$u(x) = \sum_j u_j\phi_j(x)$$

L'*indice algebrico* di convergenza è il più grande k tale che

$$\lim_{j\to\infty} |u_j| j^k < +\infty$$

Se tale limite è finito per ogni k, allora la serie si dice convergere *esponenzialmente* oppure *spettralmente*. Significa che $|u_j|$ decade più velocemente di ogni potenza negativa di j. Parleremo di *metodi spettrali* quando useremo un'approssimazione di una serie convergente spettralmente

$$u(x) \approx \sum_{j=1}^{m} \hat{u}_j \phi_j(x)$$

per $u(x)$.

Consideriamo un sistema $\{\phi_j\}_j$ ortonormale rispetto al prodotto scalare

$$\int_a^b \phi_j(x)\phi_i(x)w(x)\mathrm{d}x = \delta_{ji}$$

La formulazione di Galerkin di un problema ai limiti $Lu = f$, L operatore differenziale *lineare*, diventa

$$\sum_{j=1}^{m} \hat{u}_j \int_a^b L\phi_j(x)\phi_i(x)w(x)\mathrm{d}x = \int_a^b f(x)\phi_i(x)w(x)\mathrm{d}x, \quad 1 \le i \le m$$

Nel caso non si possano calcolare analiticamente o con formule di quadratura esatte gli integrali, si ricorre alle formule di quadratura gaussiana a m punti, dando origine al sistema lineare

$$\sum_{j=1}^{m} \hat{u}_j \left(\sum_{n=1}^{m} L\phi_j(x_n)\phi_i(x_n)w_n \right) = \sum_{n=1}^{m} f(x_n)\phi_i(x_n)w_n, \quad 1 \le i \le m \quad (13.5)$$

In tal caso si parla di metodi *pseudospettrali*. I coefficienti $\hat{u}_j$ che si trovano risolvendo il sistema lineare si chiamano solitamente soluzione *nello spazio spettrale*. Dati i coefficienti, si ricostruisce la soluzione *nello spazio fisico* $\sum_j \hat{u}_j \phi_j(x)$.

Solitamente le funzioni $\{\phi_j(x)\}_j$ sono polinomi ortonormali rispetto alla funzione peso $w(x)$. La soluzione $u(x)$, però, potrebbe non essere efficacemente approssimata da polinomi, per esempio se deve soddisfare particolari condizioni al contorno (tipo *vanishing boundary conditions, condizioni al bordo periodiche* o altro). Può essere utile allora la decomposizione

$$u(x) \approx \sum_{j=1}^{m} \hat{u}_j \phi_j(x) \sqrt{w(x)}$$

La formulazione di Galerkin di $Lu = f$ diventa allora

$$\sum_{j=1}^{m} \hat{u}_j \int_a^b L(\phi_j(x)\sqrt{w(x)})\phi_i(x)\sqrt{w(x)}\mathrm{d}x = \int_a^b f(x)\phi_i(x)\sqrt{w(x)}\mathrm{d}x,\ 1 \le i \le m$$

Consideriamo ora un caso particolare di fondamentale importanza. Molte proprietà risultato comuni anche agli altri metodi presudospettrali.

13.2.1 Trasformata di Fourier

Sia $[a, b]$ un intervallo di $\mathbb{R}$, $m > 0$ pari e fissato. Consideriamo, per ogni $j \in \mathbb{Z}$,

$$\phi_j(x) = \frac{\mathrm{e}^{\mathrm{i}(j-1-m/2)2\pi(x-a)/(b-a)}}{\sqrt{b-a}}$$

Allora,

$$\int_a^b \phi_j(x)\overline{\phi_k(x)}\mathrm{d}x = \delta_{jk} \tag{13.6}$$

Infatti, se $j = k$ allora $\phi_j(x)\overline{\phi_k(x)} = 1/(b-a)$, altrimenti

$$\phi_j(x)\overline{\phi_k(x)} = \frac{\mathrm{e}^{\mathrm{i}2\pi(j-k)(x-a)/(b-a)}}{b-a}$$

e quindi

$$\int_a^b \phi_j(x)\overline{\phi_k(x)}\mathrm{d}x = \int_0^1 \frac{\mathrm{e}^{\mathrm{i}2\pi(j-k)y}}{b-a}(b-a)\mathrm{d}y = 0\ ,$$

poiché l'integrale delle funzioni sin e cos in un intervallo multiplo del loro periodo è nullo. La famiglia di funzioni $\{\phi_j\}_j$ si dice *ortonormale* nell'intervallo $[a, b]$ rispetto al prodotto scalare

$$(\phi_j, \phi_k) = \int_a^b \phi_j(x)\overline{\phi_k(x)}\mathrm{d}x$$

Un risultato utile è il seguente

$$\sum_{n=1}^{m} \mathrm{e}^{\mathrm{i}(n-1)2\pi(j-k)/m} = m\delta_{jk}, \quad 1 \le j, k \le m \tag{13.7}$$

È ovvio per $j = k$; altrimenti

$$\sum_{n=1}^{m} \mathrm{e}^{\mathrm{i}(n-1)2\pi(j-k)/m} = \sum_{n=0}^{m-1} \left(\mathrm{e}^{\mathrm{i}2\pi(j-k)/m}\right)^n =$$

$$= \frac{1-\mathrm{e}^{\mathrm{i}2\pi(j-k)}}{1-\mathrm{e}^{\mathrm{i}2\pi(j-k)/m}} = \frac{1-\cos(2\pi(j-k))}{1-\mathrm{e}^{\mathrm{i}2\pi(j-k)/m}} = 0$$

poiché $-m + 1 \le j - k \le m - 1$.

13.2.2 Trasformata di Fourier discreta

Sia u una funzione da $[a,b]$ a $\mathbb{C}$ *periodica* ($u(a) = u(b)$). Supponiamo che u si possa scrivere (ciò è vero, per esempio, per funzioni di classe C^1) come

$$u(x) = \sum_{j=-\infty}^{+\infty} u_j \phi_j(x), \quad u_j \in \mathbb{C} \tag{13.8}$$

Fissato $k \in \mathbb{Z}$, moltiplicando entrambi i membri per $\overline{\phi_k(x)}$ e integrando nell'intervallo $[a,b]$, usando (13.6) si ottiene

$$\begin{aligned} \int_a^b u(x)\overline{\phi_k(x)}\mathrm{d}x &= \int_a^b \left(\sum_{j=-\infty}^{+\infty} u_j \phi_j(x)\overline{\phi_k(x)} \right) \mathrm{d}x = \\ &= \sum_{j=-\infty}^{\infty} u_j \int_a^b \phi_j(x)\overline{\phi_k(x)}\mathrm{d}x = u_k \end{aligned} \tag{13.9}$$

Dunque, abbiamo un'espressione esplicita per u_j. Analogamente si vede che

$$\int_a^b |u(x)|^2 \mathrm{d}x = \sum_{j=-\infty}^{+\infty} |u_j|^2 \quad \text{(identità di Parseval)}$$

La prima approssimazione da fare consiste nel troncare la serie infinita. Osserviamo che, definito $J = \mathbb{Z} \setminus \{1, 2, \ldots, m\}$,

$$\begin{aligned} \int_a^b \left| u(x) - \sum_{j=1}^{m} u_j \phi_j(x) \right|^2 \mathrm{d}x &= \int_a^b \left| \sum_{j \in J} u_j \phi_j(x) \right|^2 \mathrm{d}x = \\ &= \int_a^b \left(\sum_{j \in J} u_j \phi_j(x) \right) \left(\sum_{k \in J} \overline{u_k}\overline{\phi_k(x)} \right) \mathrm{d}x = \\ &= \sum_{k \in J} |u_k|^2 \end{aligned}$$

Stimiamo adesso u_k: si ha, per funzioni di classe $\mathcal{C}^1$, integrando per parti

$$\begin{aligned} u_k = \int_a^b u(x)\overline{\phi_k(x)}\mathrm{d}x &= -\frac{b-a}{\mathrm{i}(k-1-m/2)2\pi} \left(\frac{u(b)}{\sqrt{b-a}} - \frac{u(a)}{\sqrt{b-a}} \right) + \\ &+ \frac{b-a}{\mathrm{i}(k-1-m/2)2\pi} \int_a^b u'(x)\overline{\phi_k(x)}\mathrm{d}x = \\ &= \mathcal{O}\left((k-1-m/2)^{-1} \right) \end{aligned}$$

Se anche $u'(a) = u'(b)$ e $u' \in \mathcal{C}^1$, allora, integrando ancora per parti, si ottiene $u_k = \mathcal{O}\left((k-1-m/2)^{-2}\right)$ e così via. Se dunque u è infinitamente derivabile e periodica (cioè tutte le derivate sono periodiche), allora u_k decade più velocemente di ogni potenza negativa di k.

La seconda approssimazione da fare è utilizzare una formula di quadratura per il calcolo di u_k. Riportiamo per comodità la formula di quadratura trapezoidale a $m+1$ nodi equispaziati $x_n = (b-a)y_n + a$, ove $y_n = (n-1)/m$, $n = 1, \dots, m+1$ per funzioni periodiche $f(a) = f(b)$:

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2m}\left(f(x_1) + 2\sum_{n=2}^{m} f(x_n) + f(x_{m+1})\right) = \frac{b-a}{m}\sum_{n=1}^{m} f(x_n)$$

Usando la (13.7), abbiamo

$$\begin{aligned} m\delta_{jk} &= \sum_{n=1}^{m} \mathrm{e}^{\mathrm{i}(n-1)2\pi(j-k)/m} = \sum_{n=1}^{m} \mathrm{e}^{\mathrm{i}(j-k)2\pi y_n} = \sum_{n=1}^{m} \mathrm{e}^{\mathrm{i}(j-k)2\pi(x_n-a)/(b-a)} = \\ &= (b-a)\sum_{n=1}^{m} \frac{\mathrm{e}^{\mathrm{i}(j-1-m/2)2\pi(x_n-a)/(b-a)}}{\sqrt{b-a}} \frac{\mathrm{e}^{-\mathrm{i}(k-1-m/2)2\pi(x_n-a)/(b-a)}}{\sqrt{b-a}} = \\ &= (b-a)\sum_{n=1}^{m} \phi_j(x_n)\overline{\phi_k(x_n)} = m\int_a^b \phi_j(x)\overline{\phi_k(x)}\mathrm{d}x \end{aligned}$$

cioè la famiglia $\{\phi_j\}_j$, $1 \le j \le m$, è ortonormale anche rispetto al prodotto scalare *discreto*

$$(\phi_j, \phi_k)_{\mathrm{d}} = \frac{b-a}{m}\sum_{n=1}^{m} \phi_j(x_n)\overline{\phi_k(x_n)}$$

Questo significa che la formula di quadratura trapezoidale a m punti è esatta per le funzioni $\{\phi_j\}_{j=-m+1}^{m-1}$. Applicando la formula di quadratura ai coefficienti (13.9) si ottiene

$$\begin{aligned} u_k &= \int_a^b u(x)\frac{\mathrm{e}^{-\mathrm{i}(k-1-m/2)2\pi(x-a)/(b-a)}}{\sqrt{b-a}}\mathrm{d}x = \\ &= \sqrt{b-a}\int_0^1 u((b-a)y+a)\mathrm{e}^{-\mathrm{i}(k-1)2\pi y}\mathrm{e}^{\mathrm{i}m\pi y}\mathrm{d}y \approx \\ &\approx \frac{\sqrt{b-a}}{m}\boxed{\sum_{n=1}^{m}\left(u(x_n)\mathrm{e}^{\mathrm{i}m\pi y_n}\right)\mathrm{e}^{-\mathrm{i}(k-1)2\pi y_n}} = \hat{u}_k \end{aligned}$$

ove $x = (b-a)y + a$.

La funzione *(serie troncata di Fourier)*

$$\hat{u}(x) = \sum_{j=1}^{m} \hat{u}_j \phi_j(x) = \sum_{k=-m/2}^{m/2-1} \hat{u}_{k+1+m/2} \phi_{k+1+m/2}(x) =$$
$$= \sum_{k=-m/2}^{m/2-1} \hat{u}_{k+1+m/2} \frac{\mathrm{e}^{\mathrm{i}k2\pi(x-a)/(b-a)}}{\sqrt{b-a}}$$

è un polinomio trigonometrico che approssima $u(x)$ ed è *interpolante* nei nodi x_n. Infatti, usando (13.7),

$$\begin{aligned}
\hat{u}(x_n) &= \sum_{j=1}^{m} \hat{u}_j \phi_j(x_n) = \\
&= \sum_{j=1}^{m} \left(\frac{\sqrt{b-a}}{m} \sum_{k=1}^{m} \left(u(x_k) \mathrm{e}^{\mathrm{i}m\pi y_k} \right) \mathrm{e}^{-\mathrm{i}(j-1)2\pi y_k} \right) \frac{\mathrm{e}^{\mathrm{i}(j-1-m/2)2\pi(x_n-a)/(b-a)}}{\sqrt{b-a}} = \\
&= \frac{1}{m} \sum_{k=1}^{m} u(x_k) \mathrm{e}^{\mathrm{i}m\pi(k-1)/m} \mathrm{e}^{-\mathrm{i}m\pi(n-1)/m} \sum_{j=1}^{m} \mathrm{e}^{-\mathrm{i}(j-1)2\pi(k-1)/m} \mathrm{e}^{\mathrm{i}(j-1)2\pi(n-1)/m} = \\
&= \frac{1}{m} \sum_{k=1}^{m} u(x_k) \mathrm{e}^{\mathrm{i}(k-n)\pi} \sum_{j=1}^{m} \mathrm{e}^{\mathrm{i}(j-1)2\pi(n-k)/m} = \frac{1}{m} u(x_n) m = u(x_n) \, .
\end{aligned}$$

Si può far vedere poi che

$$\int_a^b \left| u(x) - \sum_{j=1}^{m} \hat{u}_j \phi_j(x) \right|^2 \mathrm{d}x \leq 2 \sum_{k \in J} |u_k|^2$$

La trasformazione

$$[u(x_1), u(x_2), \ldots, u(x_m)]^{\mathrm{T}} \to [\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_m]^{\mathrm{T}}$$

si chiama *trasformata di Fourier discreta* di u e $\hat{u}_1, \ldots, \hat{u}_m$ *coefficienti di Fourier* di u. Il vettore $m \cdot [\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_m]^{\mathrm{T}} / \sqrt{b-a}$ può essere scritto come prodotto matrice-vettore $F[u(x_1)\mathrm{e}^{\mathrm{i}m\pi y_1}, u(x_2)\mathrm{e}^{\mathrm{i}m\pi y_2}, \ldots, u(x_m)\mathrm{e}^{\mathrm{i}m\pi y_m}]^{\mathrm{T}}$, ove

$$F = (f_{jk}), \quad f_{jk} = \mathrm{e}^{-\mathrm{i}(j-1)2\pi y_k} \, .$$

Alternativamente, si può usare la Fast Fourier Transform (FFT). Il comando `fft` applicato al vettore $[u(x_1)\mathrm{e}^{\mathrm{i}m\pi y_1}, u(x_2)\mathrm{e}^{\mathrm{i}m\pi y_2}, \ldots, u(x_m)\mathrm{e}^{\mathrm{i}m\pi y_m}]^{\mathrm{T}}$ produce il vettore $m \cdot [\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_m]^{\mathrm{T}} / \sqrt{b-a}$, così come il comando `fftshift` applicato al risultato del comando `fft` applicato a $[u(x_1), u(x_2), \ldots, u(x_m)]$.

Dati dei coefficienti $\hat{v}_j$, $j = 1, \ldots, m$, si può considerare la funzione (periodica)

$$\sum_{j=1}^{m} \hat{v}_j \phi_j(x)$$

La valutazione nei nodi x_n, $1 \le n \le m$, porge

$$\hat{\hat{v}}_n = \sum_{j=1}^{m} \hat{v}_j \phi_j(x_n) = \sum_{j=1}^{m} \hat{v}_j \frac{\mathrm{e}^{\mathrm{i}(j-1-m/2)2\pi(x_n-a)/(b-a)}}{\sqrt{b-a}} =$$
$$= \frac{m}{\sqrt{b-a}} \boxed{\frac{1}{m}\left(\sum_{j=1}^{m} \hat{v}_j \mathrm{e}^{\mathrm{i}(j-1)2\pi y_n}\right)} \mathrm{e}^{-\mathrm{i}m\pi y_n} .$$

La trasformazione

$$[\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m]^{\mathrm{T}} \to [\hat{\hat{v}}_1, \hat{\hat{v}}_2, \ldots, \hat{\hat{v}}_m]^{\mathrm{T}}$$

si chiama *anti-trasformata di Fourier discreta*. Se i $\hat{v}_j$ sono i coefficienti di Fourier di una funzione $v(x)$, la proprietà di interpolazione comporta $\hat{\hat{v}}_n = v(x_n)$. Ma, in generale, non è vero che

$$v(x) = \sum_{j=1}^{m} \hat{v}_j \phi_j(x)$$

Il vettore $\sqrt{b-a} \cdot [\hat{\hat{v}}_1 \mathrm{e}^{\mathrm{i}m\pi y_1}, \hat{\hat{v}}_2 \mathrm{e}^{\mathrm{i}m\pi y_2}, \ldots, \hat{\hat{v}}_m \mathrm{e}^{\mathrm{i}m\pi y_m}]^{\mathrm{T}}/m$ può essere scritto come prodotto matrice-vettore $F'[\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m]^{\mathrm{T}}/m$, ove F' denota, come in GNU Octave, la trasposta coniugata di F. Alternativamente, il comando `ifft` applicato al vettore $[\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m]$ produce il vettore $\sqrt{b-a} \cdot [\hat{\hat{v}}_1 \mathrm{e}^{\mathrm{i}m\pi y_1}, \hat{\hat{v}}_2 \mathrm{e}^{\mathrm{i}m\pi y_2}, \ldots, \hat{\hat{v}}_m \mathrm{e}^{\mathrm{i}m\pi y_m}]/m$, mentre, se applicato al risultato del comando `ifftshift` applicato al vettore $[\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m]$, produce il vettore $\sqrt{b-a} \cdot [\hat{\hat{v}}_1, \hat{\hat{v}}_2, \ldots, \hat{\hat{v}}_m]/m$.

Applicazione ad un problema modello

Consideriamo la soluzione del problema

$$\begin{cases} -u''(x) + u(x) = \dfrac{1}{\sin x + 2}, & x \in (-\pi, \pi) \\ u(-\pi) = u(\pi) \end{cases}$$

mediante decomposizione in funzioni di Fourier. Posto $a = -\pi$, $b = \pi$, $f(x) = 1/(\sin x + 2)$, si ha

$$\phi_j(x) = \frac{\mathrm{e}^{\mathrm{i}(j-1-m/2)2\pi(x-a)/(b-a)}}{\sqrt{b-a}}$$

ove m è pari e fissato e, per $1 \leq k \leq m$,

$$\int_{-\pi}^{\pi} \left(-\sum_{j=1}^{m} u_j \phi_j(x) \right)'' \overline{\phi_k(x)} \mathrm{d}x + \int_{-\pi}^{\pi} \left(\sum_{j=1}^{m} u_j \phi_j(x) \right) \overline{\phi_k(x)} \mathrm{d}x = \\ = \int_{-\pi}^{\pi} f(x) \overline{\phi_k(x)} \mathrm{d}x$$

da cui

$$-\sum_{j=1}^{m} u_j \int_{-\pi}^{\pi} \phi_j''(x) \overline{\phi_k(x)} \mathrm{d}x + \sum_{j=1}^{m} u_j \int_{-\pi}^{\pi} \phi_j(x) \overline{\phi_k(x)} \mathrm{d}x = \int_{-\pi}^{\pi} f(x) \overline{\phi_k(x)} \mathrm{d}x$$

Poiché

$$\phi_j''(x) = \left(\frac{\mathrm{i}(j - 1 - m/2)2\pi}{b - a} \right)^2 \phi_j(x) = \lambda_j^2 \phi_j(x)$$

usando l'ortonormalità delle funzioni di Fourier e calcolando i coefficienti di Fourier di $f(x)$, si ha

$$-\lambda_k^2 \hat{u}_k + \hat{u}_k = \hat{f}_k, \quad 1 \leq k \leq m$$

da cui

$$\hat{u}_k = \frac{\hat{f}_k}{1 - \lambda_k^2}, \quad 1 \leq k \leq m$$

e quindi

$$u(x) \approx \sum_{j=1}^{m} \hat{u}_j \phi_j(x)$$

Da notare che le condizioni al bordo devono essere di tipo periodico: condizioni come

$$\begin{cases} u''(x) = f(x, u(x), u'(x)), & x \in (a, b) \\ u(a) = 0 \\ u(b) = 0 \end{cases}$$

sono invece di Dirichlet omogenee. Inoltre, la soluzione del problema *deve* poter essere periodica: per esempio, non possono esserci termini non omogenei *non periodici*.

Costi computazionali e stabilità

La Fast Fourier Transform di un vettore di lunghezza m ha costo $\mathcal{O}(m \log m)$, mentre il prodotto matrice-vettore $\mathcal{O}(m^2)$. Tali costi sono però asintotici e

nascondono i fattori costanti. Inoltre, GNU Octave può far uso di implementazioni ottimizzate di algebra lineare (come, ad esempio, le librerie ATLAS). In pratica, dunque, esiste un m_0 sotto il quale conviene, dal punto di vista del costo computazionale, usare il prodotto matrice-vettore e sopra il quale la FFT.

Per quanto riguarda l'accuratezza, in generale la FFT è più precisa del prodotto matrice vettore. Poiché la trasformata di Fourier discreta comporta l'uso di aritmetica complessa (anche se la funzione da trasformare è reale), la sequenza trasformata/anti-trasformata potrebbe introdurre una quantità immaginaria spuria che può essere eliminata con il comando `real`.

Anche per la trasformata di Fourier vi possono essere problemi di stabilità simili al fenomeno di Runge (qui chiamato *fenomeno di Gibbs*). Una tecnica per "smussare" (in inglese "to smooth") eventuali oscillazioni, consiste nel moltiplicare opportunamente i coefficienti di Fourier $\hat{u}_j$ per opportuni termini σ_j che decadono in j, per esempio

$$\sigma_j = \frac{\frac{m}{2} + 1 - \left|\frac{m}{2} + 1 - j\right|}{\frac{m}{2} + 1}, \quad 1 \leq j \leq m$$

Il risultato è che il coefficiente $\hat{u}_{m/2+1}$ è pesato da $\sigma_{m/2+1} = 1$, i coefficienti $\hat{u}_{m/2}$ e $u_{m/2+2}$ sono pesati da $m/(m+2)$ e così via fino al coefficiente $\hat{u}_1$ pesato da $2/(m+2)$. Questa scelta corrisponde alle *medie di Cesáro*. Infatti, si sostituisce la serie troncata di Fourier

$$\sum_{j=1}^{m} \hat{u}_j \phi_j(x)$$

con la media delle troncate

$$\frac{\displaystyle\sum_{k=0}^{\frac{m}{2}} \sum_{j=\frac{m}{2}+1-k}^{\max\{\frac{m}{2}+1+k,m\}} \hat{u}_j \phi_j(x)}{\frac{m}{2} + 1}$$

Si ricorda che se una serie è convergente, allora il limite delle medie delle sue troncate è la somma della serie.

Valutazione di un polinomio trigonometrico

Supponiamo di conoscere i coefficienti $\hat{u}_j$, $j = 1, \ldots, m$ e di voler valutare la funzione

$$u(x) = \sum_{j=1}^{m} \hat{u}_j \phi_j(x)$$

su un insieme di nodi target x_k equispaziati, $x_k = (k-1)/n$, $1 \le k \le n$, $n > m$, n pari. Si possono introdurre dei coefficienti fittizi $\hat{U}_k$

$$\begin{aligned} &\hat{U}_k = 0 && 1 \le k \le \frac{n-m}{2} \\ &\hat{U}_k = \hat{u}_{k-\frac{n-m}{2}} && \frac{n-m}{2} + 1 \le k \le m - \frac{n-m}{2} \\ &\hat{U}_k = 0 && m - \frac{n-m}{2} + 1 \le k \le n \end{aligned}$$

Si avrà

$$\hat{\hat{u}}_k = \sum_{j=1}^{m} \hat{u}_j \phi_j(x_k) = \sum_{j=1}^{n} \hat{U}_j \frac{\mathrm{e}^{\mathrm{i}(j-1-n/2)2\pi(x_k-a)/(b-a)}}{\sqrt{b-a}} =$$

$$= \frac{n}{\sqrt{b-a}} \boxed{\frac{1}{n} \left(\sum_{j=1}^{n} \hat{U}_j \mathrm{e}^{\mathrm{i}(j-1)2\pi y_k} \right)} \mathrm{e}^{-\mathrm{i}n\pi y_k}$$

Oppure si può costruire la matrice F relativa ai nodi (ciò funziona anche per nodi non equispaziati). Infine, si può usare le NFFT.

13.3 Metodi di collocazione

Si assume comunque

$$u(x) = \sum_{j=1}^{m} u_j \phi_j(x)$$

ove $\{\phi_j\}$ è un sistema ortonormale rispetto ad un prodotto scalare, ma si impone poi che l'equazione differenziale $Lu = f$ sia soddisfatta in certi nodi x_n. Si ha il seguente risultato interessante:

Teorema 5. *La soluzione del sistema lineare*

$$\sum_{j=1}^{m} u_j L\phi_j(x_n) = f(x_n), \quad n = 1, 2, \ldots, m \tag{13.10}$$

ove gli $\{x_n\}$ sono i nodi della quadratura gaussiana relativa alla famiglia $\{\phi_j\}$ è la stessa del problema di Galerkin

$$\sum_{j=1}^{m} u_j \int_a^b L\phi_j(x)\phi_i(x)w(x)\mathrm{d}x = \int_a^b f(x)\phi_i(x)w(x)\mathrm{d}x$$

quando si approssimino gli integrali con le formule gaussiane.

Dimostrazione. Per ogni i, $1 \leq i \leq m$, da (13.10), si ha

$$\sum_{j=1}^{m} u_j L\phi_j(x_n)\phi_i(x_n)w_n = f(x_n)\phi_i(x_n)w_n, \quad 1 \leq n \leq m$$

ove i $\{w_n\}_n$ sono i pesi di quadratura gaussiana, da cui, sommando su n,

$$\sum_{n=1}^{m}\left(\sum_{j=1}^{m} u_j L\phi_j(x_n)\phi_i(x_n)w_n\right) = \sum_{j=1}^{m} u_j\left(\sum_{n=1}^{m} L\phi_j(x_n)\phi_i(x_n)w_n\right) =$$
$$= \sum_{n=1}^{m} f(x_n)\phi_i(x_n)w_n, \quad 1 \leq i \leq m$$

che è precisamente la formulazione di Galerkin pseudospettrale (13.5). □

13.3.1 Condizioni al bordo

Consideriamo il problema

$$\begin{cases} Lu(x) = f(x) \\ u(a) = \alpha \\ u'(b) = \beta \end{cases}$$

Con il metodo di collocazione, si ha

$$\begin{cases} \displaystyle\sum_{j=1}^{m} u_j L\phi_j(x_n) = f(x_n), \quad 1 \leq n \leq m-2 \\ \displaystyle\sum_{j=1}^{m} u_j \phi_j(a) = \alpha \\ \displaystyle\sum_{j=1}^{m} u_j \phi_j'(b) = \beta \end{cases}$$

Anche in questo caso il metodo di collocazione può essere visto come un metodo di Galerkin pseudospettrale: basta prendere come nodi di collocazione gli $m-2$ nodi di quadratura gaussiana. Si ha poi

$$\begin{cases} \displaystyle\sum_{j=1}^{m} u_j\left(\sum_{n=1}^{m-2} L\phi_j(x_n)\phi_i(x_n)w_n\right) = \sum_{n=1}^{m-2} f(x_n)\phi_i(x_n)w_n, \quad 1 \leq i \leq m-2 \\ \displaystyle\sum_{j=1}^{m} u_j \phi_j(a) = \alpha \\ \displaystyle\sum_{j=1}^{m} u_j \phi_j'(b) = \beta \end{cases}$$

Alternativamente, si possono usare, come nodi di collocazione, quelli delle formule di quadratura di Gauss–Lobatto (che contengono i nodi al bordo).

Collocazione Gauss–Lobatto–Chebyshev

I polinomi di Chebyshev sono definiti da

$$T_j(x) = \cos(j \arccos(x)), \quad -1 \le x \le 1$$

e soddisfano

$$\int_{-1}^{1} \frac{T_j(x)T_i(x)}{\sqrt{1-x^2}} \mathrm{d}x = \begin{cases} \pi & i = j = 0 \\ \dfrac{\pi}{2} & i = j \neq 0 \\ 0 & i \neq j \end{cases}$$

(lo si vede con il cambio di variabile $x = \cos\theta$ e applicando le formule di Werner). I nodi di Gauss–Chebyshev–Lobatto sono $x_n = \cos((n-1)\pi/(m-1))$, $n = 1, 2, \ldots, m$. Possiamo allora definire la seguente famiglia di funzioni ortonormali

$$\phi_1(x) = \sqrt{\frac{1}{\pi}} T_0(x), \quad \phi_j(x) = \sqrt{\frac{2}{\pi}} T_{j-1}(x), \quad 2 \le j \le m$$

Ricordando la formula di ricorrenza tra polinomi di Chebyshev, possiamo scrivere

$$\begin{aligned} &\phi_1(x) = \sqrt{\frac{1}{\pi}}, \quad \phi_2(x) = \sqrt{\frac{2}{\pi}} x, \quad \phi_3(x) = 2x\phi_2(x) - \sqrt{2}\phi_1(x), \\ &\phi_{j+1}(x) = 2x\phi_j(x) - \phi_{j-1}(x), \quad 3 \le j \le m-1 \end{aligned}$$

Da qui, possiamo calcolare anche la derivata prima e seconda delle funzioni:

$$\begin{aligned} &\phi_1'(x) = 0, \quad \phi_2'(x) = \sqrt{\frac{2}{\pi}}, \quad \phi_3'(x) = 2\phi_2(x) + 2x\phi_2'(x), \\ &\phi_{j+1}'(x) = 2\phi_j(x) + 2x\phi_j'(x) - \phi_{j-1}'(x), \quad 3 \le j \le m-1 \end{aligned}$$

$$\begin{aligned} &\phi_1''(x) = 0, \quad \phi_2''(x) = 0, \quad \phi_3''(x) = 4\phi_2'(x), \\ &\phi_{j+1}''(x) = 4\phi_j'(x) + 2x\phi_j''(x) - \phi_{j-1}''(x), \quad 3 \le j \le m-1 \end{aligned}$$

Conviene calcolare le matrici

$$\mathbb{T} = \begin{bmatrix} \phi_1(x_1) & \phi_1(x_2) & \dots & \phi_1(x_m) \\ \phi_2(x_1) & \phi_2(x_2) & \dots & \phi_2(x_m) \\ \vdots & \vdots & \dots & \vdots \\ \phi_m(x_1) & \phi_m(x_2) & \dots & \phi_m(x_m) \end{bmatrix}$$

$$\mathbb{T}' = \begin{bmatrix} \phi_1'(x_1) & \phi_1'(x_2) & \dots & \phi_1'(x_m) \\ \phi_2'(x_1) & \phi_2'(x_2) & \dots & \phi_2'(x_m) \\ \vdots & \vdots & \dots & \vdots \\ \phi_m'(x_1) & \phi_m'(x_2) & \dots & \phi_m'(x_m) \end{bmatrix}$$

$$\mathbb{T}'' = \begin{bmatrix} \phi_1''(x_1) & \phi_1''(x_2) & \dots & \phi_1''(x_m) \\ \phi_2''(x_1) & \phi_2''(x_2) & \dots & \phi_2''(x_m) \\ \vdots & \vdots & \dots & \vdots \\ \phi_m''(x_1) & \phi_m''(x_2) & \dots & \phi_m''(x_m) \end{bmatrix}$$

Consideriamo, a titolo di esempio, il seguente problema modello

$$\begin{cases} -u''(x) + q(x)u(x) = f(x) \\ u(-1) = \alpha \\ u'(1) = \beta \end{cases}$$

Il sistema lineare risultante da risolvere per il metodo di collocazione Gauss–Chebyshev–Lobatto (per il momento *senza* tener conto delle condizioni al bordo) è

$$\left(-\mathbb{T}''^{\mathrm{T}} + \begin{bmatrix} q(x_1) & & & \\ & q(x_2) & & \\ & & \ddots & \\ & & & q(x_m) \end{bmatrix} \mathbb{T}^{\mathrm{T}} \right) \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{bmatrix}$$

Per imporre le condizioni al bordo, si sostituisce la prima riga della matrice con la prima riga di $\mathbb{T}^{\mathrm{T}}$ e il primo elemento del termine noto con α. Poi, l'ultima riga della matrice con l'ultima riga di $\mathbb{T}'^{\mathrm{T}}$ e l'ultimo elemento del termine noto con β. Una volta noti i coefficienti u_j, si ricostruisce la soluzione nello spazio fisico tramite

$$\begin{bmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_m) \end{bmatrix} = \mathbb{T}^{\mathrm{T}} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$$

Capitolo 14

Esercizi

1. Si risolva il problema ai limiti

$$\begin{cases} u''(x) = u(x) + x, \quad x \in (0,1) \\ u(0) = 0 \\ u(1) = 0 \end{cases} \tag{14.1}$$

usando il metodo delle differenze finite del secondo ordine. Sapendo che la soluzione esatta è $u(x) = (e^x - e^{-x})/(e - e^{-1}) - x$, si mostri inoltre l'ordine del metodo mediante un grafico logaritmico-logaritmico dell'errore in norma infinito.

2. Si risolva il problema ai limiti

$$\begin{cases} u''(x) + u'(x) + u(x) - \cos(x) = 0, \quad x \in (0, \pi/2) \\ u(0) = 0 \\ u(\pi/2) = 1 \end{cases}$$

usando il metodo delle differenze finite del secondo ordine. Si mostri inoltre l'ordine del metodo mediante un grafico logaritmico-logaritmico dell'errore in norma infinito rispetto ad una soluzione di riferimento.

3. Si risolva il problema ai limiti

$$\begin{cases} u''(x) + u'(x) + u(x) - \cos(x) = 0, \quad x \in (0, \pi/2) \\ u'(0) = 1 \\ u(\pi/2) = 1 \end{cases}$$

usando il metodo delle differenze finite del secondo ordine. Si mostri inoltre l'ordine del metodo mediante un grafico logaritmico-logaritmico dell'errore in norma infinito rispetto ad una soluzione di riferimento.

4. Si risolva il problema ai limiti

$$\begin{cases} u''(x) = \cos(u(x)), & x \in (0,1) \\ u(0) = 0 \\ u(1) = 1 \end{cases}$$

usando il metodo delle differenze finite del secondo ordine.

5. Si risolva il problema ai limiti

$$\begin{cases} -\dfrac{\mathrm{d}}{\mathrm{d}x}\left((1+x)\dfrac{\mathrm{d}}{\mathrm{d}x}u(x)\right) = 1, & x \in (0,1) \\ u(0) = 0 \\ u(1) = 0 \end{cases}$$

Si mostri inoltre l'ordine del metodo mediante un grafico logaritmico-logaritmico dell'errore in norma infinito rispetto alla soluzione esatta.

6. Si risolva il problema ai limiti

$$\begin{cases} u''(x) = 20u'(x) + u(x), & x \in (0,1) \\ u(0) = 0 \\ u(1) = 1 \end{cases}$$

Visto l'andamento della soluzione, si implementi uno schema di differenze finite su nodi non equispaziati secondo una distribuzione di tipo coseno. Si confrontino gli errori rispetto alla soluzione analitica.

7. Si ricavi la relazione di ricorrenza dei polinomi ortonormali nell'intervallo $[-\infty, \infty]$ rispetto alla funzione peso $w(x) = e^{-\alpha^2 x^2}$

8. Noti gli zeri dei polinomi di Legendre e i pesi di quadratura della rispettiva formula gaussiana, si ricavino i nodi e i pesi di una formula gaussiana nell'intervallo $[a, b]$ rispetto al peso $w(x) = 1$.

9. Si risolva il problema ai limiti (14.1) usando il metodo di collocazione con polinomi di Legendre. Gli N nodi di collocazione in $[a, b]$ e la valutazione dei polinomi di Legendre e delle loro derivate sono dati dalla function `[L,x,L1,L2] = legendrepolynomials(N,a,b)`. Si mostri inoltre l'ordine del metodo mediante un grafico logaritmico-logaritmico dell'errore in norma infinito.

Parte 2

ODEs (Equazioni differenziali ordinarie)

Capitolo 15

Introduzione

Consideriamo il sistema di equazioni differenziali ordinarie (ODEs) *ai valori iniziali (initial value problem)*

$$\begin{cases} y_1'(t) = f_1(t, y_1(t), y_2(t), \dots, y_d(t)) \\ y_2'(t) = f_2(t, y_1(t), y_2(t), \dots, y_d(t)) \\ \quad \vdots \\ y_d'(t) = f_d(t, y_1(t), y_2(t), \dots, y_d(t)) \end{cases}$$

con dato iniziale

$$\begin{cases} y_1(t_0) = y_{1_0} \\ y_2(t_0) = y_{2_0} \\ \quad \vdots \\ y_d(t_0) = y_{d_0} \end{cases}$$

che può essere riscritto, in forma compatta,

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad t > t_0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases} \tag{15.1}$$

Assumiamo $\boldsymbol{y}_0 \in \mathbb{R}^d$ e $\boldsymbol{f} \colon [t_0, +\infty) \times \mathbb{R}^d \to \mathbb{R}^d$ globalmente lipschitziana nel secondo argomento

$$\|\boldsymbol{f}(t, \boldsymbol{x}) - \boldsymbol{f}(t, \boldsymbol{y})\| \leq \lambda \|\boldsymbol{x} - \boldsymbol{y}\| \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$$

Allora il sistema (15.1) ha un'unica soluzione.

15.1 Riduzione in forma autonoma

Un sistema in forma *non autonoma*

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), & t > t_0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

può essere ricondotto in forma autonoma mediante l'introduzione della variabile

$$y_{d+1}(t) = t$$

Si giunge a

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(y_{d+1}(t), \boldsymbol{y}(t)), & t > t_0 \\ y'_{d+1}(t) = 1, & t > t_0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \\ y_{d+1}(t_0) = t_0 \end{cases}$$

15.2 Equazioni di ordine superiore al primo

Le equazioni differenziali di ordine d del tipo

$$\begin{cases} y^{(d)}(t) = f(t, y(t), y'(t), \ldots, y^{(d-1)}(t)), & t > t_0 \\ y(t_0) = y_{0,0} \\ y'(t_0) = y_{0,1} \\ \vdots \\ y^{(d-1)}(t_0) = y_{0,d-1} \end{cases}$$

(cioè in cui vengono prescritti i valori iniziali della funzione e delle derivate) si possono ricondurre ad un sistema di ODEs di ordine uno, mediante la sostituzione

$$\begin{cases} y_1(t) = y(t) \\ y_2(t) = y'(t) \\ \vdots \\ y_d(t) = y^{(d-1)}(t) \end{cases}$$

dando così luogo a

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)) \\ \boldsymbol{y}(t_0) = [y_{0,0}, y_{0,1}, \ldots, y_{0,d-1}]^{\mathrm{T}} \end{cases}$$

ove

$$\boldsymbol{f}(t, \boldsymbol{y}(t)) = [y_2(t), y_3(t), \ldots, y_d(t), f(t, y_1(t), y_2(t), \ldots, y_{d-1}(t))]^{\mathrm{T}}$$

Capitolo 16

Metodi ad un passo

16.1 Metodo di Eulero

Il *metodo di Eulero* (o *Eulero esplicito*, o *forward Euler*) si basa sull'approssimazione

$$\boldsymbol{y}'(t) \approx \frac{\boldsymbol{y}(t) - \boldsymbol{y}(t_0)}{t - t_0}$$
$$\boldsymbol{f}(t, \boldsymbol{y}(t)) \approx \boldsymbol{f}(t_0, \boldsymbol{y}(t_0))$$

per cui $\boldsymbol{y}(t) \approx \boldsymbol{y}(t_0) + (t - t_0)\boldsymbol{f}(t_0, \boldsymbol{y}(t_0))$. Pertanto l'approssimazione di $\boldsymbol{y}(t)$ è ottenuta per interpolazione lineare a partire da $(t_0, \boldsymbol{f}(t_0, \boldsymbol{y}(t_0)))$, con pendenza $\boldsymbol{f}(t_0, \boldsymbol{y}(t_0))$. Può essere visto anche come applicazione della formula di quadratura del rettangolo (estremo sinistro) alla soluzione analitica

$$\boldsymbol{y}(t) = \boldsymbol{y}(t_0) + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\mathrm{d}\tau$$

Data la sequenza $t_0, t_1 = t_0 + k, t_2 = t_0 + 2k, \ldots, t_n = t_0 + nk, \ldots$, ove k è il *passo temporale* (o *time step*), lo schema numerico che ne risulta è

$$\begin{aligned} \boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + k\boldsymbol{f}(t_n, \boldsymbol{y}_n), \quad n \geq 0, \\ \boldsymbol{y}_0 &= \boldsymbol{y}(t_0) \end{aligned} \tag{16.1}$$

ove $\boldsymbol{y}_n \approx \boldsymbol{y}(t_n)$. In pratica, $\boldsymbol{y}_{n+1}$ è la soluzione approssimata al tempo $t_n + k$, mediante un passo del metodo di Eulero, del sistema

$$\begin{cases} \boldsymbol{y}^{*\prime}(t) = \boldsymbol{f}(t, \boldsymbol{y}^*(t)) \\ \boldsymbol{y}^*(t_n) = \boldsymbol{y}_n \end{cases}$$

Se consideriamo l'intervallo temporale $[t_0, t_0 + t^*]$, indichiamo con $\boldsymbol{y}_{n,m}$ (oppure $\boldsymbol{y}_{n,k}$), $n \leq m$, la soluzione approssimata al tempo t_n mediante un generico metodo $\boldsymbol{y}_{n+1,k} = \boldsymbol{\mathcal{Y}}_n(\boldsymbol{f}, k, \boldsymbol{y}_0, \boldsymbol{y}_{1,k}, \ldots, \boldsymbol{y}_{n,k})$ per la soluzione del sistema differenziale (15.1), ove il passo temporale è $k = t^*/m$.

Definizione 1. *La quantità* $\boldsymbol{y}(t_{n+1}) - \boldsymbol{\mathcal{Y}}_n(\boldsymbol{f}, k, \boldsymbol{y}_0, \boldsymbol{y}(t_1), \ldots, \boldsymbol{y}(t_n))$ *si chiama* errore locale *del metodo.*

Se consideriamo il problema differenziale

$$\begin{cases} \boldsymbol{y}^{*\prime}(t) = \boldsymbol{f}(t, \boldsymbol{y}^*(t)) \\ \boldsymbol{y}^*(t_n) = \boldsymbol{y}(t_n) \end{cases}$$

si vede che l'errore locale coincide con $\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}^*_{n+1}$, cioè con la differenza tra la soluzione esatta al tempo t_{n+1} e la soluzione approssimata al tempo t_{n+1} che si *otterrebbe* applicando il metodo numerico al problema differenziale e supponendo esatta la soluzione al tempo t_n. È chiaro allora che ad ogni passo si commette un nuovo errore che si accumula con l'errore prodotto ai passi precedenti. Per il metodo di Eulero, si ha

$$\begin{aligned} &\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}(t_n) - k\boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) = \\ &\qquad = \boldsymbol{y}(t_n) + k\boldsymbol{y}'(t_n) + \mathcal{O}(k^2) - \boldsymbol{y}(t_n) - k\boldsymbol{y}'(t_n) = \mathcal{O}(k^2) \end{aligned} \tag{16.2}$$

Poiché ad ogni passo si commette un errore di ordine $\mathcal{O}(k^2)$ e i passi sono $m = t^*/k$, *se tutto va bene* alla fine si commette un errore di ordine $\mathcal{O}(k)$. È giustificata allora la seguente

Definizione 2. *Un metodo* $\boldsymbol{y}_{n+1} = \boldsymbol{\mathcal{Y}}_n(\boldsymbol{f}, k, \boldsymbol{y}_0, \ldots, \boldsymbol{y}_n)$ *è di ordine* p *se* $\boldsymbol{y}(t_{n+1}) - \boldsymbol{\mathcal{Y}}_n(\boldsymbol{f}, k, \boldsymbol{y}(t_0), \ldots, \boldsymbol{y}(t_n)) = \mathcal{O}(k^{p+1})$*, per* $k \to 0$*, per qualunque* $\boldsymbol{f}$ *analitica e* $0 \le n \le m - 1$.

La definizione sopra dice in verità che il metodo è *almeno* di ordine p. Un metodo di ordine $p \ge 1$ si dice *consistente di ordine* p, o semplicemente *consistente*. Se $\boldsymbol{y}(t_{n+1}) - \boldsymbol{\mathcal{Y}}_n(\boldsymbol{f}, k, \boldsymbol{y}(t_0), \ldots, \boldsymbol{y}(t_n)) = 0$ il metodo si dice *fortemente consistente*. Dunque il metodo di Eulero è consistente di ordine p. Si tratta ora di dimostrare che *tutto va bene*.

Definizione 3. *Il metodo* $\boldsymbol{y}_{n+1,k} = \boldsymbol{\mathcal{Y}}_n(\boldsymbol{f}, k, \boldsymbol{y}_0, \boldsymbol{y}_{1,k}, \ldots, \boldsymbol{y}_{n,k})$ *è* convergente *se*

$$\lim_{k \to 0^+} \max_{0 \le n \le m} \|\boldsymbol{e}_{n,k}\| = 0$$

ove $\boldsymbol{e}_{n,k} = \boldsymbol{y}_{n,k} - \boldsymbol{y}(t_n)$*. La quantità* $\max_n \|\boldsymbol{e}_{n,k}\|$ *si chiama* errore globale.

Teorema 6. *Il metodo di Eulero è convergente.*

Dimostrazione. Assumiamo $\boldsymbol{f}$ (e dunque $\boldsymbol{y}$) analitica. Dalle uguaglianze

$$\begin{aligned} \boldsymbol{y}_{n+1,k} &= \boldsymbol{y}_{n,k} + k\boldsymbol{f}(t_n, \boldsymbol{y}_{n,k}) && \text{(definizione del metodo)} \\ \boldsymbol{y}(t_{n+1}) &= \boldsymbol{y}(t_n) + k\boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) + \mathcal{O}(k^2) && \text{(errore locale (16.2))} \end{aligned}$$

si ricava

$$\boldsymbol{e}_{n+1,k} = \boldsymbol{e}_{n,k} + k[\boldsymbol{f}(t_n, \boldsymbol{e}_{n,k} + \boldsymbol{y}(t_n)) - \boldsymbol{f}(t_n, \boldsymbol{y}(t_n))] + \mathcal{O}(k^2)$$

da cui

$$\begin{aligned}\|\boldsymbol{e}_{n+1,k}\| &\leq \|\boldsymbol{e}_{n,k}\| + k\|\boldsymbol{f}(t_n, \boldsymbol{e}_{n,k} + \boldsymbol{y}(t_n)) - \boldsymbol{f}(t_n, \boldsymbol{y}(t_n))\| + ck^2 \leq \\ &\leq (1 + k\lambda)\|\boldsymbol{e}_{n,k}\| + ck^2, \quad c > 0\end{aligned}$$

Allora

$$\|\boldsymbol{e}_{n,k}\| \leq \frac{c}{\lambda} k[(1 + k\lambda)^n - 1], \quad 0 \leq n \leq m$$

(si dimostra per induzione). Poiché $1 + k\lambda < \mathrm{e}^{k\lambda}$, $(1 + k\lambda)^n < \mathrm{e}^{nk\lambda} \leq \mathrm{e}^{mk\lambda} = \mathrm{e}^{t^*\lambda}$. Dunque

$$\|\boldsymbol{e}_{n,k}\| \leq \frac{c}{\lambda} k(e^{t^*\lambda} - 1), \quad 0 \leq n \leq m$$

da cui

$$\lim_{k \to 0^+} \max_{0 \leq n \leq m} \|\boldsymbol{e}_{n,k}\| = 0$$

□

In particolare, l'errore globale tende a 0 come k, come ci si aspettava.

16.2 Metodo dei trapezi

Il *metodo dei trapezi* (o metodo di *Crank–Nicolson*) si basa sull'approssimazione

$$\begin{aligned}\boldsymbol{y}'(t) &\approx \frac{\boldsymbol{y}(t) - \boldsymbol{y}(t_0)}{t - t_0} \\ \boldsymbol{f}(t, \boldsymbol{y}(t)) &\approx \frac{1}{2}(\boldsymbol{f}(t_0, \boldsymbol{y}(t_0)) + \boldsymbol{f}(t, \boldsymbol{y}(t)))\end{aligned}$$

Può essere visto anche come applicazione della formula di quadratura del trapezio alla soluzione analitica

$$\boldsymbol{y}(t) = \boldsymbol{y}(t_0) + \int_{t_0}^{t} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\mathrm{d}\tau$$

Data la sequenza $t_0, t_1 = t_0 + k, t_2 = t_0 + 2k, \ldots, t_n = t_0 + nk, \ldots$, lo schema numerico che ne risulta è

$$\begin{aligned}\boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + \frac{k}{2}(\boldsymbol{f}(t_n, \boldsymbol{y}_n) + \boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1})), \quad n \geq 0, \\ \boldsymbol{y}_0 &= \boldsymbol{y}(t_0)\end{aligned} \tag{16.3}$$

ove $\boldsymbol{y}_n \approx \boldsymbol{y}(t_n)$. Dato lo schema

$$\boldsymbol{y}_{n+1} - \boldsymbol{y}_n - \frac{k}{2}(\boldsymbol{f}(t_n, \boldsymbol{y}_n) + \boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1})) = 0$$

sostituendo $\boldsymbol{y}_n$ con $\boldsymbol{y}(t_n)$ e $\boldsymbol{y}_{n+1}$ con $\boldsymbol{y}(t_{n+1})$ si ottiene

$$\begin{aligned}
&\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}(t_n) - \frac{k}{2}(\boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) + \boldsymbol{f}(t_{n+1}, \boldsymbol{y}(t_{n+1}))) = \\
&= \boldsymbol{y}(t_n) + k\boldsymbol{y}'(t_n) + \frac{k^2}{2}\boldsymbol{y}''(t_n) + \mathcal{O}(k^3) - \boldsymbol{y}(t_n) - \frac{k}{2}(\boldsymbol{y}'(t_n) + \boldsymbol{y}'(t_{n+1})) = \\
&\quad k\boldsymbol{y}'(t_n) + \frac{k^2}{2}\boldsymbol{y}''(t_n) - \frac{k}{2}(\boldsymbol{y}'(t_n) + \boldsymbol{y}'(t_n) + k\boldsymbol{y}''(t_n)) + \mathcal{O}(k^3) = \mathcal{O}(k^3)
\end{aligned}$$

Dunque il metodo dei trapezi è di ordine 2.

Teorema 7. *Il metodo dei trapezi è convergente.*

Dimostrazione. Si ha

$$\begin{aligned}
\boldsymbol{e}_{n+1,k} = \boldsymbol{e}_{n,k} &+ \frac{k}{2}(\boldsymbol{f}(t_n, \boldsymbol{y}_n) - \boldsymbol{f}(t_n, \boldsymbol{y}(t_n))) + \\
&+ \frac{k}{2}(\boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1}) - \boldsymbol{f}(t_{n+1}, \boldsymbol{y}(t_{n+1}))) + \mathcal{O}(k^3)
\end{aligned}$$

da cui

$$\|\boldsymbol{e}_{n+1,k}\| \leq \|\boldsymbol{e}_{n,k}\| + \frac{k\lambda}{2}(\|\boldsymbol{e}_{n,k}\| + \|\boldsymbol{e}_{n+1,k}\|) + ck^3, \quad c > 0$$

Se $k < 2/\lambda$,

$$\|\boldsymbol{e}_{n+1,k}\| \leq \left(\frac{1 + \frac{k\lambda}{2}}{1 - \frac{k\lambda}{2}}\right)\|\boldsymbol{e}_{n,k}\| + \left(\frac{c}{1 - \frac{k\lambda}{2}}\right)k^3$$

Allora

$$\|\boldsymbol{e}_{n,k}\| \leq \frac{ck^2}{\lambda}\left[\left(\frac{1 + \frac{k\lambda}{2}}{1 - \frac{k\lambda}{2}}\right)^n - 1\right], \quad 0 \leq n \leq m$$

(si dimostra per induzione). Si conclude osservando che

$$\left(\frac{1 + \frac{k\lambda}{2}}{1 - \frac{k\lambda}{2}}\right)^n = \left(1 + \frac{k\lambda}{1 - \frac{k\lambda}{2}}\right)^n \leq \exp\left(\frac{nk\lambda}{1 - k\lambda/2}\right) \leq \exp\left(\frac{t^*\lambda}{1 - k\lambda/2}\right)$$

□

Entrambi i metodi descritti sono *ad un passo* (cioè la soluzione $\boldsymbol{y}_{n+1}$ dipende esplicitamente solo da $\boldsymbol{y}_n$). Il metodo dei trapezi è però *implicito,* cioè la soluzione $\boldsymbol{y}_{n+1}$ è implicitamente definita dall'equazione (in generale non lineare)

$$F_n(\boldsymbol{y}_{n+1}) = \boldsymbol{y}_{n+1} - \frac{k}{2}\boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1}) - \boldsymbol{y}_n - \frac{k}{2}\boldsymbol{f}(t_n, \boldsymbol{y}_n) = 0$$

16.3 θ-metodo

Il θ-metodo è una generalizzazione dei metodi precedenti e si scrive

$$\begin{aligned} \boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + k[\theta \boldsymbol{f}(t_n, \boldsymbol{y}_n) + (1-\theta)\boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1})], \quad n \geq 0 \\ \boldsymbol{y}_0 &= \boldsymbol{y}(t_0) \end{aligned} \tag{16.4}$$

È facile verificare che

$$\begin{aligned} \boldsymbol{y}(t_{n+1}) - \boldsymbol{y}(t_n) - k[\theta \boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) + (1-\theta)\boldsymbol{f}(t_{n+1}, \boldsymbol{y}(t_{n+1})] = \\ = \left(\theta - \frac{1}{2}\right) k^2 \boldsymbol{y}''(t_n) + \left(\frac{\theta}{2} - \frac{1}{3}\right) k^3 \boldsymbol{y}'''(t_n) + \mathcal{O}(k^4) \end{aligned}$$

e dunque il metodo ha ordine due se $\theta = \frac{1}{2}$, e ordine uno altrimenti. In particolare, se $\boldsymbol{y}''(t)$ è nulla, tale è l'errore locale per il θ-metodo. E se $\theta = \frac{1}{2}$ e $\boldsymbol{y}'''(t)$ è nulla, tale è l'errore locale. Poiché però ad ogni passo si commettono errori di troncamento (nel caso esplicito) o di troncamento e approssimazione (nel caso implicito, in cui c'è da risolvere un sistema non lineare), per avere la convergenza è comunque necessario che questi non si accumulino in maniera distruttiva.

Teorema 8. *Il θ-metodo, per $\theta \in [0, 1]$, è convergente.*

Osserviamo che:

- il metodo è esplicito per $\theta = 1$ (e in tal caso si riduce al metodo di Eulero esplicito);
- il metodo è di ordine due per $\theta = \frac{1}{2}$ (e in tal caso si riduce al metodo dei trapezi);
- il metodo per $\theta = 0$ si chiama *Eulero implicito (backward Euler)*;
- per $\theta = \frac{2}{3}$ il metodo è di ordine uno, ma il termine contenente la derivata terza della soluzione è annullato.

Nel caso implicito ($\theta \neq 1$), ad ogni passo n si deve risolvere un sistema di equazioni non lineari $F_n(\boldsymbol{x}) = 0$, $\boldsymbol{x} = \boldsymbol{y}_{n+1}$, ove

$$F_n(\boldsymbol{x}) = \boldsymbol{x} - k(1-\theta)\boldsymbol{f}(t_{n+1}, \boldsymbol{x}) - \boldsymbol{y}_n - k\theta\boldsymbol{f}(t_n, \boldsymbol{y}_n).$$

La matrice Jacobiana associata (utile per l'applicazione del metodo di Newton) è

$$J_n(\boldsymbol{x}) = \left(I - k(1-\theta)\frac{\partial f_i(t_{n+1}, \boldsymbol{x})}{\partial x_j} \right)_{ij}$$

Ovviamente, si può scegliere un metodo di Newton modificato. Il vettore iniziale per il calcolo di $\boldsymbol{y}_{n+1}$ è di solito la soluzione al passo precedente $\boldsymbol{y}_n$. Il calcolo di $\boldsymbol{y}_{n+1}$ a partire da $\boldsymbol{y}_n$ con il metodo di Newton avviene dunque secondo il seguente algoritmo:

- $r = 0$
- $\boldsymbol{y}_{n+1}^{(r)} = \boldsymbol{y}_n$
- $J_n(\boldsymbol{y}_{n+1}^{(r)})\boldsymbol{\delta}^{(r)} = -F_n(\boldsymbol{y}_{n+1}^{(r)})$
- WHILE $\|\boldsymbol{\delta}^{(r)}\| >$ Newt_tol

  $\boldsymbol{y}_{n+1}^{(r+1)} = \boldsymbol{y}_{n+1}^{(r)} + \boldsymbol{\delta}^{(r)}$
  $r = r + 1$
  $J_n(\boldsymbol{y}_{n+1}^{(r)})\boldsymbol{\delta}^{(r)} = -F_n(\boldsymbol{y}_{n+1}^{(r)})$

  END
- $\boldsymbol{y}_{n+1} = \boldsymbol{y}_{n+1}^{(r)}$

La tolleranza Newt_tol va presa tenendo conto che si sta comunque commettendo un errore proporzionale a k^2 (trapezi) o addirittura k.

16.3.1 Caso lineare

Un caso molto frequente è quello lineare autonomo a coefficienti costanti

$$\begin{cases} \boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{b} \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

con passo di integrazione k costante. In tal caso, il metodo si scrive

$$(I - k(1-\theta)A)\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + k\theta A\boldsymbol{y}_n + k\boldsymbol{b}$$

Nel caso implicito, si tratta dunque di risolvere un sistema lineare di matrice $I - k(1-\theta)A$ ad ogni passo. Pertanto, per problemi di piccola dimensione, è conveniente precalcolare la fattorizzazione LU della matrice. Altrimenti, si può considerare un metodo iterativo, ove si scelga come vettore iniziale per il calcolo di $\boldsymbol{y}_{n+1}$ la soluzione al passo precedente $\boldsymbol{y}_n$.

16.4 Verifica della correttezza dell'implementazione

Supponiamo di aver implementato un metodo di ordine p per la soluzione del sistema differenziale

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)) \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

e di volerne testare la corretta implementazione. L'idea è quella di creare una soluzione artificiale $\boldsymbol{x}(t)$, inserirla nell'equazione e calcolarne il residuo

$$\boldsymbol{x}'(t) - \boldsymbol{f}(t, \boldsymbol{x}(t)) = \boldsymbol{g}(t)$$

A questo punto, si risolve il sistema differenziale

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)) + \boldsymbol{g}(t) = \hat{\boldsymbol{f}}(t, \boldsymbol{y}(t)) \\ \boldsymbol{y}(t_0) = \boldsymbol{x}(t_0) \end{cases}$$

fino ad un tempo $t_0 + t^*$ fissato, con due discretizzazioni di passo costante $k_1 = t^*/m_1$ e $k_2 = t^*/m_2$, rispettivamente. Si avranno errori finali $\boldsymbol{e}_{m_1,k_1} = \|\boldsymbol{y}_{m_1,k_1} - \boldsymbol{x}(t_0 + t^*)\| = Ck_1^p$ e $\boldsymbol{e}_{m_2,k_2} = \|\boldsymbol{y}_{m_2,k_2} - \boldsymbol{x}(t_0 + t^*)\| = Ck_2^p$. Si ha dunque

$$\frac{\boldsymbol{e}_{m_2,k_2}}{\boldsymbol{e}_{m_1,k_1}} = \left(\frac{k_2}{k_1}\right)^p ,$$

da cui

$$\log \boldsymbol{e}_{m_2,k_2} - \log \boldsymbol{e}_{m_1,k_1} = p(\log k_2 - \log k_1) = -p(\log m_2 - \log m_1).$$

Dunque, rappresentando in un grafico logaritmico-logaritmico l'errore in dipendenza dal numero di passi, la pendenza della retta corrisponde all'ordine del metodo, cambiato di segno. Tale verifica è valida anche nel caso di passi non costanti.

Nel caso $\boldsymbol{f}(t, \boldsymbol{y}(t))$ sia particolarmente complicato, invece di calcolare il residuo, si può calcolare una *soluzione di riferimento* $\boldsymbol{y}_{\bar{m},\bar{k}}$ e poi confrontare con essa le soluzioni $\boldsymbol{y}_{m_1,k_1}$ e $\boldsymbol{y}_{m_2,k_2}$, ove $m_1, m_2 \ll \bar{m}$. In questo caso, però, si può mostrare solo che il metodo converge con l'ordine giusto ad *una* soluzione, non necessariamente quella giusta.

Falsa superconvergenza

Supponiamo che la soluzione esatta di un problema differenziale sia $y = 1$ e che, discretizzandolo con $m = 1, 2, 4, 8$ passi si ottengano le soluzioni $y_1 = 17$, $y_2 = 9$, $y_4 = 5$ e $y_8 = 3$. Supponiamo poi di averne calcolato una soluzione di riferimento con $m = 16$ passi $\bar{y} = y_{16} = 2$. In Figura 16.1 il grafico degli errori che si ottiene.

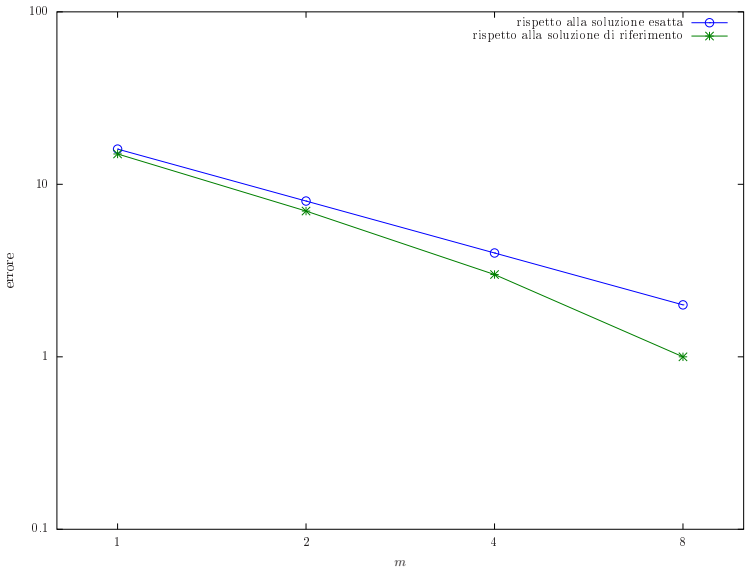


Figura 16.1: Ordine dei metodi

Capitolo 17

Metodi multistep

17.1 Metodi di Adams–Bashforth

Invece di costruire la soluzione $\boldsymbol{y}_{n+1}$ a partire dalla sola soluzione al passo precedente $\boldsymbol{y}_n$, si può pensare di usare le soluzioni di più passi precedenti. Fissato s numero naturale maggiore di 0 e una discretizzazione dell'intervallo $[t_0, t_0 + t^*]$ in m passi di ampiezza costante k, data la formula di risoluzione

$$\boldsymbol{y}(t_{n+s}) = \boldsymbol{y}(t_{n+s-1}) + \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\mathrm{d}\tau \tag{17.1}$$

l'idea è quella di sostituire la funzione integranda in (17.1) con il "suo" polinomio interpolatore sui nodi equispaziati $t_n, t_{n+1}, \ldots, t_{n+s-1}$ ($t_{n+j} = t_n + jk$)

$$\boldsymbol{p}(\tau) = \sum_{j=0}^{s-1} L_j(\tau)\boldsymbol{f}(t_{n+j}, \boldsymbol{y}_{n+j})$$

ove $L_j(t)$ è il polinomio elementare di Lagrange di grado $s-1$ definito da $L_j(t_{n+i}) = \delta_{ij}$. Poiché $\boldsymbol{p}(t_{n+j}) = \boldsymbol{f}(t_{n+j}, \boldsymbol{y}_{n+j})$, $j = 0, 1, \ldots, s-1$ (e non, ovviamente, $\boldsymbol{p}(t_{n+j}) = \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j}))$), dobbiamo supporre di avere già a disposizione i valori $\boldsymbol{y}_{n+j} \approx \boldsymbol{y}(t_{n+j})$, $j = 0, 1, \ldots, s-1$. Si ha dunque

$$\int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{p}(\tau)\mathrm{d}\tau = \sum_{j=0}^{s-1} \left(\int_{t_{n+s-1}}^{t_{n+s}} L_j(\tau)\mathrm{d}\tau \right) \boldsymbol{f}(t_{n+j}, \boldsymbol{y}_{n+j}) = k \sum_{j=0}^{s-1} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}_{n+j})$$

da cui il metodo esplicito *multistep Adams–Bashforth*

$$\boldsymbol{y}_{n+s} = \boldsymbol{y}_{n+s-1} + k \sum_{j=0}^{s-1} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}_{n+j}) \tag{17.2}$$

I coefficienti b_j non dipendono da n e neanche da k: infatti

$$b_m = \frac{1}{k}\int_{t_{n+s-1}}^{t_{n+s}} \prod_{\substack{i=0\\ i\neq j}}^{s-1} \frac{\tau - t_{n+i}}{t_{n+j} - t_{n+i}} \mathrm{d}\tau = \int_0^1 \prod_{\substack{i=0\\ i\neq j}}^{s-1} \frac{t_{n+s-1} + rk - t_{n+i}}{t_{n+j} - t_{n+i}} \mathrm{d}r =$$

$$= \int_0^1 \prod_{\substack{i=0\\ i\neq m}}^{s-1} \frac{((s-1-i)k + rk)}{(j-i)k} \mathrm{d}r = \int_0^1 \prod_{\substack{i=0\\ i\neq j}}^{s-1} \frac{((s-1-i) + r)}{(j-i)} \mathrm{d}r$$

Dunque possono essere calcolati una volta per tutte. Calcoliamo l'ordine di tale metodo: come al solito, dobbiamo valutare l'espressione

$$\boldsymbol{y}(t_{n+s}) - \boldsymbol{y}(t_{n+s-1}) - k\sum_{j=0}^{s-1} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j}))$$

L'ultimo termine è l'integrale del polinomio $\boldsymbol{q}(\tau)$ di grado $s-1$ che interpola $\boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j}))$, $j = 0, 1, \ldots, s-1$. Dunque, per $t_{n+s-1} \leq \tau \leq t_{n+s}$,

$$\|\boldsymbol{q}(\tau) - \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\| \leq \frac{\|\boldsymbol{y}^{(s+1)}(\bar{\tau})\|}{s!} \underbrace{|(\tau - t_n)\cdot \ldots \cdot (\tau - t_{n+s-1})|}_{s \text{ termini}} \leq$$

$$\leq \frac{\|\boldsymbol{y}^{(s+1)}(\bar{\tau})\|}{s!} s! k^s = \mathcal{O}(k^s), \quad t_{n+s-1} \leq \tau \leq t_{n+s}$$

e quindi

$$\boldsymbol{y}(t_{n+s}) - \boldsymbol{y}(t_{n+s-1}) - k\sum_{j=0}^{s-1} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j})) =$$

$$- k\sum_{j=0}^{s-1} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j})) + \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\mathrm{d}\tau =$$

$$- \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{q}(\tau)\mathrm{d}\tau + \int_{t_{n+s-1}}^{t_{n+s}} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\mathrm{d}\tau = \mathcal{O}(k^{s+1})$$

perché un ulteriore fattore k deriva dal fatto che si integra in un intervallo di ampiezza k. Quindi, se anche $\boldsymbol{y}_n = \boldsymbol{y}(t_n) + \mathcal{O}(k^{s+1})$, $n = 1, 2, \ldots, s-1$ (queste approssimazioni *non* possono essere ottenute con il metodo stesso), il metodo è di ordine p. Calcoliamo esplicitamente i metodi che corrispondono a $s = 1$ e $s = 2$. Se $s = 1$, dobbiamo cercare il polinomio di grado 0 che interpola $\boldsymbol{f}(t_n, \boldsymbol{y}_n)$. È ovviamente $\boldsymbol{p}(\tau) \equiv \boldsymbol{f}(t_n, \boldsymbol{y}_n)$ e $b_0 = 1$, quindi

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + k\boldsymbol{f}(t_n, \boldsymbol{y}_n)$$

e si ritrova il metodo di Eulero. Nel caso $s = 2$, il polinomio interpolatore è

$$\boldsymbol{p}(\tau) = \frac{t_{n+1} - \tau}{k}\boldsymbol{f}(t_n, \boldsymbol{y}_n) + \frac{\tau - t_n}{k}\boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1})$$

e dunque

$$b_0 = \frac{1}{k}\int_{t_{n+1}}^{t_{n+2}} \frac{t_{n+1} - \tau}{k}\mathrm{d}\tau = -\frac{1}{2}, \quad b_1 = \frac{1}{k}\int_{t_{n+1}}^{t_{n+2}} \frac{\tau - t_n}{k}\mathrm{d}\tau = \frac{3}{2}$$

da cui

$$\boldsymbol{y}_{n+2} = \boldsymbol{y}_{n+1} - \frac{k}{2}\boldsymbol{f}(t_n, \boldsymbol{y}_n) + \frac{3k}{2}\boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1})$$

Il valore $\boldsymbol{y}_1$ può essere ricavato, per esempio, con il metodo dei trapezi, poiché si ha, in tal caso, $\boldsymbol{y}_1 = \boldsymbol{y}(t_1) + \mathcal{O}(k^3)$.

17.2 Metodi lineari multistep

Una semplice generalizzazione del metodo di Adams–Bashforth è permettere la possibilità che il metodo sia implicito:

$$\boldsymbol{y}_{n+s} = \boldsymbol{y}_{n+s-1} + k\sum_{j=0}^{s} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}_{n+j}) \tag{17.3}$$

(questa famiglia di metodi prende il nome di *Adams–Moulton*) e, ancora più in generale,

$$\sum_{j=0}^{s} a_j \boldsymbol{y}_{n+j} = k\sum_{j=0}^{s} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}_{n+j}) \tag{17.4}$$

con la normalizzazione $a_s = 1$. Rientra allora in questa famiglia anche il θ-metodo ($a_0 = -1$, $b_0 = \theta$, $b_1 = (1 - \theta)$). Il metodo è di ordine p se, come al solito,

$$\sum_{j=0}^{s} a_j \boldsymbol{y}(t_{n+j}) - k\sum_{j=0}^{s} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j})) = \mathcal{O}(k^{s+1})$$

per ogni funzione $\boldsymbol{f}$ analitica e $0 \le n \le m - s$. Siccome la verifica può risultare molto tediosa, risulta utile il seguente

Teorema 9. *Dato un metodo multistep* (17.4)*, definiamo i due polinomi*

$$\rho(w) = \sum_{j=0}^{s} a_j w^j, \quad \sigma(w) = \sum_{j=0}^{s} b_j w^j$$

Allora il metodo è di ordine p se e solo se esiste $c \neq 0$ tale che

$$\rho(w) - \sigma(w) \cdot \ln w = c(w-1)^{p+1} + \mathcal{O}\left((w-1)^{p+2}\right) \quad \textit{per } w \to 1$$

Prima di vedere la traccia della dimostrazione, proviamo ad applicare il teorema a qualche caso noto. Per il metodo di Eulero si ha $\rho(w) = w - 1$ e $\sigma(w) = 1$. Posto $\xi = w - 1$, si ha

$$\xi - 1 \cdot \left(\xi - \frac{\xi^2}{2} + \mathcal{O}(\xi^3)\right) = \frac{\xi^2}{2} + \mathcal{O}(\xi^3)$$

e dunque il metodo è di ordine 1, come noto. Per il metodo dei trapezi si ha $\rho(w) = w - 1$ e $\sigma(w) = (w+1)/2$. Posto $\xi = w - 1$, si ha

$$\xi - \left(\frac{\xi}{2} + 1\right) \cdot \left(\xi - \frac{\xi^2}{2} + \frac{\xi^3}{3} - \mathcal{O}(\xi^4)\right) = -\frac{\xi^3}{12} + \mathcal{O}(\xi^4)$$

e dunque il metodo è di ordine 2, come noto.

Traccia della dimostrazione del Teorema 9. Si ha

$$\begin{aligned}
&\sum_{j=0}^{s} a_j \boldsymbol{y}(t_{n+j}) - k \sum_{j=0}^{s} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j})) = \\
&\qquad \sum_{j=0}^{s} a_j \boldsymbol{y}(t_n + jk) - k \sum_{j=0}^{s} b_j \boldsymbol{y}'(t_n + jk) = \\
&\sum_{j=0}^{s} a_j \sum_{i=0}^{\infty} \boldsymbol{y}^{(i)}(t_n) \frac{j^i k^i}{i!} - k \sum_{j=0}^{s} b_j \sum_{i=0}^{\infty} \boldsymbol{y}^{(i+1)}(t_n) \frac{j^i k^i}{i!} = \\
&= \left(\sum_{j=0}^{s} a_j\right) \boldsymbol{y}(t_n) + \sum_{i=1}^{\infty} \frac{1}{i!} \left(\sum_{j=0}^{s} j^i a_j - i \sum_{j=0}^{s} j^{i-1} b_j\right) k^i \boldsymbol{y}^{(i)}(t_n)
\end{aligned}$$

Dunque l'ordine è p se e solo se i coefficienti delle potenze fino a p di k sono nulli, e cioè:

$$\begin{aligned}
&\sum_{j=0}^{s} a_j = 0 \\
&\sum_{j=0}^{s} j^i a_j - i \sum_{j=0}^{s} j^{i-1} b_j = 0, \quad i = 1, 2, \ldots, p
\end{aligned} \tag{17.5}$$

Per finire la dimostrazione, si calcola lo sviluppo in serie di Taylor di $\rho(\mathrm{e}^z) - \sigma(\mathrm{e}^z)z$ per $z \to 0$ e si osserva che esso è $\mathcal{O}(z^{p+1})$ se e solo se vale (17.5). Posto

$w = \mathrm{e}^z$, l'ordine è p se e solo se lo sviluppo di $\rho(w) - \sigma(w) \cdot \ln w$ per $w \to 1$ vale

$$-c(\ln w)^{p+1} + \mathcal{O}\left((\ln w)^{p+2}\right) = c(w-1)^{p+1} + \mathcal{O}\left((w-1)^{p+2}\right)$$

□

In realtà, anche le condizioni (17.5) sono molto utili per determinare l'ordine di un metodo multistep: per esempio, per il metodo Adams–Moulton a due passi

$$\begin{aligned} \boldsymbol{y}_{n+2} - 3\boldsymbol{y}_{n+1} + 2\boldsymbol{y}_n = \\ = \frac{k}{12}\left[-5\boldsymbol{f}(t_n, \boldsymbol{y}_n) - 20\boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1}) + 13\boldsymbol{f}(t_{n+2}, \boldsymbol{y}_{n+2})\right] \end{aligned} \quad (17.6)$$

si ha

$$\begin{aligned} a_0 + a_1 + a_2 &= 0 \\ a_1 + 2a_2 &= b_0 + b_1 + b_2 \quad \Rightarrow \text{ordine (almeno) } 1 \\ a_1 + 4a_2 &= 2(b_1 + 2b_2) \quad \Rightarrow \text{ordine (almeno) } 2 \\ a_1 + 8a_2 &\neq 3(b_1 + 4b_2) \quad \Rightarrow \text{ordine } 2 \end{aligned}$$

17.2.1 Metodi BDF

I metodi BDF *(Backward differentiation formulas)* sono metodi multistep impliciti a s passi, di ordine s e con $\sigma(w) = \beta w^s$. Dato $\sigma(w)$ e le condizioni d'ordine, si può costruire $\rho(w)$. Poiché però tali metodi sono della forma

$$\sum_{j=0}^{s} a_j \boldsymbol{y}_{n+j} = k b_s \boldsymbol{f}(t_{n+s}, \boldsymbol{y}_{n+s})$$

e $kb_s\boldsymbol{f}(t_{n+s}, \boldsymbol{y}_{n+s}) \approx kb_s\boldsymbol{y}'(t_n + sk)$, conviene cercare una combinazione lineare di $\boldsymbol{y}_{n+j}$, $j = 0, 1, \ldots, s$ che approssimi $kb_s\boldsymbol{y}'(t_{n+s})$. Si procede dunque con lo sviluppo in serie di Taylor di $\boldsymbol{y}(t_{n+j})$, $j = 0, 1, \ldots, s$, centrato in $\boldsymbol{y}(t_n + sk)$. Per esempio, per $s = 1$,

$$\begin{cases} \boldsymbol{y}(t_n) = \boldsymbol{y}(t_n + k) - k\boldsymbol{y}'(t_n + k) + \mathcal{O}(k^2) \\ \boldsymbol{y}(t_n + k) = \boldsymbol{y}(t_n + k) \end{cases}$$

da cui $a_1 = 1$ e $a_0 = -1$. Dunque, il metodo BDF di ordine 1 è il metodo di Eulero implicito *(backward Euler)* ed è di ordine 1, come noto. Per $s = 2$

$$\begin{cases} \boldsymbol{y}(t_n) = \boldsymbol{y}(t_n + 2k) - 2k\boldsymbol{y}'(t_n + 2k) + \dfrac{4k^2}{2}\boldsymbol{y}''(t_n + 2k) + \mathcal{O}(k^3) \\ \boldsymbol{y}(t_n + k) = \boldsymbol{y}(t_n + 2k) - k\boldsymbol{y}'(t_n + 2k) + \dfrac{k^2}{2}\boldsymbol{y}''(t_n + 2k) + \mathcal{O}(k^3) \\ \boldsymbol{y}(t_n + 2k) = \boldsymbol{y}(t_n + 2k) \end{cases}$$

da cui

$$\begin{cases} a_2 = 1 \\ a_0 + a_1 + a_2 = 0 \\ -2a_0 - a_1 = b_2 \\ 2a_0 + \dfrac{a_1}{2} = 0 \end{cases} \Rightarrow \begin{cases} a_0 = \dfrac{1}{3} \\ a_1 = -\dfrac{4}{3} \\ a_2 = 1 \\ b_2 = \dfrac{2}{3} \end{cases}$$

e il metodo è di ordine 2.

I metodi BDF sono gli unici metodi multistep in cui non è difficile calcolare i coefficienti anche nel caso di passi temporali variabili. Sempre per $s = 2$, se $t_{n+1} = t_n + k_{n+1}$ e $t_{n+2} = t_{n+1} + k_{n+2}$, allora

$$\begin{cases} \boldsymbol{y}(t_n) = \boldsymbol{y}(t_n + k_{n+1} + k_{n+2}) - (k_{n+1} + k_{n+2})\boldsymbol{y}'(t_n + k_{n+1} + k_{n+2}) + \\ \quad + \dfrac{(k_{n+1} + k_{n+2})^2}{2}\boldsymbol{y}''(t_n + k_{n+1} + k_{n+2}) + \dots \\ \boldsymbol{y}(t_n + k_{n+1}) = \boldsymbol{y}(t_n + k_{n+1} + k_{n+2}) - k_{n+2}\boldsymbol{y}'(t_n + k_{n+1} + k_{n+2}) + \\ \quad + \dfrac{k_{n+2}^2}{2}\boldsymbol{y}''(t_n + k_{n+1} + k_{n+2}) + \dots \\ \boldsymbol{y}(t_n + k_{n+1} + k_{n+2}) = \boldsymbol{y}(t_n + k_{n+1} + k_{n+2}) \end{cases}$$

da cui i coefficienti per calcolare $\boldsymbol{y}_{n+2}$

$$\begin{cases} a_2 = 1 \\ a_0 + a_1 + a_2 = 0 \\ -a_0(k_{n+1} + k_{n+2}) - a_1 k_{n+2} = b_2 k_{n+2} \\ \dfrac{a_0}{2}(k_{n+1} + k_{n+2})^2 + \dfrac{a_1}{2}k_{n+2}^2 = 0 \end{cases} \Rightarrow \begin{cases} a_0 = \dfrac{k_{n+2}^2}{(k_{n+1} + 2k_{n+2})k_{n+1}} \\ a_1 = -\dfrac{(k_{n+1} + k_{n+2})^2}{(k_{n+1} + 2k_{n+2})k_{n+1}} \\ a_2 = 1 \\ b_2 = \dfrac{(k_{n+1} + k_{n+2})}{(k_{n+1} + 2k_{n+2})} \end{cases}$$

Va notato però che il metodo che ne risulta in generale *non* converge se $k_{n+2}/k_{n+1} \geq 1 + \sqrt{2}$. E rimane aperto poi il problema di scegliere come cambiare il passo (vedi paragrafo 18.1). Questi metodi risultano particolarmente vantaggiosi quando la valutazione della funzione $\boldsymbol{f}$ è particolarmente onerosa, poiché permettono di raggiungere un ordine elevato con una sola valutazione (nel caso lineare, altrimenti è necessario valutare $\boldsymbol{f}$ in un ciclo di Newton ad ogni passo temporale).

17.3 Consistenza e stabilità

Dalle condizioni d'ordine (17.5), si vede che un metodo lineare multistep è consistente se

$$\rho(1) = 1, \quad \rho'(1) = \sigma(1)$$

La consistenza, però, non è sufficiente ad assicurare la convergenza di un metodo. Consideriamo il metodo Adams–Moulton a due passi (17.6) applicato al semplice problema differenziale

$$\begin{cases} y'(t) = 0, & t > 0 \\ y(0) = 1 \end{cases}$$

la cui soluzione è evidentemente $y(t) \equiv 1$. L'applicazione del metodo porge

$$y_{n+2} - 3y_{n+1} + 2y_n = 0$$

Se $y_{n+1} = y_n$ per ogni n, allora $y_{n+2} = 1$ per ogni n. A causa degli errori di troncamento numerico, però, potrà succedere che per un certo $\bar{n}$ si ha $y_{\bar{n}+1} = y_{\bar{n}} + \varepsilon$. Allora

$$\begin{aligned} y_{\bar{n}+2} &= 3y_{\bar{n}+1} - 2y_{\bar{n}} = y_{\bar{n}} + 3\varepsilon \\ y_{\bar{n}+3} &= 3y_{\bar{n}+2} - 2y_{\bar{n}+1} = y_{\bar{n}} + 7\varepsilon \\ y_{\bar{n}+4} &= 3y_{\bar{n}+3} - 2y_{\bar{n}+2} = y_{\bar{n}} + 15\varepsilon \\ &\vdots \\ y_{\bar{n}+j} &= y_{\bar{n}} + (2^j - 1)\varepsilon \end{aligned}$$

Dunque, se il numero di passi è $m = t^*/k$, si ha

$$\lim_{m\to\infty} y_{m,k} = \lim_{k\to 0} y_{m,k} = \infty$$

Abbiamo quindi un metodo la cui soluzione numerica diverge facendo tendere il passo temporale a 0 (cioè proprio l'opposto di quanto dovrebbe succedere). È proprio un piccolo errore commesso ad un passo che si accumula in maniera distruttiva. Infatti, posto, come al solito,

$$\boldsymbol{y}^*_{n+s} + \sum_{j=0}^{s-1} a_j \boldsymbol{y}(t_{n+j}) = k \sum_{j=0}^{s-1} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j})) + k b_s \boldsymbol{f}(t_{n+s}, \boldsymbol{y}^*_{n+s})$$

l'errore al passo $n+s$ può essere espresso come

$$\boldsymbol{e}_{n+s} = \boldsymbol{y}_{n+s} - \boldsymbol{y}(t_{n+s}) = (\boldsymbol{y}_{n+s} - \boldsymbol{y}^*_{n+s}) + (\boldsymbol{y}^*_{n+s} - \boldsymbol{y}(t_{n+s})) \tag{17.7}$$

ove il secondo termine è l'errore locale e il primo termine tiene conto dell'accumulazione degli errori ai passi precedenti, cioè delle *perturbazioni* tra la soluzione esatta e la soluzione numerica ai passi precedenti. È giustificata allora (in analogia con quanto visto al paragrafo 10.2.7) la seguente

Definizione 4. *Dato un metodo lineare multistep* (17.4)*, siano* $\boldsymbol{z}_n^i$*,* $i = 1, 2$*, due perturbazioni della soluzione definite da*

$$\begin{aligned} \boldsymbol{z}_j^i &= \boldsymbol{y}_j + \boldsymbol{\delta}_j^i, && j = 0, 1, \ldots, s-1 \\ \sum_{j=0}^{s} a_j \boldsymbol{z}_{n+j}^i &= k \sum_{j=0}^{s} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{z}_{n+j}^i) + \boldsymbol{\delta}_{n+s}^i, && 0 \le n \le m - s \end{aligned}$$

Se da $\max_n \|\boldsymbol{\delta}_n^1 - \boldsymbol{\delta}_n^2\| \le \varepsilon$ *segue* $\max_n \|\boldsymbol{z}_n^1 - \boldsymbol{z}_n^2\| \le C\varepsilon$*, allora il metodo* (17.4) *si dice* (zero-)stabile*.*

Guardando la rappresentazione dell'errore (17.7), si vede che il primo termine $(\boldsymbol{y}_{n+s} - \boldsymbol{y}_{n+s}^*)$ è la differenza tra due particolari soluzioni perturbate $\boldsymbol{z}_n^1$ e $\boldsymbol{z}_n^2$ corrispondenti a

$$\boldsymbol{\delta}_j^1 = 0, \quad 0 \le j \le m$$

e

$$\begin{aligned} \boldsymbol{\delta}_j^2 &= \boldsymbol{y}(t_j) - \boldsymbol{y}_j, && 0 \le j \le s-1 \\ \boldsymbol{\delta}_{n+s}^2 &= -k \sum_{j=0}^{s} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{z}_{n+j}^2) + \\ &\quad + k \sum_{j=0}^{s-1} b_j \boldsymbol{f}(t_{n+j}, \boldsymbol{y}(t_{n+j})) + k b_s \boldsymbol{f}(t_{n+s}, \boldsymbol{y}_{n+s}^*) + \\ &\quad + \sum_{j=0}^{s} a_j \boldsymbol{z}_{n+j}^2 - \boldsymbol{y}_{n+s}^* - \sum_{j=0}^{s-1} a_j \boldsymbol{y}(t_{n+j}), && 0 \le n \le m - s \end{aligned}$$

Dunque, un metodo è stabile se piccole perturbazioni della soluzione ad un certo passo rimangono limitate nella soluzione ad ogni passo. Cerchiamo un criterio semplice per determinare se un metodo è stabile. Consideriamo ancora il problema visto all'inizio di questo paragrafo. Supponiamo che θ sia una radice del polinomio $\rho(w)$ (che d'ora in poi chiameremo *caratteristico*) di un metodo multistep almeno consistente. Allora $z_n = k\theta^n + 1$ è *una* soluzione numerica (in un metodo multistep a s stadi vi è una certa libertà di costruire i primi z_n, $n = 0, \ldots, s-1$, da cui la *non* unicità di soluzione). Infatti

$$\sum_{j=0}^{s} a_j z_{n+j} = k\theta^n \sum_{j=0}^{s} a_j \theta^j + \sum_{j=0}^{s} a_j = k\theta^n \rho(\theta) + 0 = 0$$

Essa è generata dai valori iniziali $y_j = 1$, $j = 0, 1, \ldots, s-1$, *perturbati* dalla quantità $\delta_j = \theta^j k$, perfettamente lecita dal punto di vista della consistenza, e supponendo che non intervengano ulteriori perturbazioni. Se vogliamo che z_m converga alla soluzione analitica $y(t) \equiv 1$ per $k \to 0$ (o, equivalentemente, per $m \to \infty$), deve essere $|\theta| \leq 1$. Se θ è radice doppia di $\rho(w)$, allora è radice anche di $\rho'(w)$ e pertanto soddisfa

$$\rho'(\theta) = \sum_{j=0}^{s} a_j j \theta^{j-1} = 0$$

Allora anche $z_n = kn\theta^n + 1$ è una soluzione numerica. Infatti

$$\sum_{j=0}^{s} a_j z_{n+j} = k^p \left(\theta^n \sum_{j=0}^{s} a_j n \theta^j + \theta^{n+1} \sum_{j=0}^{s} a_j j \theta^{j-1} \right) + \sum_{j=0}^{s} a_j =$$
$$= k^p \theta^n n \rho(\theta) + k^p \theta^{n+1} \rho'(\theta) + 0 = 0$$

ed è generata dai valori iniziali $y_j = 1$ perturbati da $\delta_j = kj\theta^j$, $0 \leq j \leq s-1$. Se vogliamo che z_m converga alla soluzione analitica, deve essere $|\theta| < 1$. Se il polinomio caratteristico ha radici semplici θ tali che $|\theta| \leq 1$ e radici doppie θ tali che $|\theta| < 1$, diremo che il polinomio soddisfa la *condizione delle radici*. La condizione delle radici risulta anche sufficiente, assieme alla consistenza, per la convergenza del metodo lineare multistep. Si ha infatti il seguente teorema fondamentale:

Teorema 10. *Un metodo lineare multistep è convergente se e solo se è consistente e il suo polinomio caratteristico soddisfa la condizione delle radici.*

Ritornando al metodo (17.6), si ha che $\theta = 2$ è radice del polinomio caratteristico e infatti $y_n = k2^n + 1$ è una soluzione del problema differenziale che abbiamo usato come modello. Pertanto, il metodo non è stabile.

Come corollario al teorema precedente, abbiamo che ogni metodo ad un passo è stabile (perché $\rho(w) = w - 1$) e che i metodi di Adams–Bashforth sono stabili (perché $\rho(w) = w^s - w^{s-1}$). Esiste un limite superiore per l'ordine di un metodo a s passi, dato dal seguente

Teorema 11 (Prima barriera di Dahlquist)**.** *Il massimo ordine per un metodo a s passi convergente è $2\lfloor (s+2)/2 \rfloor$ se implicito e s se esplicito.*

Per quanto riguarda i metodi BDF (speciali metodi impliciti) si ha che sono convergenti (cioè sono stabili) solo per $1 \leq s \leq 6$.

Capitolo 18

Metodi di Runge–Kutta

I metodi lineari multistep lasciano aperti alcuni problemi. Come calcolare i valori iniziai per i metodi di ordine elevato? Abbiamo visto che il massimo ordine per un metodo ad un passo convergente è 2 se implicito (lo raggiunge il solo metodo dei trapezi). È possibile modificarlo e renderlo esplicito (e dunque di più facile applicazione)? Si possono costruire metodi di ordine elevato e che permettano un passo temporale "adattabile" all'andamento della soluzione? Cominciamo a rispondere alla seconda domanda: una modifica abbastanza ovvia al metodo dei trapezi

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \frac{k}{2}(\boldsymbol{f}(t_n, \boldsymbol{y}_n) + \boldsymbol{f}(t_{n+1}, \boldsymbol{y}_{n+1}))$$

per renderlo esplicito è sostituire $\boldsymbol{y}_{n+1}$ con $\boldsymbol{y}_n + k\boldsymbol{f}(t_n, \boldsymbol{y}_n)$ così da avere

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + \frac{k}{2}(\boldsymbol{f}(t_n, \boldsymbol{y}_n) + \boldsymbol{f}(t_{n+1}, \boldsymbol{y}_n + k\boldsymbol{f}(t_n, \boldsymbol{y}_n)) \tag{18.1}$$

Da un punto di vista "logico", esso può essere definito come

$$\begin{aligned}
\boldsymbol{\xi}_1 &= \boldsymbol{y}_n \approx \boldsymbol{y}(t_n) \\
\boldsymbol{\xi}_2 &= \boldsymbol{y}_n + k\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) \approx \boldsymbol{y}(t_{n+1}) \\
\boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + \frac{k}{2}(\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) + \boldsymbol{f}(t_{n+1}, \boldsymbol{\xi}_2)) \approx \boldsymbol{y}(t_{n+1})
\end{aligned}$$

Un altro modo di rendere esplicito il metodo dei trapezi è sostituire la media delle funzioni $\boldsymbol{f}$ con la funzione $\boldsymbol{f}$ valutata nel "punto medio"

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + k\boldsymbol{f}\left(t_n + \frac{k}{2}, \boldsymbol{y}_n + \frac{k}{2}\boldsymbol{f}(t_n, \boldsymbol{y}_n)\right) \tag{18.2}$$

cioè

$$\begin{aligned}\boldsymbol{\xi}_1 &= \boldsymbol{y}_n \approx \boldsymbol{y}(t_n)\\ \boldsymbol{\xi}_2 &= \boldsymbol{y}_n + \frac{k}{2}\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) \approx \boldsymbol{y}\left(t_n + \frac{k}{2}\right)\\ \boldsymbol{y}_{n+1} &= \boldsymbol{y}_n + k\boldsymbol{f}\left(t_n + \frac{k}{2}, \boldsymbol{\xi}_2\right) \approx \boldsymbol{y}(t_{n+1})\end{aligned}$$

L'idea generale dei metodi espliciti di *Runge–Kutta* è quella, come al solito, di sostituire l'integrale nella formula risolutiva

$$\boldsymbol{y}(t_{n+1}) = \boldsymbol{y}(t_n) + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(\tau, \boldsymbol{y}(\tau))\mathrm{d}\tau$$

con una formula di quadratura su nodi $t_n + c_j k$, $1 \le j \le \nu$ nell'intervallo $[t_n, t_{n+1}]$. Si giunge quindi a

$$\boldsymbol{y}(t_{n+1}) \approx \boldsymbol{y}(t_n) + k\sum_{j=1}^{\nu} b_j \boldsymbol{f}(t_n + c_j k, \boldsymbol{y}(t_n + c_j k))$$

Si tratta ora di trovare delle approssimazioni $\boldsymbol{\xi}_j$ di $\boldsymbol{y}(t_n + c_j k)$. Si procede ricorsivamente in questo modo

$$\begin{cases} \boldsymbol{y}(t_n) \approx \boldsymbol{y}_n = \boldsymbol{\xi}_1 \quad (\Rightarrow\ c_1 = 0)\\ \boldsymbol{y}(t_n + c_2 k) \approx \boldsymbol{y}_n + c_2 k\boldsymbol{f}(t_n, \boldsymbol{y}_n) = \boldsymbol{y}_n + a_{2,1}k\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) = \boldsymbol{\xi}_2\\ \boldsymbol{y}(t_n + c_3 k) \approx \boldsymbol{y}_n + a_{3,1}k\boldsymbol{f}(t_n, \boldsymbol{\xi}_1) + a_{3,2}k\boldsymbol{f}(t_n + c_2 k, \boldsymbol{\xi}_2) = \boldsymbol{\xi}_3\\ \qquad\vdots\\ \boldsymbol{y}(t_n + c_\nu k) \approx \boldsymbol{y}_n + k\sum_{j=1}^{\nu-1} a_{\nu,j}\boldsymbol{f}(t_n + c_j k, \boldsymbol{\xi}_j) = \boldsymbol{\xi}_\nu\\ \boldsymbol{y}_{n+1} = \boldsymbol{y}_n + k\sum_{j=1}^{\nu} b_j \boldsymbol{f}(t_n + c_j k, \boldsymbol{\xi}_j) \end{cases}$$

ove i parametri c_j, b_j e $a_{i,j}$ sono da determinare in modo da ottenere l'ordine desiderato. Il numero ν indica il numero di *stadi*. I parametri c_j, b_j e $a_{i,j}$ si racchiudono di solito nel *tableau di Butcher*. Se $\nu = 1$, ci si riconduce al

$$
\begin{array}{c|ccccccc}
0 & & & & & & & \\
c_2 & a_{2,1} & & & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & & & \\
\vdots & \vdots & \vdots & \ddots & & & & \\
c_{\nu-1} & a_{\nu-1,1} & a_{\nu-1,2} & \dots & a_{\nu-1,\nu-2} & & & \\
c_\nu & a_{\nu,1} & a_{\nu,2} & \dots & a_{\nu,\nu-2} & a_{\nu,\nu-1} & & \\
\hline
 & b_1 & b_2 & \dots & b_{\nu-2} & b_{\nu-1} & b_\nu
\end{array}
$$

Tabella 18.1: Tableau di Butcher.

metodo di Eulero. Per $\nu = 2$, l'ordine si ricava al solito modo

$$
\begin{aligned}
\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}(t_n) - kb_1\boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) - kb_2\boldsymbol{f}(t_n + c_2k, \boldsymbol{y}(t_n) + a_{2,1}k\boldsymbol{f}(t_n, \boldsymbol{y}(t_n))) = \\
= \boldsymbol{y}(t_n) + k\boldsymbol{y}'(t_n) + \frac{k^2}{2}\boldsymbol{y}''(t_n) + \mathcal{O}(k^3) - \boldsymbol{y}(t_n) - kb_1\boldsymbol{y}'(t_n) + \\
-kb_2\left[\boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) + \frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n))c_2k + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t_n, \boldsymbol{y}(t_n))a_{2,1}k\boldsymbol{y}'(t_n) + \mathcal{O}(k^2)\right] = \\
= k\boldsymbol{y}'(t_n) + \frac{k^2}{2}\boldsymbol{y}''(t_n) + \mathcal{O}(k^3) - kb_1\boldsymbol{y}'(t_n) - kb_2\boldsymbol{y}'(t_n) + \\
- kb_2\frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n))c_2k - kb_2\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t_n, \boldsymbol{y}(t_n))a_{2,1}k\boldsymbol{y}'(t_n) = \\
= k\boldsymbol{y}'(t_n) + \frac{k^2}{2}\boldsymbol{y}''(t_n) + \mathcal{O}(k^3) - kb_1\boldsymbol{y}'(t_n) - kb_2\boldsymbol{y}'(t_n) + \\
- kb_2\frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n))c_2k + \left[k^2a_{2,1}b_2\frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n)) - k^2a_{2,1}b_2\boldsymbol{y}''(t_n)\right] = \\
= k(1 - b_1 - b_2)\boldsymbol{y}'(t_n) + k^2\left(\frac{1}{2} - a_{2,1}b_2\right)\boldsymbol{y}''(t_n) + \\
- k^2(b_2c_2 - a_{2,1}b_2)\frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n)) + \mathcal{O}(k^3)
\end{aligned}
$$

Dunque l'ordine è due se

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

Tabella 18.2: Metodi Runge–Kutta espliciti di ordine 2.

$$\begin{cases} b_1 + b_2 = 1 \\ a_{2,1} b_2 = \dfrac{1}{2} \\ b_2 c_2 = b_2 a_{2,1} \end{cases} \tag{18.3}$$

da cui, per esempio, i metodi di ordine 2 riportati in Tabella 18.2. I primi due corrispondono ai due metodi visti all'inizio del capitolo e si chiamano, rispettivamente, *metodo di Eulero modificato* e *metodo di Heun*. Il punto cruciale di questo sviluppo in serie di Taylor è l'uguaglianza tra

$$\begin{aligned} \boldsymbol{y}'(t_n) + k\boldsymbol{y}''(t_n) &= \boldsymbol{y}'(t_n) + k\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) = \\ &= \boldsymbol{y}'(t_n) + k\frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n)) + k\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t_n, \boldsymbol{y}(t_n))\boldsymbol{y}'(t_n) \end{aligned}$$

e

$$\begin{aligned} \boldsymbol{f}(t_n + k, \boldsymbol{y}(t_n) + k\boldsymbol{f}(t_n, \boldsymbol{y}(t_n))) &= \boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) + \\ &+ \frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n))k + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t_n, \boldsymbol{y}(t_n))k\boldsymbol{f}(t_n, \boldsymbol{y}(t_n)) + \mathcal{O}(k^2) = \\ &= \boldsymbol{y}'(t_n) + k\frac{\partial \boldsymbol{f}}{\partial t}(t_n, \boldsymbol{y}(t_n)) + k\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(t_n, \boldsymbol{y}(t_n))\boldsymbol{y}'(t_n) + \mathcal{O}(k^2) \end{aligned}$$

(a meno di $\mathcal{O}(k^2)$) in cui le derivate di ordine superiore di $\boldsymbol{y}$ (e quindi di $\boldsymbol{f}$) sono sostituite da funzioni di funzioni $\boldsymbol{f}$.

Per ogni $\nu > 1$, il corrispondente sistema *non* lineare che si ottiene per la determinazione dell'ordine può avere infinite soluzioni. Solitamente si impone l'ulteriore vincolo

$$\sum_{j=1}^{i-1} a_{i,j} = c_i, \quad 2 \leq i \leq \nu$$

Da notare che la condizione

$$\sum_{j=1}^{\nu} b_j = 1$$

è necessaria per avere almeno ordine 1 (cioè la consistenza). Per quanto riguarda la stabilità, si può ripetere tutto il ragionamento fatto per il caso dei metodi multistep: si arriva ad osservare che il polinomio caratteristico è $\rho(w) = w - 1$ e pertanto tutti i metodi Runge–Kutta espliciti sono stabili. Ne discende la convergenza. Per quanto riguarda il massimo ordine che si può raggiungere dato il numero di stadi ν, si ha quanto riportato in Tabella 18.3.

numero stadi ν	1	2	3	4	5	6	7	8
massimo ordine p	1	2	3	4	4	5	6	6

Tabella 18.3: Massimo ordine per i metodi Runge–Kutta espliciti dato il numero di stadi.

Il numero di stadi equivale al numero di valutazioni della funzione $\boldsymbol{f}$ (e dunque al costo del metodo). Ai fini dell'implementazione, per evitare di calcolare più volte la funzione $\boldsymbol{f}$ negli stessi punti, si usa lo schema

$$\begin{cases} \boldsymbol{f}_i = \boldsymbol{f}(t_n + c_i k, \boldsymbol{y}_n + k \sum_{j=1}^{i-1} a_{i,j} \boldsymbol{f}_j), \quad i = 1, \dots, \nu \\ \boldsymbol{y}_{n+1} = \boldsymbol{y}_n + k \sum_{j=1}^{\nu} b_j \boldsymbol{f}_j \end{cases}$$

$$\begin{array}{c|ccccccc} c_1 & a_{1,1} & & & & & & \\ c_2 & a_{2,1} & a_{2,2} & & & & & \\ c_3 & a_{3,1} & a_{3,2} & a_{3,3} & & & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & & & \\ c_{\nu-1} & a_{\nu-1,1} & a_{\nu-1,2} & \dots & a_{\nu-1,\nu-2} & a_{\nu-1,\nu-1} & & \\ c_\nu & a_{\nu,1} & a_{\nu,2} & \dots & a_{\nu,\nu-2} & a_{\nu,\nu-1} & a_{\nu,\nu} \\ \hline & b_1 & b_2 & \dots & b_{\nu-2} & b_{\nu-1} & b_\nu \end{array}$$

$$\begin{array}{c|cccccc} c_1 & a_{1,1} & a_{1,2} & \dots & \dots & a_{1,\nu-1} & a_{1,\nu} \\ c_2 & a_{2,1} & a_{2,2} & \dots & \dots & a_{2,\nu-1} & a_{2,\nu} \\ c_3 & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ c_{\nu-1} & a_{\nu-1,1} & a_{\nu-1,2} & \dots & \dots & a_{\nu-1,\nu-1} & a_{\nu-1,\nu} \\ c_\nu & a_{\nu,1} & a_{\nu,2} & \dots & \dots & a_{\nu,\nu-1} & a_{\nu,\nu} \\ \hline & b_1 & b_2 & \dots & \dots & b_{\nu-1} & b_\nu \end{array}$$

Tabella 18.4: Tableaux di Butcher per i metodi Runge–Kutta semiimpliciti (sopra) e impliciti (sotto).

È possibile generalizzare i metodi espliciti di Runge–Kutta per ottenere metodi *semiimpliciti* e impliciti, i cui tableaux sono riportati in Tabella 18.4. Anche il θ-metodo può essere fatto rientrare nella classe dei metodi Runge–

Kutta semiimpliciti:

$$\begin{cases} \boldsymbol{\xi}_1 = \boldsymbol{y}_n \\ \boldsymbol{\xi}_2 = \boldsymbol{y}_n + k\theta \boldsymbol{f}(t_n, \boldsymbol{\xi}_1) + k(1-\theta)\boldsymbol{f}(t_n + k, \boldsymbol{\xi}_2) \\ \boldsymbol{y}_{n+1} = \boldsymbol{y}_n + k\theta \boldsymbol{f}(t_n, \boldsymbol{\xi}_1) + k(1-\theta)\boldsymbol{f}(t_n + k, \boldsymbol{\xi}_2) \end{cases}$$

o, in forma *implementativa*,

$$\begin{cases} \boldsymbol{f}_1 = \boldsymbol{f}(t_n, \boldsymbol{y}_n) \\ \boldsymbol{f}_2 = \boldsymbol{f}(t_n + k, \boldsymbol{y}_n + k\theta \boldsymbol{f}_1 + k(1-\theta)\boldsymbol{f}_2) \\ \boldsymbol{y}_{n+1} = \boldsymbol{y}_n + k\theta \boldsymbol{f}_1 + k(1-\theta)\boldsymbol{f}_2 \end{cases}$$

Dunque, abbiamo risposto anche alla prima domanda all'inizio di questo capitolo. Vediamo ora come rispondere alla terza domanda.

18.1 Metodi di Runge–Kutta *embedded*

Per i metodi ad un passo risulta alquanto facile adottare un passo temporale k_n variabile nel tempo (non così con i multistep, in cui i parametri dipendono dall'avere assunto i passi temporali costanti). In generale, più l'equazione ha un comportamento "lineare", più i passi possono essere presi grandi. Ma come adattare automaticamente il passo all'andamento della soluzione? Supponiamo di avere due metodi di Runge–Kutta espliciti di ordine $p-1$ e p rispettivamente, i cui tableaux sono riportati in Tabella 18.5. È chiaro che, dopo aver costruito il primo metodo, con una sola nuova valutazione della funzione $\boldsymbol{f}$ si può costruire il secondo metodo. Una tale coppia di metodi si dice *embedded* e si scrive di solito un unico tableau, come nella Tabella 18.6. Il fatto che per trovare metodi di Runge–Kutta sia necessario risolvere sistemi non lineari per i coefficienti, rende difficile *ma non impossibile* trovare coppie di metodi con tali caratteristiche.

Consideriamo il sistema differenziale

$$\begin{cases} \tilde{\boldsymbol{y}}'(t) = \boldsymbol{f}(t, \tilde{\boldsymbol{y}}(t)) \\ \tilde{\boldsymbol{y}}(t_n) = \boldsymbol{y}_n^{(p)} \end{cases}$$

ove $\boldsymbol{y}_n^{(p)}$ è l'approssimazione di $\boldsymbol{y}(t_n)$ ottenuta con il metodo Runge–Kutta di ordine p. Si ha allora

$$\|\boldsymbol{y}_{n+1}^{(p)} - \boldsymbol{y}_{n+1}^{(p-1)}\| = \|\boldsymbol{y}_{n+1}^{(p)} - \tilde{\boldsymbol{y}}(t_{n+1}) + \tilde{\boldsymbol{y}}(t_{n+1}) - \boldsymbol{y}_{n+1}^{(p-1)}\| = Ck_{n+1}^p, \quad (18.4)$$

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_{\nu-1} & a_{\nu-1,1} & a_{\nu-1,2} & \dots & a_{\nu-1,\nu-2} & \\
\hline
 & b_1 & b_2 & \dots & b_{\nu-2} & b_{\nu-1}
\end{array}
$$

$$
\begin{array}{c|cccccc}
0 & & & & & & \\
c_2 & a_{2,1} & & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & & \\
\vdots & \vdots & \vdots & \ddots & & & \\
c_{\nu-1} & a_{\nu-1,1} & a_{\nu-1,2} & \dots & a_{\nu-1,\nu-2} & & \\
c_\nu & a_{\nu,1} & a_{\nu,2} & \dots & a_{\nu,\nu-2} & a_{\nu,\nu-1} & \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \dots & \hat{b}_{\nu-2} & \hat{b}_{\nu-1} & \hat{b}_\nu
\end{array}
$$

Tabella 18.5: Metodi di Runge–Kutta di ordine $p-1$ e p.

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{3,1} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_\nu & a_{\nu,1} & a_{\nu,2} & \dots & a_{\nu,\nu-1} & \\
\hline
 & b_1 & b_2 & \dots & b_{\nu-1} & \\
\hline
 & \hat{b}_1 & \hat{b}_2 & \dots & \hat{b}_{\nu-1} & \hat{b}_\nu
\end{array}
$$

Tabella 18.6: Metodi di Runge–Kutta embedded di ordine $p-1$ e p.

per un opportuno $C > 0$, ove $k_{n+1} = t_{n+1} - t_n$ è il passo di integrazione e Ck_{n+1}^p è l'errore locale del metodo di ordine $p-1$. Se si vuole controllare tale errore si può allora imporre, ad ogni passo, che

$$\|\boldsymbol{y}_{n+1}^{(p)} - \boldsymbol{y}_{n+1}^{(p-1)}\| \leq \mathrm{tol_a} + \|\boldsymbol{y}_{n+1}^{(p)}\| \cdot \mathrm{tol_r} \tag{18.5}$$

rifiutando $\boldsymbol{y}_{n+1}^{(p)}$ (e $\boldsymbol{y}_{n+1}^{(p-1)}$) nel caso la disuguaglianza non sia soddisfatta e calcolando un nuovo passo di integrazione minore del precedente. Per calcolare il successivo passo di integrazione k, sia nel caso che la disuguaglianza sia stata soddisfatta o meno, imponiamo che valga

$$Ck^p = \mathrm{tol_a} + \|\boldsymbol{y}_{n+1}^{(p)}\| \cdot \mathrm{tol_r}$$

e, ricavando $1/C$ da (18.4), si ha

$$k = \left(\frac{\mathrm{tol_a} + \|\boldsymbol{y}_{n+1}^{(p)}\| \mathrm{tol_r}}{\|\boldsymbol{y}_{n+1}^{(p)} - \boldsymbol{y}_{n+1}^{(p-1)}\|} \right)^{1/p} \cdot k_{n+1}$$

Tale espressione indica il nuovo passo temporale $k_{n+2} = k$ nel caso la disuguaglianza (18.5) sia stata soddisfatta oppure la nuova misura del vecchio $k_{n+1} = k$ nel caso contrario. Per evitare che il passo di integrazione cambi troppo bruscamente, si può adottare una correzione del tipo

$$k = \min \left(2, \max \left(0.6, 0.9 \cdot \left(\frac{\mathrm{tol_a} + \|\boldsymbol{y}_{n+1}^{(p)}\| \mathrm{tol_r}}{\|\boldsymbol{y}_{n+1}^{(p)} - \boldsymbol{y}_{n+1}^{(p-1)}\|} \right)^{1/p} \right) \right) \cdot k_{n+1}$$

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

Tabella 18.7: Metodo di Runge–Kutta–Fehlberg.

Forse il più importante metodo di Runge–Kutta embedded è il Runge–Kutta–Fehlberg, di ordine (4)5, il cui tableau è riportato in Tabella 18.7.

Capitolo 19

A-stabilità

Purtroppo la consistenza e la stabilità di un metodo non sono sufficienti per avere un *buon* solutore di qualunque equazione differenziale ordinaria. Consideriamo infatti il seguente problema lineare

$$\begin{cases} y'(t) = \lambda y(t) & t > t_0 \\ y(t_0) = y_0 \end{cases} \tag{19.1}$$

la cui soluzione esatta $y(t) = \mathrm{e}^{\lambda t} y_0$ tende a zero, per $t \to +\infty$, se $\Re(\lambda) < 0$. Analizziamo il comportamento del metodo di Eulero per questo problema, supponendo di avere fissato il passo temporale k: si ha

$$y_{n+1} = y_n + k\lambda y_n = (1 + k\lambda) y_n$$

da cui

$$y_{n+1} = (1 + k\lambda)^n y_0$$

Si ha

$$\lim_{n\to\infty} y_n = 0 \Leftrightarrow |1 + k\lambda| < 1 \Leftrightarrow 1 + k^2\Re(\lambda)^2 + 2k\Re(\lambda) + k^2\Im(\lambda)^2 < 1$$

da cui

$$\lim_{n\to\infty} y_n = 0 \Leftrightarrow k < -\frac{2\Re(\lambda)}{|\lambda|^2}$$

Dunque, la soluzione numerica ottenuta con il metodo di Eulero ha lo stesso comportamento della soluzione analitica solo se il passo temporale è sufficientemente piccolo. Altrimenti, la soluzione può essere completamente diversa ($\lim_{n\to\infty}|y_n| = |y_0|$ o $\lim_{n\to\infty} y_n = \infty$). Nel caso di Eulero implicito, invece, si ha

$$y_n = \left(\frac{1}{1 - k\lambda}\right)^n y_0$$

da cui

$$\lim_{n\to\infty} y_n = 0 \Leftrightarrow |1 - k\lambda| > 1 \Leftrightarrow |1 - k\Re(\lambda) - k\mathrm{i}\Im(\lambda)| > 1$$

disuguaglianza sempre soddisfatta, poiché $\Re(\lambda) < 0$. Anche per il metodo dei trapezi la soluzione numerica tende a 0 per $n \to \infty$. Ma non è vero, in generale, per qualunque metodo implicito. Analizziamo infatti il comportamento generale del θ-metodo per questo problema: si ha

$$y_{n+1} = y_n + \theta k\lambda y_n + (1-\theta)k\lambda y_{n+1}$$

da cui

$$y_n = \left[\frac{1+\theta k\lambda}{1-(1-\theta)k\lambda}\right]^n y_0$$

Si ha

$$\lim_{n\to\infty} y_n = 0 \Leftrightarrow \left|\frac{1+\theta k\lambda}{1-(1-\theta)k\lambda}\right| < 1 \Leftrightarrow |1+\theta k\lambda| < |1-(1-\theta)k\lambda| \Leftrightarrow$$
$$0 < k^2\Re(\lambda)^2 - 2k\Re(\lambda) - 2k^2\theta\Re(\lambda)^2 + k^2\Im(\lambda)^2 - 2\theta k^2\Im(\lambda)^2$$

da cui

$$\lim_{n\to\infty} y_n = 0 \Leftrightarrow 0 < (1-2\theta)k^2|\lambda|^2 - 2k\Re(\lambda)$$

Se $1 - 2\theta \geq 0$, certamente la disequazione è soddisfatta. Altrimenti,

$$\lim_{n\to\infty} y_n = 0 \Leftrightarrow k < \frac{2\Re(\lambda)}{(1-2\theta)|\lambda|^2}, \quad (1-2\theta < 0) \tag{19.2}$$

Definizione 5. *Dato un metodo numerico* $\boldsymbol{y}_{n+1} = \boldsymbol{\mathcal{Y}}_n(\boldsymbol{f}, k, \boldsymbol{y}_0, \ldots, \boldsymbol{y}_n)$*, la* regione di assoluta stabilità *(o* linear stability domain*) è l'insieme dei numeri* $z = k\lambda$ *per cui la soluzione di* (19.1) *soddisfa* $\lim_{n\to\infty} \boldsymbol{y}_n = 0$.

Con riferimento al θ-metodo, la regione di assoluta stabilità del metodo di Eulero è $\{z \in \mathbb{C}\colon |1+z| < 1\}$, per Eulero implicito è $\{z \in \mathbb{C}\colon |1-z| > 1\}$ e per il metodo dei trapezi è $\{z \in \mathbb{C}\colon \Re(z) < 0\}$. Diremo che un metodo è *assolutamente stabile* (o *A-stabile*) se la sua regione di assoluta stabilità contiene $\mathbb{C}^- = \{z \in \mathbb{C}\colon \Re(z) < 0\}$, cioè se riproduce correttamente il comportamento della soluzione analitica di (19.1) quando $\Re(\lambda) < 0$. Per inciso, se λ è puramente immaginario $\lambda = \delta\mathrm{i}$, $|y_n| \to +\infty$ per Eulero, $|y_n| \to 0$ per Eulero implicito e $|y_n| = 1 = |y(t_n)|$ per il metodo dei trapezi.

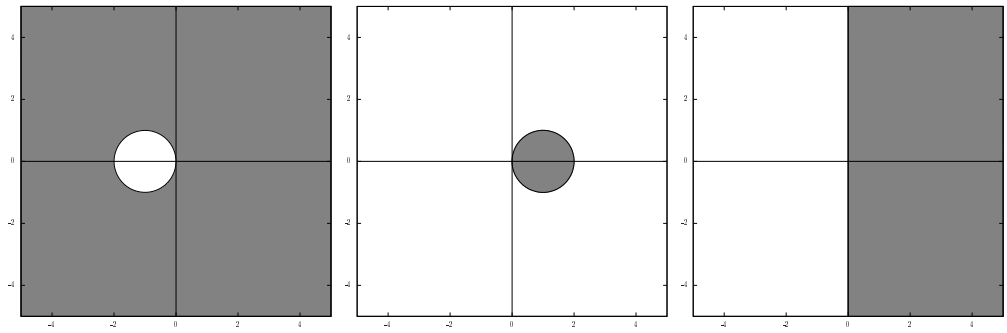


Figura 19.1: Regioni di assoluta stabilità (bianche) per i metodi di Eulero, Eulero implicito e dei trapezi.

19.1 A-stabilità dei metodi di Runge–Kutta espliciti

Per analizzare l'A-stabilità dei metodi Runge–Kutta espliciti, bisogna esprimere la soluzione al problema (19.1) come

$$y_n = r(k\lambda)^n y_0$$

e studiare la funzione $r(k\lambda)$.

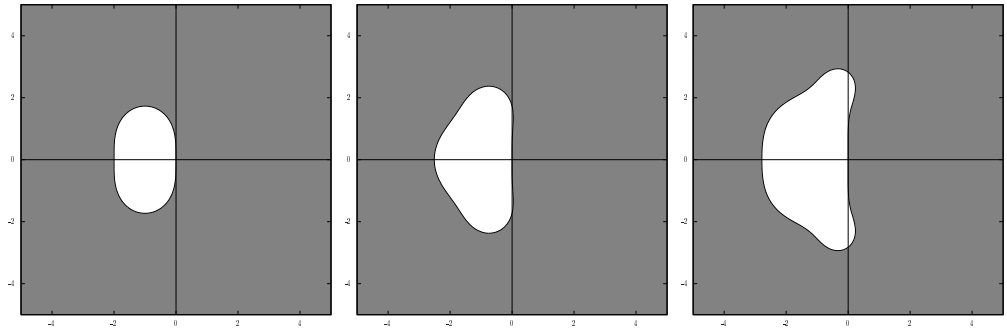


Figura 19.2: Regioni di assoluta stabilità (bianche) per i metodi di Runge–Kutta di ordine 2, 3 e 4.

Teorema 12. *Per un metodo Runge–Kutta esplicito di ordine p uguale al numero di stadi ν, si ha*

$$r(z) = 1 + z + \frac{z^2}{2!} + \ldots + \frac{z^\nu}{\nu!}$$

Dimostrazione. Iniziamo col dimostrare che $r(z)$ è un polinomio di grado ν. Si ha ξ_1 polinomio di grado 0. Supponiamo che ξ_j sia un polinomio di grado $j-1$ in $z = k\lambda$ per $j = 2, 3, \ldots, \nu - 1$: allora

$$\xi_\nu = y_n + k \sum_{j=1}^{\nu-1} a_{\nu,j} \lambda \xi_j = y_n + k\lambda \sum_{j=1}^{\nu-1} a_{\nu,j} \xi_j$$

è un polinomio di grado ν in z. Quindi $y_{n+1} = r(k\lambda) y_n$ con $r(z)$ polinomio di grado ν in z. Poi, se l'ordine del metodo è p, significa che

$$y_1 - y(k) = r(k\lambda) y_0 - y(k) = \mathcal{O}(k^{p+1})$$

Ma $y(k) = \mathrm{e}^{k\lambda} y_0$. Quindi

$$r(z) = \left(1 + z + \frac{z^2}{2!} + \ldots + \frac{z^p}{p!}\right) + \mathcal{O}(k^{p+1})$$

da cui la tesi. □

In ogni caso, la dimostrazione qui sopra mostra che per un metodo Runge–Kutta esplicito $r(z)$ è un polinomio e $r(0) = 1$. Dunque, i metodi di Runge–Kutta di ordine p uguale al numero di stadi ν hanno tutti la stessa regione di stabilità. Si può allora facilmente dimostrare il seguente

Teorema 13. *Nessun metodo Runge–Kutta esplicito è A-stabile.*

Dimostrazione. Si ha

$$\lim_{n\to\infty} y_n = 0 \Leftrightarrow |r(k\lambda)| < 1$$

ma $r(0) = 1$ e dunque $r(z)$ non è un polinomio costante $r(z) \equiv c \in (-1, 1)$. Se non è così, certamente esiste k tale che $|r(k\lambda)| > 1$ e dunque la regione di assoluta stabilità non contiene $\mathbb{C}^-$. □

19.2 A-stabilità dei metodi lineari multistep

Ci limitiamo a riportare alcuni risultati.

Teorema 14. *Nessun metodo esplicito multistep è A-stabile.*

Teorema 15. *I metodi BDF ad un passo (Eulero implicito) e a due passi sono A-stabili.*

Teorema 16 (Seconda barriera di Dahlquist)**.** *L'ordine più alto che un metodo multistep A-stabile può raggiungere è due.*

19.3 Equazioni stiff

Se consideriamo il problema

$$\begin{cases} y'(t) = -100y(t), & t > 0 \\ y(0) = 1 \end{cases}$$

la condizione (19.2) per il metodo di Eulero impone $k < 1/50 = 0.02$. D'altra parte, la soluzione analitica del problema per $t^* = 0.4$ è minore di 10^{-17} (e dunque, trascurabile). Dunque, con poco più di 20 passi il metodo di Eulero arriva a calcolare la soluzione sino a t^*. Qual è dunque il problema? Eccolo:

$$\begin{cases} \boldsymbol{y}'(t) = \begin{bmatrix} -100 & 0 \\ 0 & -1 \end{bmatrix} \boldsymbol{y}(t), & t > 0 \\ \boldsymbol{y}(0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{cases}$$

La soluzione analitica è

$$\boldsymbol{y}(t) = \begin{bmatrix} \mathrm{e}^{-100t} \\ \mathrm{e}^{-t} \end{bmatrix}$$

e la sua norma infinito è minore di 10^{-17} per $t^* = 40$. Poiché però per poter calcolare la prima componente serve un passo temporale $k < 0.02$, sono necessari più di 2000 passi, anche se la prima componente diventa trascurabile dopo pochi passi e la seconda non richiederebbe un così elevato numero di passi. Dunque, anche se il metodo è convergente e il passo, per esempio, $k = 0.1$ garantisce un errore locale proporzionale a $k^2 = 0.01$, il metodo di Eulero non può essere usato con tale passo. Usando il metodo di Eulero implicito sarebbe possibile invece usare un passo piccolo all'inizio e poi, quando ormai la prima componente è trascurabile, si potrebbe incrementare il passo, senza pericolo di esplosione della soluzione. Per questo semplice problema, sarebbe possibile calcolare le due componenti separatamente. Nel caso generale, però, il sistema non è disaccoppiato, ma ci si può sempre ridurre, eventualmente in maniera approssimata, ad uno disaccoppiato e ragionare per componenti. Infatti, se A è una matrice diagonalizzabile,

$$\boldsymbol{y}'(t) = A\boldsymbol{y}(t) \Leftrightarrow \boldsymbol{z}'(t) = D\boldsymbol{z}(t) \Leftrightarrow \boldsymbol{z}(t) = \exp(tD)\boldsymbol{z}_0$$

ove $AV = VD$, $D = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$, e $\boldsymbol{y}(t) = V\boldsymbol{z}(t)$. Poi

$$\boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{b} \Leftrightarrow \boldsymbol{z}'(t) = D\boldsymbol{z}(t) + V^{-1}\boldsymbol{b} \Leftrightarrow$$
$$\Leftrightarrow \boldsymbol{z}(t) = \boldsymbol{z}_0 + t\varphi_1(tD)(D\boldsymbol{z}_0 + V^{-1}\boldsymbol{b})$$

ove

$$\varphi_1(\lambda) = \begin{cases} \dfrac{\mathrm{e}^{\lambda} - 1}{\lambda} & \text{se } \lambda \neq 0 \\ 1 & \text{se } \lambda = 0 \end{cases}$$

Infine

$$\boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)) \Leftrightarrow \boldsymbol{y}'(t) \approx \boldsymbol{f}(t_n, \boldsymbol{y}_n) + J_n(\boldsymbol{y}(t) - \boldsymbol{y}_n)$$

ove J_n è la matrice jacobiana

$$J_n = \frac{\partial f_i}{\partial y_j}(t_n, \boldsymbol{y}_n)$$

e, se J_n è diagonalizzabile, ci si riconduce al caso precedente. Dunque, si ha sempre a che fare con gli autovalori di J_n (nel caso J_n non sia diagonalizzabile, si ragiona in maniera equivalente con blocchi di Jordan).

Definizione 6. *Un sistema di ODEs* (15.1) *si dice* stiff *in un intorno di* t_n *se esiste almeno una coppia di autovalori* λ_1, λ_2 *della matrice jacobiana* J_n *tali che*

- $\Re(\lambda_1) < 0$, $\Re(\lambda_2) < 0$
- $\Re(\lambda_1) \ll \Re(\lambda_2)$

In pratica, può essere molto difficile capire se un sistema non lineare presenta regioni di *stiffness* o meno. Altrettanto difficile è rispondere alla domanda: per un problema stiff, conviene usare un metodo esplicito con passo piccolo o un metodo implicito? È chiaro che il metodo esplicito è di facile implementazione e applicazione, ma richiede molti passi temporali. Il metodo implicito richiede la soluzione ad ognuno dei "pochi" passi di un sistema, in generale, non lineare.

Capitolo 20

Integratori esponenziali

I problemi di assoluta stabilità per semplici problemi lineari visti nel capitolo precedente, portano alla ricerca di nuovi metodi. Consideriamo il sistema differenziale

$$\begin{cases} \boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{b}, & t > 0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

La soluzione analitica è

$$\boldsymbol{y}(t) = \exp((t-t_0)A)\boldsymbol{y}_0 + (t-t_0)\varphi_1((t-t_0)A)\boldsymbol{b} = \boldsymbol{y}_0 + t\varphi_1((t-t_0)A)(A\boldsymbol{y}_0 + \boldsymbol{b})$$

Infatti $\boldsymbol{y}(t_0) = \boldsymbol{y}_0$ e

$$\begin{aligned} \boldsymbol{y}'(t) &= A\exp((t-t_0)A)\boldsymbol{y}_0 + \exp((t-t_0)A)\boldsymbol{b} = \\ &= A(\exp((t-t_0)A)\boldsymbol{y}_0 + (t-t_0)((t-t_0)A)^{-1}\exp((t-t_0)A)\boldsymbol{b} + \\ &- (t-t_0)((t-t_0)A)^{-1}\boldsymbol{b} + A^{-1}\boldsymbol{b}) = \\ &= A(\exp((t-t_0)A)\boldsymbol{y}_0 + (t-t_0)\varphi_1((t-t_0)A)\boldsymbol{b}) + \boldsymbol{b} = A\boldsymbol{y}(t) + \boldsymbol{b} \end{aligned}$$

Le funzioni exp e φ_1 di matrice possono essere approssimate come visto al paragrafo 7. Da questa osservazione, per un problema

$$\begin{cases} \boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{b}(t, \boldsymbol{y}(t)), & t > 0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

il metodo *Eulero esponenziale* è

$$\boldsymbol{y}_{n+1} = \exp(kA)\boldsymbol{y}_n + k\varphi_1(kA)\boldsymbol{b}(t_n, \boldsymbol{y}_n)$$

Proposizione 4. *Il metodo di Eulero esponenziale è* esatto *se* $\boldsymbol{b}(\boldsymbol{y}(t)) = \boldsymbol{b}(\boldsymbol{y}_0) \equiv \boldsymbol{b}$ *e di ordine uno altrimenti.*

Dimostrazione. Si ha

$$\boldsymbol{y}_{n+1} = \exp(kA)\boldsymbol{y}_n + \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)\boldsymbol{g}(t_n)\mathrm{d}\tau$$

ove si è posto $\boldsymbol{g}(t) = \boldsymbol{b}(t, \boldsymbol{y}(t))$. Per la formula di variazioni delle costanti (7.2)

$$\begin{aligned}
&\boldsymbol{y}(t_{n+1}) - \exp(kA)\boldsymbol{y}(t_n) - \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)\boldsymbol{g}(t_n)\mathrm{d}\tau = \\
&\quad = \exp(kA)\boldsymbol{y}(t_n) + \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)\boldsymbol{g}(\tau)\mathrm{d}\tau + \\
&\quad - \exp(kA)\boldsymbol{y}(t_n) - \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)\boldsymbol{g}(t_n)\mathrm{d}\tau = \\
&\quad = \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)(\boldsymbol{g}(t_n) + \boldsymbol{g}'(\tau_n)(\tau - t_n) - \boldsymbol{g}(t_n))\mathrm{d}\tau = \\
&\quad = k^2\varphi_2(kA)\boldsymbol{g}'(\tau_n) = \mathcal{O}(k^2)
\end{aligned}$$

□

Si può inoltre dimostrare che il metodo converge (cioè è stabile). Poiché risolve esattamente i problemi lineari, il metodo è A-stabile.

Proposizione 5. *Per un problema* lineare, non autonomo

$$\begin{cases} \boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{b}(t), \quad t > 0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

il metodo esponenziale—punto medio

$$\boldsymbol{y}_{n+1} = \exp(kA)\boldsymbol{y}_n + k\varphi_1(kA)\boldsymbol{b}(t_n + k/2)$$

è esatto se $\boldsymbol{b}(t) \equiv \boldsymbol{b}$ *e di ordine 2 altrimenti.*

Dimostrazione. Procedendo come sopra, si arriva a

$$\begin{aligned}
&\boldsymbol{y}(t_{n+1}) - \exp(kA)\boldsymbol{y}(t_n) - \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)\boldsymbol{b}(t_n + k/2)\mathrm{d}\tau = \\
&\quad = \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)\boldsymbol{b}'(\tau_n + k/2)(\tau - (t_n + k/2))\mathrm{d}\tau = \\
&\quad = \int_{t_n}^{t_{n+1}} \exp((t_{n+1}-\tau)A)\boldsymbol{b}'(\tau_n + k/2)(\tau - t_n - k/2)\mathrm{d}\tau = \\
&\quad = (k^2\varphi_2(kA) - k^2/2\varphi_1(kA))\boldsymbol{b}'(\tau_n + k/2) = \\
&\quad = \left(\frac{k^2 I}{2} + \frac{k^3 A}{6} + \mathcal{O}(k^4) - \frac{k^2 I}{2} - \frac{k^3 A}{2} + \mathcal{O}(k^4)\right)\boldsymbol{b}'(\tau_n + k/2) = \\
&\quad = \mathcal{O}(k^3)
\end{aligned}$$

□

Anche in questo caso si può dimostrare che il metodo converge e che è A-stabile. Dato un problema differenziale in forma autonoma

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(\boldsymbol{y}(t)), \quad t > t_0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

si può pensare di linearizzarlo ad ogni passo

$$\boldsymbol{y}'(t) = J_n \boldsymbol{y}(t) + \boldsymbol{b}_n(\boldsymbol{y}(t))$$

ove

$$J_n = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}(\boldsymbol{y}_n), \quad \boldsymbol{b}_n(\boldsymbol{y}(t)) = \boldsymbol{f}(\boldsymbol{y}(t)) - J_n \boldsymbol{y}(t)$$

e applicarvi il metodo di Eulero esponenziale. Si arriva così al metodo di Eulero–Rosenbrock esponenziale

$$\boldsymbol{y}_{n+1} = \exp(kJ_n)\boldsymbol{y}_n + k\varphi_1(kJ_n)\boldsymbol{b}_n(\boldsymbol{y}_n) = \boldsymbol{y}_n + k\varphi_1(kJ_n)\boldsymbol{f}(\boldsymbol{y}_n)$$

Il metodo è di ordine 2 e convergente.

Capitolo 21

Esercizi

1. Si consideri il seguente problema differenziale del secondo ordine *ai limiti*
$$\begin{cases} u''(x) - 3\cos(u(x)) = 0, \quad x \in (0,1) \\ u(0) = 0 \\ u(1) = 1 \end{cases}$$
Lo si trasformi in un sistema del primo ordine ($t = x$, $y_1(t) = u(x)$, $y_2(t) = u'(x)$) da risolvere con il metodo di Eulero esplicito e si determini, con una opportuna strategia, quale dovrebbe essere il valore iniziale $y_2(0)$ affinché $y_1(t) = u(x)$ sia soluzione del problema originale.

2. Con riferimento alla Figura 21.1, l'equazione del pendolo è
$$\begin{cases} l\vartheta''(t) = -g\sin\vartheta(t) \\ \vartheta(0) = \vartheta_0 \\ \vartheta'(0) = 0 \end{cases}$$
La si risolva con il metodo dei trapezi fino al tempo $t^* = \pi\sqrt{l/g}$ (assumendo $l = 1$, $\vartheta_0 = \pi/4$). Si confronti la traiettoria con quella del pendolo *linearizzato* ($\sin\vartheta(t) \approx \vartheta(t)$). Di quest'ultimo, si trovi il numero minimo di passi temporali affinché il metodo di Eulero esplicito produca una soluzione al tempo t^* che dista da $\vartheta(t^*)$ meno di 10^{-2}.

3. Si calcoli $\boldsymbol{y}(1)$, ove $\boldsymbol{y}'(t) = A\boldsymbol{y}(t)$, $\boldsymbol{y}(0) = [1,\ldots,1]^{\mathrm{T}}$, con A data da `A = 100*toeplitz(sparse([1,1],[1,2],[-2,1],1,10))`, usando il θ-metodo con $\theta = 0, 1/2, 1$ e diversi passi temporali $k = 2^{-3}, 2^{-4}, \ldots, 2^{-8}$. Si confrontino i risultati con la *soluzione di riferimento* ottenuta usando $\theta = 1/2$ e $k = 2^{-10}$, mettendo in evidenza l'ordine del metodo usato. Si provi anche il valore $\theta = 2/3$, discutendo i risultati ottenuti.

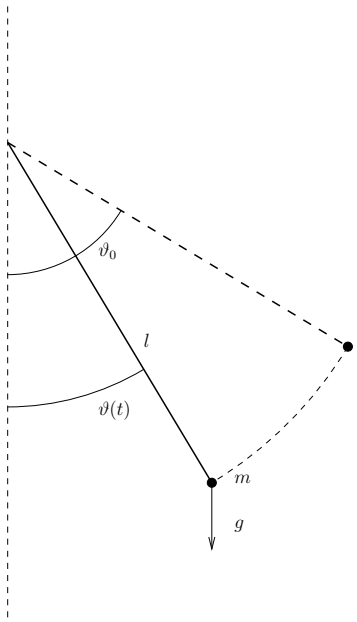


Figura 21.1: Pendolo

4. Si risolva il sistema di ODEs

$$\begin{cases} A'(t) = -2a(t)A(t) \\ a'(t) = A(t)^2 + \Omega(t)^2 - a(t)^2 - 1 \\ \Omega'(t) = -2(a(t) + A(t))\Omega(t) \end{cases} \tag{21.1}$$

con dato iniziale

$$\begin{cases} A(0) = 0.5 \\ a(0) = 2 \\ \Omega(0) = 10 \end{cases}$$

con il metodo di Eulero implicito fino ad un tempo finale $t^* = 15$, producendo un grafico della quantità $E(t) = (A(t)^2 + a(t)^2 + \Omega(t)^2 + 1)/(2A(t))$. Si confrontino le soluzioni ottenute usando 300 o 900 timesteps.

5. Si implementi il metodo di Eulero modificato (secondo tableau in Tabella 18.2) e lo si testi per il sistema differenziale (21.1), producendo il grafico della quantità $E(t)$.

6. Si implementino gli altri due metodi di ordine 2 in Tabella 18.2, li si testi per il sistema differenziale (21.1), mettendone in evidenza l'ordine.

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Tabella 21.1: Metodo di Runge–Kutta a 3 stadi.

7. Si implementi il metodo Runge–Kutta di tableau in Tabella 21.1, determinandone numericamente l'ordine.

8. Si implementi la function relativa ad un generico metodo di Runge–Kutta con tableaux dato da

$\boldsymbol{c}$	A
	$\boldsymbol{b}^{\mathrm{T}}$

   ove $\boldsymbol{c}$, A e $\boldsymbol{b}$ sono dati.

9. Si implementi il metodo Runge–Kutta (embedded) di tableau

0			
$\frac{1}{2}$	$\frac{1}{2}$		
1	-1	2	
	0	1	
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

   e lo si applichi al problema differenziale 21.1.

10. Si implementi il metodo Runge–Kutta–Fehlberg il cui tableau è riportato nella Tabella 18.7, e se ne individui numericamente l'ordine. Lo si testi sul sistema differenziale (21.1).

Parte 3

PDEs (Equazioni alle derivate parziali)

Capitolo 22

Equazione del calore

22.1 Equazione del calore con dati iniziali e condizioni ai limiti

Consideriamo la seguente equazione alle derivate parziali

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) = \dfrac{\partial^2 u}{\partial x^2}(t,x), & t>0,\ x\in(0,L) \\ u(0,t)=u(L,t)=0, & t>0 \text{ \textit{(condizioni ai bordi)}} \\ u(0,x)=u_0(x), & x\in(0,L) \text{ \textit{(condizioni iniziali)}} \end{cases} \tag{22.1}$$

Supponiamo che $u_0(x)$ verifichi le *condizioni di compatibilità* $u_0(a)=u_0(b)=0$. Tale equazione rappresenta, per esempio, l'andamento della temperatura u su una barra di lunghezza L, i cui estremi sono tenuti a temperatura zero, e con una distribuzione iniziale di temperatura $u_0(x)$.

22.1.1 Esistenza di una soluzione

Cerchiamo una soluzione *a variabili separabili*

$$u(t,x)=\psi(t)\phi(x)$$

Inserendo tale rappresentazione in (22.1), si deduce

$$\psi'(t)\phi(x)=\psi(t)\phi''(x), \quad t>0,\ x\in(0,L)$$

da cui

$$\frac{\psi'(t)}{\psi(t)} = -K \text{ (costante)} \Rightarrow \psi(t)=A\mathrm{e}^{-Kt}$$

Per quanto riguarda $\phi(x)$, la soluzione generale è

$$\phi(x) = c_1 \mathrm{e}^{\sqrt{-K}x} + c_2 \mathrm{e}^{-\sqrt{-K}x}$$

Imponendo le condizioni al bordo

$$\begin{aligned} 0 &= \phi(0) = c_1 + c_2 \\ 0 &= \phi(L) = c_1 \mathrm{e}^{\sqrt{-K}L} + c_2 \mathrm{e}^{-\sqrt{-K}L} = c_1 \left(\mathrm{e}^{\sqrt{-K}L} - \mathrm{e}^{-\sqrt{-K}L} \right) \end{aligned}$$

Se $K \leq 0$, allora $\mathrm{e}^{\sqrt{-K}L} + \mathrm{e}^{-\sqrt{-K}L} > 0$ e dunque $c_1 = 0$ (e anche c_2). Quindi $\phi(x) = 0$, ma in tal caso $\psi(0)\phi(x) \neq u_0(x)$. Se invece $K = \lambda^2 > 0$, $\lambda > 0$, allora

$$\phi(x) = c_1 \left(\mathrm{e}^{\mathrm{i}\lambda x} - \mathrm{e}^{-\mathrm{i}\lambda x} \right) = 2c_1 \mathrm{i} \sin(\lambda x) = B \sin(\lambda x)$$

e poiché $\phi(L) = 0$, l'unica possibilità non banale è $\lambda = j\pi/L$, j numero naturale non nullo. Pertanto, la funzione

$$u_j(t,x) = \exp\left(-\frac{j^2\pi^2}{L^2} t \right) \sin\left(\frac{j\pi}{L} x \right)$$

è soluzione dell'equazione del calore (e soddisfa le condizioni ai bordi) per ogni j. Quindi, la seguente combinazione lineare infinita

$$u(t,x) = \sum_{j=1}^{\infty} c_j u_j(t,x)$$

è soluzione *formale* dell'equazione del calore. Per quanto riguarda la condizione iniziale, si deve imporre

$$u_0(x) = u(0,x) = \sum_{j=1}^{\infty} c_j \sin\left(\frac{j\pi}{L} x \right) \tag{22.2}$$

Poiché $u_0(x)$ è nulla agli estremi, la possiamo prolungare per *antisimmetria* all'intervallo $[-L, L]$ e poi per periodicità a $\mathbb{R}$. Sotto opportune ipotesi, la sua serie di Fourier

$$\bar{u}_0(x) = \sum_{j=-\infty}^{+\infty} u_{0j} \phi_j(x)$$

converge. Poiché $\bar{u}_0(x)$ è dispari, con riferimento al paragrafo 13.2.1,

$$\begin{aligned} u_{0m/2+1+j} &= \int_{-L}^{L} \bar{u}_0(x) \overline{\phi_{m/2+1+j}(x)} \mathrm{d}x = \frac{-\mathrm{i}}{\sqrt{2L}} \int_{-L}^{L} \bar{u}_0(x) \sin\left(\frac{2\pi j(x+L)}{2L} \right) \mathrm{d}x = \\ &= \frac{-\mathrm{i}\sqrt{2}}{\sqrt{L}} \int_0^L \bar{u}_0(x) \sin\left(\frac{j\pi x}{L} + j\pi \right) \mathrm{d}x = \\ &= \frac{-\mathrm{i}\sqrt{2}}{\sqrt{L}} \int_0^L u_0(x) \sin\left(\frac{j\pi x}{L} + j\pi \right) \mathrm{d}x \end{aligned}$$

e

$$u_{0m/2+1-j} = \int_{-L}^{L} \bar{u}_0(x)\overline{\phi_{m/2+1-j}(x)}\mathrm{d}x = \frac{-\mathrm{i}}{\sqrt{2L}}\int_{-L}^{L} \bar{u}_0(x)\sin\left(\frac{-2\pi j(x+L)}{2L}\right)\mathrm{d}x =$$
$$= \frac{\mathrm{i}\sqrt{2}}{\sqrt{L}}\int_0^L \bar{u}_0(x)\sin(j\pi x/L + j\pi)\mathrm{d}x = -u_{0m/2+1+j}$$

da cui

$$\begin{aligned}
\bar{u}_0(x) &= \sum_{j=-\infty}^{+\infty} u_{0m/2+1+j}\phi_{m/2+1+j}(x) = \\
&= \sum_{j=-\infty}^{+\infty} u_{0m/2+1+j}\frac{\cos(j\pi x/L + j\pi) + \mathrm{i}\sin(j\pi x/L + j\pi)}{\sqrt{2L}} = \\
&= \sum_{j=-\infty}^{-1} u_{0m/2+1+j}\frac{\cos(j\pi x/L + j\pi) + \mathrm{i}\sin(j\pi x/L + j\pi)}{\sqrt{2L}} + \\
&+ \sum_{j=1}^{+\infty} u_{0m/2+1+j}\frac{\cos(j\pi x/L + j\pi) + \mathrm{i}\sin(j\pi x/L + j\pi)}{\sqrt{2L}} = \\
&= \sum_{j=1}^{+\infty} -u_{0m/2+1+j}\frac{\cos(j\pi x/L + j\pi) - \mathrm{i}\sin(j\pi x/L + j\pi)}{\sqrt{2L}} + \\
&+ \sum_{j=1}^{+\infty} u_{0m/2+1+j}\frac{\cos(j\pi x/L + j\pi) + \mathrm{i}\sin(j\pi x/L + j\pi)}{\sqrt{2L}} = \\
&= \sum_{j=1}^{\infty} u_{0m/2+1+j}\frac{\sqrt{2}}{\sqrt{L}}\mathrm{i}\sin(j\pi x/L + j\pi) = \\
&= \sum_{j=1}^{\infty}\left[\frac{2}{L}\int_0^L u_0(x)\sin\left(\frac{j\pi}{L}x\right)\mathrm{d}x\right]\sin\left(\frac{j\pi}{L}x\right)
\end{aligned}$$

Confrontanto quest'ultima espressione con (22.2), si deduce

$$c_j = \left[\frac{2}{L}\int_0^L u_0(x)\sin\left(\frac{j\pi}{L}x\right)\mathrm{d}x\right]$$

Si potrebbe mostrare adesso che

$$u(t,x) = \sum_{j=1}^{\infty} c_j \exp\left(-\frac{j^2\pi^2}{L^2}t\right)\sin\left(\frac{j\pi}{L}x\right)$$

è soluzione di (22.1) (bisogna derivare sotto il segno di serie). Dalla presenza del termine esponenziale negativo nel tempo per ogni componente $u_j(t,x)$, si deduce ogni componente tende a zero per $t \to +\infty$ (e dunque anche la soluzione), ma con *diverse* velocità dipendenti da un fattore proporzionale a j^2. L'equazione del calore rappresenta il modello dei fenomeni di *diffusione*. La diffusione è il processo mediante il quale la materia (o l'energia) è trasportata da una parte di un sistema ad un'altra come risultato di moti molecolari random.

22.1.2 Unicità della soluzione

Introduciamo la seguente quantità *(energia)*

$$E(t) = \int_0^L \frac{1}{2}u^2(t,x)\mathrm{d}x$$

Si ha

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \int_0^L \frac{\partial}{\partial t}\left[\frac{1}{2}u^2(t,x)\right]\mathrm{d}x = \int_0^L u\frac{\partial u}{\partial t}\mathrm{d}x = \int_0^L u\frac{\partial^2 u}{\partial x^2}\mathrm{d}x$$

Integrando per parti e tenendo conto delle condizioni ai bordi, si ha

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\int_0^L \left(\frac{\partial u}{\partial x}\right)^2 \mathrm{d}x \leq 0$$

Per dimostrare l'unicità, consideriamo come al solito il problema omogeneo (corrispondente a (22.1) con $u_0 \equiv 0$. Per tale problema $E_0(0) = 0$ e quindi $0 \leq E_0(t) \leq E_0(0)$ da cui $E_0(t) = 0$ per ogni t. Quindi $u(t,x) \equiv 0$ è l'unica soluzione del problema omogeneo. Dunque, se $u_1(t,x)$ e $u_2(t,x)$ fossero due soluzioni del problema (22.1), allora $u_1(t,x) - u_2(t,x)$ sarebbe soluzione del problema omogeneo e quindi $u_1(t,x) \equiv u_2(t,x)$.

Se $u_0(x) \geq 0$, si può dimostrare *(principio del massimo debole)* che la soluzione rimane non negativa per ogni t (dall'interpretazione fisica, è ovvio). Infatti, dato $\varepsilon > 0$, si ponga $v(t,x) = u(t,x) - \varepsilon x^2$. Allora $\partial_t v - \partial_{xx} v = 2\varepsilon > 0$. Se il minimo di $v(t,x)$ stesse in $(\bar{t},\bar{x})$, $0 < \bar{t}$, $0 < \bar{x} < L$, allora $\partial_t v(\bar{t},\bar{x}) = 0$ (punto critico) e $\partial_{xx} v(\bar{t},\bar{x}) \geq 0$ (punto di minimo). Dunque

$$\partial_t v(\bar{t},\bar{x}) - \partial_{xx} v(\bar{t},\bar{x}) \leq 0$$

assurdo. Quindi, il punto di minimo per $v(t,x)$ sta in $\Gamma = \{0\} \times [0,L] \cup [0,+\infty) \times \{0,L\}$. Dunque

$$\min_\Gamma u - \varepsilon L^2 \leq \min_\Gamma v = \min v \leq \min u$$

e facendo tendere $\varepsilon \to 0$, si ottiene

$$\min_{\Gamma} u \leq \min u$$

Poichè ovviamente vale anche la disuguaglianza opposta,

$$\min u = \min_{\Gamma} u = \min\{\min u_0, 0\} = 0$$

22.2 Metodo delle linee

Il *metodo delle linee* per la risoluzione di problemi del tipo

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) = \dfrac{\partial^2 u}{\partial x^2}(t,x) + f(u(t,x)) + g(t,x), \quad t > 0,\ x \in (a,b) \\ + \textit{ condizioni ai bordi} \\ + \textit{ condizioni iniziali} \end{cases} \tag{22.3}$$

ove il termine $f(u(t,x))$ si chiama *reazione* e il termine $g(t,x)$ *sorgente*, prevede di discretizzare gli operatori differenziali spaziali con uno dei metodi visti per i problemi con valori ai bordi e poi risolvere il sistema di ODEs che ne risulta con un metodo per problemi ai valori iniziali visti. Vediamo qualche esempio.

22.2.1 Differenze finite

Trascurando per il momento le condizioni ai bordi e usando differenze finite centrate del secondo ordine a passo costante h

$$\begin{bmatrix} y_1'(t) \\ y_2'(t) \\ \vdots \\ y_{m-1}'(t) \\ y_m'(t) \end{bmatrix} = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_{m-1}(t) \\ y_m(t) \end{bmatrix} + \begin{bmatrix} f(y_1(t)) \\ f(y_2(t)) \\ \vdots \\ f(y_{m-1}(t)) \\ f(y_m(t)) \end{bmatrix} + \begin{bmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_{m-1}(t) \\ g_m(t) \end{bmatrix}$$

ove $y_j(t) \approx y(t, x_j)$ o, in maniera compatta,

$$\boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{f}(\boldsymbol{y}(t)) + \boldsymbol{g}(t) \tag{22.4}$$

(con l'ovvia definizione dei simboli). A questo punto, si sceglie il metodo di integrazione temporale (θ-metodo, Runge–Kutta, multistep, esponenziale). Si tenga presente che il problema (22.4), che si dice *semidiscretizzato*, è solitamente un problema stiff. Infine, si sistemano le condizioni ai bordi.

Condizioni al bordo di Dirichlet

Vediamo come imporre una condizione di Dirichlet costante in $x_1 = a$. La prima equazione del sistema di ODEs riguarda $y_1(t) \approx y(t, x_1)$ e sarà della forma $y_1'(t) = \dots$ Se si vuole imporre una condizione di Dirichlet costante, è sufficiente modifcare opportunamente la prima del termine di destra del sistema di ODEs in modo da ottenere l'equazione $y_1'(t) = 0$.

Nel caso si volesse imporre una condizione di Dirichlet dipendente dal tempo, per esempio $y(t, a) = \alpha(t)$, si deve ancora modificare la prima riga del sistema differenziale, ma tale modifica dipende dal metodo scelto per l'integrazione temporale. Per esempio, avendo scelto il metodo di Eulero implicito, si ha

$$(I - kA)\boldsymbol{y}_{n+1} - k\boldsymbol{f}(\boldsymbol{y}_{n+1}) = \boldsymbol{y}_n + k\boldsymbol{g}(t_{n+1})$$

ove k è il passo di integrazione temporale, cioè si deve risolvere il sistema non lineare in $\boldsymbol{x} = \boldsymbol{y}_{n+1}$

$$F_n(\boldsymbol{x}) = 0$$

A questo punto, si può modificare la prima riga di $F_n(\boldsymbol{y}_{n+1})$ con $x_1 - \alpha(t_{n+1})$.

Per il metodo di Eulero esponenziale, invece, si ha

$$\boldsymbol{y}_{n+1} = \exp(kA)\boldsymbol{y}_n + k\varphi_1(kA)(\boldsymbol{f}(\boldsymbol{y}_n) + \boldsymbol{g}(t_n)) = \exp(kA)\boldsymbol{y}_n + k\varphi_1(kA)\boldsymbol{b}$$

Se la prima riga di A viene messa a zero, la prima riga di $\exp(kA)$ e $\varphi_1(kA)$ è il primo vettore della base canonica e dunque basta porre il primo elemento di $\boldsymbol{b}$ uguale a $(\alpha(t_{n+1}) - \alpha(t_n))/k$.

Condizioni al bordo di Neumann

Per quanto riguarda una condizione di Neumann omogenea, per esempio in $x = b$, si può pensare di introdurre la variabile fittizia $y_{m+1}(t) \approx u(t, x_{m+1})$, $x_{m+1} = b + h$ e imporre che $y_{m+1}(t) = y_{m-1}(t)$. L'approssimazione da usare per $\frac{\partial^2 u}{\partial x^2}(t, b)$ diventa dunque

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}(t, b) &\approx \frac{u(t, x_{m+1}) - 2u(t, x_m) + u(t, x_{m-1})}{h^2} = \\ &= \frac{y_{m+1}(t) - 2y_m(t) + y_{m-1}(t)}{h^2} = \frac{2y_{m-1}(t) - 2y_m(t)}{h^2}\end{aligned}$$

In maniera analoga si possono trattare condizioni di Neumann non omogenee (vedi paragrafo 10.5).

Equazione di trasporto-diffusione

Consideriamo l'equazione del *trasporto*

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) + c\dfrac{\partial u}{\partial x}(t,x) = 0, \quad t > 0,\ x \in \mathbb{R} \\ u(0,x) = u_0(x) \end{cases} \tag{22.5}$$

È facile verificare che la soluzione analitica è $u(t,x) = u_0(x - ct)$, da cui il nome dell'equazione. Se consideriamo, più in generale, l'equazione di *trasporto-diffusione*

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) + c\dfrac{\partial u}{\partial x}(t,x) = d\dfrac{\partial^2 u}{\partial x^2} \quad t > 0,\ x \in (0,L) \\ u(0,x) = u_0(x) \\ u(0,0) = u(0,L) = 0 \end{cases}$$

ove $d > 0$, è lecito aspettarsi che entrambi i fenomeni di diffusione e trasporto si manifestino. Ancora, se $u_0(x) \geq 0$, tale rimane la soluzione per ogni t. Ma ciò è vero dopo aver discretizzato con il metodo delle linee? Abbiamo i due risultati seguenti.

Teorema 17. *Dato*

$$\begin{cases} \boldsymbol{y}'(t) = \boldsymbol{f}(t, \boldsymbol{y}(t)), \quad t > 0 \\ \boldsymbol{y}(t_0) = \boldsymbol{y}_0 \end{cases}$$

sono equivalenti le seguenti proprietà:

- *se* $\boldsymbol{y}_0 \geq 0$, *allora* $\boldsymbol{y}(t) \geq 0$ *per ogni* t *(il sistema si dice* positivo*)*
- *dato* $\boldsymbol{x}$, *con* $x_i = 0$, $x_j \geq 0$, $j \neq i$, *allora* $f_i(t, \boldsymbol{x}) \geq 0$

Da questo teorema segue, come corollario, il seguente, che può però essere dimostrato in maniera diretta.

Teorema 18. *Un sistema lineare* $\boldsymbol{y}'(t) = A\boldsymbol{y}(t)$ *è positivo se e solo se*

$$a_{ij} \geq 0 \text{ per ogni } j \neq i$$

ove $A = (a_{ij})$.

Dimostrazione. Supponiamo che il sistema sia positivo. Allora, se $\boldsymbol{y}_0 \geq 0$, si ha $\boldsymbol{y}(\tau) \geq 0$. Ma

$$\boldsymbol{y}(\tau) = \exp(\tau A)\boldsymbol{y}_0 = (I + \tau A)\boldsymbol{y}_0 + \mathcal{O}(\tau^2)$$

se τ è sufficientemente piccolo. Se $a_{\bar{i}\bar{j}} < 0$, $\bar{j} \neq \bar{i}$, allora

$$(I + \tau A)\boldsymbol{e}_{\bar{j}} = \begin{bmatrix} * \\ \vdots \\ * \\ \tau a_{\bar{i}\bar{j}} \\ * \\ \vdots \\ * \end{bmatrix}$$

e dunque la componente $\bar{j}$-esima di $\boldsymbol{y}(\tau)$ sarebbe negativa, assurdo.

Se invece $a_{ij} \geq 0$, $j \neq i$, allora

$$\exp(t_n A) = \lim_{n\to\infty} \left(I + \frac{t_n}{n} A\right)^n \geq 0$$

da cui la positività. □

Tornando all'equazione (22.5), la discretizzazione mediante differenze finite centrate del secondo ordine porge, nei nodi interni,

$$y_i'(t) + c\frac{y_{i+1}(t) - y_{i-1}(t)}{2h} = d\frac{y_{i+1}(t) - 2y_i(t) + y_{i-1}(t)}{h^2}$$

I termini extradiagonali della matrice che ne deriva sono

$$-\frac{c}{2h} + \frac{d}{h^2} \text{ e } \frac{c}{2h} + \frac{d}{h^2}$$

Le condizioni per la positività sono

$$\begin{aligned} c > 0 \quad &\Rightarrow -\frac{c}{2h} + \frac{d}{h^2} \geq 0 \\ c < 0 \quad &\Rightarrow \frac{c}{2h} + \frac{d}{h^2} \geq 0 \end{aligned}$$

da cui

$$\frac{|c|h}{2d} \leq 1$$

La quantità $|c|h/(2d)$ si chiama *numero di Péclet di grilia.*

22.2.2 Elementi finiti

Nel caso di discretizzazione spaziale con elementi finiti lineari, la discretizzazione del problema (22.3) porta al sistema di ODEs

$$P\boldsymbol{y}'(t) = A\boldsymbol{y}(t) + \boldsymbol{f}(\boldsymbol{y}(t)) + \boldsymbol{g}(t) \tag{22.6}$$

ove A è (l'opposta de) la *stiffness matrix* e P la *mass matrix*, definita da,

$$\begin{aligned} p_{jj} &= \int_{x_{j-1}}^{x_{j+1}} \phi_j(x)\phi_j(x)\mathrm{d}x = \frac{h_{j-1}+h_j}{3} \\ p_{j\,j+1} = p_{j+1\,j} &= \int_{x_j}^{x_{j+1}} \phi_j(x)\phi_{j+1}(x)\mathrm{d}x = \frac{h_j}{6} \end{aligned} \tag{22.7a}$$

mentre, per $j = 1$ e $j = m$,

$$\begin{aligned} p_{11} &= \int_{x_1}^{x_2} \phi_1(x)\phi_1(x)\mathrm{d}x = \frac{h_1}{3} \\ p_{12} &= \int_{x_1}^{x_2} \phi_1(x)\phi_2(x)\mathrm{d}x = \frac{h_1}{6} \\ p_{m-1\,m} = p_{m\,m-1} &= \int_{x_{m-1}}^{x_m} \phi_m(x)\phi_{m-1}(x)\mathrm{d}x = \frac{h_{m-1}}{6} \\ p_{mm} &= \int_{x_{m-1}}^{x_m} \phi_m(x)\phi_m(x)\mathrm{d}x = \frac{h_{m-1}}{3} \end{aligned} \tag{22.7b}$$

Poi, per $1 < i < m$,

$$\begin{aligned} f_i &= \int_{x_{i-1}}^{x_{i+1}} f\left(\sum_{j=1}^{m} u_j\phi_j(x)\right)\phi_i(x)\mathrm{d}x = \\ &= \int_{x_{i-1}}^{x_i} f\left(\sum_{j=1}^{m} u_j\phi_j(x)\right)\phi_i(x)\mathrm{d}x + \int_{x_i}^{x_{i+1}} f\left(\sum_{j=1}^{m} u_j\phi_j(x)\right)\phi_i(x)\mathrm{d}x \approx \\ &\approx f(u_i)\frac{h_{i-1}}{2} + f(u_i)\frac{h_i}{2} = f(u_i)\frac{h_{i-1}+h_i}{2} \\ g_i &= \int_{x_{i-1}}^{x_{i+1}} g(t,x)\phi_i(x)\mathrm{d}x \approx g(t,x_i)\frac{h_{i-1}+h_i}{2} \end{aligned}$$

mentre per $i = 1$ e $i = m$

$$\begin{aligned} f_1 &= f(u_1)\frac{h_1}{2}, \qquad f_m = f(u_m)\frac{h_{m-1}}{2} \\ g_1 &= g(t,x_1)\frac{h_1}{2}, \quad g_m = g(t,x_m)\frac{h_{m-1}}{2} \end{aligned}$$

Le condizioni di Dirichlet omogenee per un nodo x_i si impongono sostituendo la riga corrispondente di P con zeri e 1 in diagonale e la riga corrispondente del termine di destra del sistema di ODEs con zeri.

Usando un metodo esplicito per la risoluzione del sistema differenziale (22.6), è necessaria l'inversione della matrice di massa. Per tale motivo, si può ricorrere alla tecnica del *mass lumping* che consiste nel rendere diagonale la matrice P sostituendo ogni sua riga con una riga di zeri e la somma degli elementi originali in diagonale. Tale modifica è equivalente all'approssimazione degli integrali in (22.7) mediante la formula dei trapezi e dunque non riduce l'accuratezza del metodo. Infatti, la matrice $P_{\mathrm{L}}^{(-1)}A$ (P_{L} la matrice di massa con lumping) risulta uguale alla matrice che si ottiene discretizzando con differenze finite centrate del secondo ordine.

Usando invece un metodo implicito per la risoluzione del sistema differenziale (22.6), non è necessaria la tecnica del mass lumping: semplicemente, si devono risolvere sistemi lineari in cui la matrice identità è sostituita dalla matrice di massa.

22.3 Esercizi

1. Si calcoli la soluzione analitica dell'equazione del calore con sorgente

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) = \dfrac{\partial^2 u}{\partial x^2}(t,x) + 2\mathrm{e}^t \sin(x), & t > 0,\ x \in (0,\pi/2) \\ u(t,0) = 0, & t > 0 \\ \dfrac{\partial u}{\partial x}(t,\pi/2) = 0, & t > 0 \\ u(0,x) = \sin(x), & x \in (0,\pi/2) \end{cases}$$

   usando differenze finite del secondo ordine nello spazio e Eulero implicito nel tempo. Si mostrino gli ordini spaziali e temporali della convergenza alla soluzione analitica al tempo $t^* = 1$.

2. Per l'esercizio sopra, discretizzato nello spazio tramite differenze finite centrate del secondo ordine con $m = 100$ nodi, si determini il numero minimo di passi temporali per avere un errore al tempo $t^* = 1$ rispetto alla soluzione analitica inferiore a 10^{-3}, avendo usato nel tempo

   - il metodo di Eulero
   - il metodo di Eulero implicito
   - il metodo dei trapezi

- il metodo di Heun
- il metodo Runge–Kutta di tableau in Tabella 21.1

3. Si ripeta l'esercizio 1. usando Eulero esponenziale e esponenziale—punto medio nel tempo.