Riassunto - Calcolo numerico - Argomenti rilevanti

Calcolo numerico (Università degli Studi di Trento)

Fattorizzazione LU

A			b
$\begin{bmatrix} 0 & 2 & 0 \\ 3 & 7 & 0 \\ 1 & 1 & -1 \end{bmatrix}$			$\begin{bmatrix} 2 \\ 7 \\ -1 \end{bmatrix}$

Table 1: Valori dall'esercizio 1 della prova di esame del 26 agosto 2014.

Si calcoli la fattorizzazione LU (con pivoting) della matrice A e, utilizzandola, si risolva il sistema lineare $Ax = b$.

Fattorizzazione di Gauss

Supponiamo che esistano due matrici L e U, rispettivamente triangolare inferiore e triangolare superiore, tali che $A = LU$.

Risolvere $Ax = LUx = b$ equivale a risolvere $Ly = b$ e $Ux = y$. Tali sistemi sono semplici in quanto triangolari, infatti

$$y_1 = \frac{b_1}{l_{11}} \qquad y_k = \frac{b_k - \sum_{j=1}^{k-1} l_{kj} y_j}{l_{kk}} \quad k = 2, \dots, n$$

$$x_n = \frac{y_n}{u_{nn}} \qquad x_k = \frac{y_k - \sum_{j=k+1}^{n} u_{kj} x_j}{u_{kk}} \quad k = n-1, \dots, 1$$

La matrice U si calcola con il metodo dell'eliminazione di Gauss (MEG), il cui k-esimo passo corrisponde al prodotto $M_k A^{(k)}$, dove $A^{(k)} = M_{k-1} \dots M_1 A$ e M_k e' una matrice identica la cui k-esima colonna ha, per le righe $i > k$, i valori $-l_{ik} = -a_{ik}^{(k)} / a_{kk}^{(k)}$.

Si puo' verificare che, ad ogni passo k, gli elementi $a_{ik}^{(k)}$ con $i > k$ si annullano e al passo $k = n - 1$ si ottiene una matrice $U = M_{n-1} \dots M_1 A = MA$ triangolare superiore.

$$U = MA = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -l_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -l_{n1} & \cdots & -l_{n(n-1)} & 1 \end{bmatrix} A \qquad \text{con } l_{ij} = a_{ij}/a_{jj}$$

Considerato $A^{(k+1)} = M_k A^{(k)}$, per tornare alla matrice $A^{(k)}$ da $A^{(k+1)}$, e' necessario operare $M_k^{-1} A^{(k+1)}$ ovvero moltiplicare $A^{(k+1)}$ per M_k^{-1} che e' la matrice inversa di M_k ed ha, per elementi della colonna k nelle righe i-esime con $i > k$, le quantita' l_{ik} anziche' $-l_{ik}$.

Essendo il prodotto tra matrici associativo, e' facile dimostrare che $M_1^{-1} \dots M_{n-2}^{-1} M_{n-1}^{-1} M_{n-1} M_{n-2} \dots M_1 A = A = M_1^{-1} \dots M_{n-1}^{-1} U$ e la matrice $L = M_1^{-1} \dots M_{n-1}^{-1}$ a diagonale unitaria formata dagli elementi l_{ij} di tutte le matrici M_k nella rispettiva posizione, costituisce proprio la matrice triangolare inferiore per cui $LU = A$.

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{n1} & \cdots & l_{n(n-1)} & 1 \end{bmatrix} \qquad \text{con } l_{ij} = a_{ij}/a_{jj}$$

Risulta pratico memorizzare gli elementi l_{ij}, mano a mano che vengono calcolati applicando MEG, al posto degli 0 ottenuti su A.

Pivoting parziale

L'algoritmo si arresta se $a_{kk}^{(k)} = 0$, impedendo il calcolo di $Ax = b$ anche con A non singolare. Inoltre elementi $a_{kk}^{(k)}$ molto piccoli causano problemi di stabilita' numerica.

Si introduce quindi il pivoting parziale: all'inizio di ogni passo k, si sostituisce $a_{kk}^{(k)}$ con l'elemento maggiore in valore assoluto tra $a_{ik}^{(k)}$ dove $i \geq k$, operando eventualmente lo scambio di righe appropriato su $A^{(k)}$ e parallelamente su $L^{(k)} = M_1^{-1} \dots M_{k-1}^{-1}$.

Si ottiene $PA = LU$ con P un'opportuna matrice di permutazione non individuabile a priori. Si verifica che P, inizialmente posta uguale all'identita', subisce gli stessi scambi di righe a cui sono sottoposte le matrici $A^{(k)}$ ed $L^{(k)}$. Essendo $PA = LU$, moltiplicando $Ax = b$ a sinistra per P, si nota che la risoluzione del sistema iniziale diviene $Ly = Pb$ e $Ux = y$.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{3} & -\frac{2}{3} & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 3 & 7 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad x = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

Fattorizzazione di Gauss con Matlab

In Matlab e' possibile calcolare le matrici L, U e P con la fattorizzazione di Gauss e pivoting parziale utlizzando l'istruzione `lu`.

`A=[`$A_{11}, \dots, A_{1n}; \dots; A_{n1}, \dots, A_{nn}$`];`

```
[L,U,P]=lu(A)
```

Il sistema $Ax = b$ puo' essere risolto con il comando `\`.

`b=[`$b_1; \dots; b_n$`];`

```
A\b
```

Metodi `iterativi`

Dato il sistema lineare $Ax = b$ e lo splitting $P - Q$, si scriva esplicitamente il metodo iterativo $Px^{k+1} = b + Qx^k$, se ne studi la convergenza e si facciano due iterazioni a partire dal vettore x^0.

A			x	b
$\begin{bmatrix}1 & -1 & 0\\ 2 & 2 & 0\\ 1 & 0 & 1\end{bmatrix}$			$\begin{bmatrix}x\\ y\\ z\end{bmatrix}$	$\begin{bmatrix}0\\ 4\\ 2\end{bmatrix}$

P			x^0
$\begin{bmatrix}2 & 0 & 0\\ 0 & 2 & 0\\ 2 & 1 & 1\end{bmatrix}$			$\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}$

Table 2: Valori dall'esercizio 2 della prova di esame del 26 agosto 2014.

Metodi iterativi

I metodi iterativi sono utilizzati alternativamente ai metodi diretti per la risoluzione di sistemi lineari $Ax = b$, quando A e' una matrice di ordine elevato e sparsa. Da una stima iniziale x^0, si costruisce una successione di vettori $\{x^k\}$ che converge alla soluzione esatta x.

Consideriamo una decomposizione (splitting) di A del tipo $A = P - Q$ con P una matrice non singolare. Allora $(P - Q)x = b$, da cui il procedimento iterativo

$$Px^{k+1} = b + Qx^k$$

Dopo aver calcolato $Q = P - A$, dalla formula, si ottiene il sistema

$$\begin{cases} x^{k+1} = \frac{1}{2}x^k + \frac{1}{2}y^k \\ y^{k+1} = 2 - x^k \\ z^{k+1} = x^k \end{cases}$$

In due iterate si trova $x^2 = 1$, $y^2 = 2$ e $z^2 = 0$.

Studio della convergenza

Definita la matrice di iterazione $B = P^{-1}Q$, il metodo si puo' scrivere nella forma $x^{k+1} = g + Bx^k$, $g = P^{-1}b$. La convergenza e' verificata se e solo se $\rho(B) < 1$ dove $\rho(B) = \max\{|\lambda_i|\}$ e' il raggio spettrale di B.

Anziche' calcolare gli autovalori di B come radici del polinomio caratteristico $|P^{-1}Q - \lambda I| = 0$, conviene l'equivalente

$$|\lambda P - Q| = 0$$

Il modulo di autovalori complessi $z = \alpha + i\beta$ vale $|z| = \sqrt{\alpha^2 + \beta^2}$.

$$\rho(B) \approx 0.7$$

Il raggio spettrale di B e' minore di 1, dunque il metodo converge.

Metodo di `Newton`

$f(x)$	x_0
$e^x - x\sin x$	0

Table 3: Valori dall'esercizio 3 della prova di esame del 26 agosto 2014.

Si scriva il metodo di Newton per l'equazione non lineare $f(x) = 0$ e si approssimi una soluzione con 2 iterate del metodo a partire da x_0.

Metodo di Newton

Consideriamo una funzione non lineare $f(x)$ di cui vogliamo trovare uno zero, un valore α per cui $f(\alpha) = 0$, in prossimita' di un punto x_0.

Cerchiamo il valore x_{k+1} che annulla la retta tangente al punto x_k con $k = 0, \ldots, N$.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k \geq 0$$

Il metodo suggerisce di calcolare lo zero α di una funzione f sostituendo localmente ad f la sua retta tangente. Un risultato analogo puo' essere ottenuto sviluppando f in serie di Taylor in un intorno di x_k e trascurando $o(x_{k+1} - x_k)$.

$$x_{k+1} = x_k - \frac{e^{x_k} - x_k \sin x_k}{e^{x_k} - \sin x_k - x_k \cos x_k}$$

In due iterate otteniamo

$$x_1 = -1, \quad x_2 \approx -0.73$$

Metodo di Newton con Matlab

Lo zero di una funzione puo' essere calcolato in Matlab con N iterate del metodo di Newton definendo una nuova funzione `newton`.

```
function x=newton(fun,dfun,x,N)
for j=1:N
     x=x-fun(x)/dfun(x);
end
```

dove `x` in input e' il valore di x_0 e `dfun` la derivata prima della funzione `fun`. La funzione puo' essere richiamata come segue.

```
fun=@(x) exp(x)-x*sin(x); dfun=@(x) exp(x)-sin(x)-x*cos(x);
newton(fun,dfun,0,2)
```

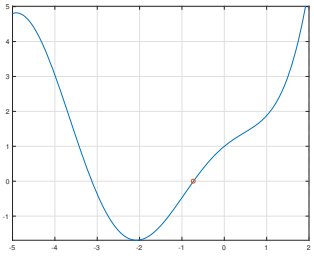

Figure 1: Il grafico della funzione $f(x) = e^x - x\sin x$. In rosso, lo zero α trovato con il metodo di Newton a partire da $x_0 = 0$.

Interpolazione `polinomiale`

Per i dati elencati di fianco, calcolare la tabella delle differenze divise, i polinomi intermedi $p_k(x)$ che interpolano i punti (x_i, y_i) con $i = 0, 1, \ldots, k$ e il polinomio interpolante $p(x)$.

x_i	y_i
0	-1
-1	-4
1	2
2	17
3	56

Table 4: Valori dall'esercizio 4 della prova di esame del 26 agosto 2014.

Interpolazione di Newton

Consideriamo $n+1$ coppie ordinate di valori (x_i, y_i) con $i = 0, 1, \ldots, n$ e costruiamo un polinomio in modo incrementale, ovvero tale che $p_{k+1}(x)$ possa essere ricavato a partire da $p_k(x)$, e per cui valga

$$p(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_n x_i^n = y_i$$

Differenze divise

In generale il polinomio interpolante si scrive come

$$p(x) = f[x_0]\omega_0(x) + f[x_0, x_1]\omega_1(x) + \ldots + f[x_0, x_1, \ldots, x_n]\omega_n(x)$$

dove $\omega_0(x) = 1$, $\omega_1(x) = (x - x_0)$, $\omega_2(x) = (x - x_0)(x - x_1)$, $\ldots$

Il coefficiente $f[x_0, x_1, \ldots, x_k]$ e' chiamato k-esima differenza divisa di Newton e puo' essere calcolato ricorsivamente nel modo seguente.

$$f[x_i] = f(x_i) = y_i$$

$$f[x_i, \ldots, x_k] = \frac{f[x_{i+1}, \ldots, x_k] - f[x_i, \ldots, x_{k-1}]}{x_k - x_i} \quad \text{per } k \geq i+1$$

Calcolo della tabella

Il calcolo della tabella delle differenze divise viene fatto a partire dalla formula ricorsiva precedente.

x_i	$f[x_i]$	$f[x_{i-1}, x_i]$	$f[x_{i-2}, x_{i-1}, x_i]$	$f[x_{i-3}, x_{i-2}, x_{i-1}, x_i]$	$f[x_{i-4}, x_{i-3}, x_{i-2}, x_{i-1}, x_i]$
0	-1				
-1	-4	3			
1	2	3	0		
2	17	15	4	2	
3	56	39	12	2	0

Differenze divise in Matlab

Le tabella delle differenze divise puo' essere calcolata in Matlab definendo una nuova funzione `divdiff`.

```
function A=divdiff(x,y)
n=length(x); A=zeros(n); A(:,1)=y;
for j=2:n
     for i=j:n
          A(i,j)=(A(i,j-1)-A(i-1,j-1))/(x(i)-x(i-j+1));
     end
end
```

La funzione puo' essere richiamata, dopo aver creato i vettori $x = [x_0, \ldots, x_n]$ e $y = f(x) = [y_0, \ldots, y_n]$ che contengono le coordinate dei punti (x_i, y_i), nel modo seguente.

```
>> x = [ x_0 , x_1 , ... , x_n ];  y = [ y_0 , y_1 , ... , y_n ];
>> divdiff(x,y)
```

I polinomi intermedi $p_k(x)$ e il polinomio interpolante $p(x)$

I polinomi intermedi possono essere calcolati a partire dalla tabella delle differenze divise ricordando

$$p(x) = f[x_0]\omega_0(x) + f[x_0, x_1]\omega_1(x) + \ldots + f[x_0, x_1, \ldots, x_n]\omega_n(x)$$

Si ottiene dunque $p_0(x) = -1$, $p_1(x) = 3x - 1$, $p_2(x) = 3x - 1$, $p_3(x) = 2x^3 + x - 1$, $p_4(x) = p(x) = 2x^3 + x - 1$.

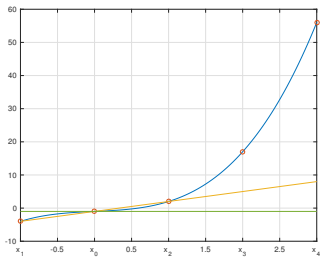

Figure 2: Il grafico dei polinomi intermedi $p_k(x)$ e del polinomio (di grado $k = n = 4$) interpolante i punti della tabella 4. Si noti che i coefficienti dei termini x^4 e x^2 sono nulli.

Calcolo dei coefficienti in Matlab

Dopo aver creato i vettori x e y che contengono le coordinate dei k punti da interpolare, si puo' utilizzare l'istruzione `polyfit`.

```
>> x = [ x_0 , x_1 , ... , x_k ];  y = [ y_0 , y_1 , ... , y_k ];
>> coeff = polyfit(x,y,k)
```

Il vettore `coeff = [` a_k , a_{k-1} , ... , a_0 `]` contiene i coefficienti del polinomio interpolante $p_k(x)$ di grado k.

Integrazione numerica

Dato il seguente integrale

$$\int_a^b f(x)dx$$

stimare il numero di intervalli necessari affinche' l'errore di approssimazione con il metodo dei trapezi e di Simpson sia minore di 10^{-4}. Calcolarlo con il metodo di Simpson e 4 intervalli (piccoli).

a	b	$f(x)$
-1	1	$e^{\cos x}$

Table 5: Valori dall'esercizio 5 della prova di esame del 26 agosto 2014.

Metodo dei trapezi

Consideriamo una suddivisione in n parti dell'intervallo $[a, b]$, $[x_{k-1}, x_k]$ con $k = 1, \ldots, n$ dove $x_0 = a$ e $x_n = b$. Ciascun sottointervallo ha lunghezza $H = (b-a)/n$ e punto medio $\tilde{x}_k = (x_{k-1} + x_k)/2$.

L'integrale I puo' essere calcolato con il metodo dei trapezi usando la formula seguente.

$$I_t = \frac{H}{2}\left[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)\right]$$

L'errore derivante dall'uso di questa formula vale

$$E_t = -\frac{b-a}{12}H^2 f^{(2)}(\xi)$$

per un opportuno $\xi \in [a, b]$ purche' $f \in C^2([a, b])$. Essendo ξ incognito si assume il valore che maggiora la derivata. Il numero di intervalli necessari affinche' l'errore sia inferiore ad un valore di tolleranza T si ottiene scrivendo H in forma esplicita.

$$n \geq \sqrt{\frac{(b-a)^3 |f^{(2)}(\xi)|}{12T}}$$

Essendo $f^{(2)}(x) = e^{\cos x}(\sin^2 x - \cos x)$, la derivata seconda si puo' maggiorare in modo abbastanza preciso nell'intervallo con 0.3 o si puo' procedere con una maggiorazione piu' approssimativa.

$$n \geq 45$$

Metodo dei trapezi con Matlab

L'integrale puo' essere calcolato in Matlab con il metodo dei trapezi per mezzo delle istruzioni `trapz` e `cumtrapz` dopo aver creato i vettori dei punti $x = [x_0, \dots, x_n]$ e $y = f(x) = [y_0, \dots, y_n]$. E' utile definire l'integranda $f(x)$, ad esempio come un'anonymous function.

```
>> x = [ x_0 , x_1 , ... , x_n ];  f = @(x) exp(cos(x));  y = f(x);
>> int = trapz(x,y)
>> step = cumtrapz(x,y)
```

`int` e' il valore dell'integrale, `step` contiene tutti i valori delle somme parziali con $k = 1, \dots, n$. L'ultimo valore di `step` coincide con `int`.

Regola di Simpson

L'integrale I puo' essere calcolato anche con la regola di Simpson.

$$I_S = \frac{H}{6} \sum_{k=1}^{n} [f(x_{k-1}) + 4f(\tilde{x}_k) + f(x_k)]$$

Considerando $N = 2n$ intervalli (piccoli) di lunghezza $h = (b-a)/N$

$$I_S = \frac{h}{3}[f(a) + 4f(x_1) + 2f(x_2) + \ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(b)]$$

Si ottiene $I_S \approx 4.68$. L'errore introdotto vale

$$E_S = -\frac{b-a}{180}\frac{H^4}{16} f^{(4)}(\xi) = -\frac{b-a}{180} h^4 f^{(4)}(\xi)$$

per un opportuno $\xi \in [a,b]$ purche' $f \in C^4([a,b])$. Il numero di intervalli necessari affinche' l'errore sia inferiore a T verifica

$$n \geq \sqrt[4]{\frac{(b-a)^5 |f^{(4)}(\xi)|}{2880T}}, \qquad N \geq \sqrt[4]{\frac{(b-a)^5 |f^{(4)}(\xi)|}{180T}} \text{ con } N \text{ pari}$$

La derivata quarta si puo' maggiorare in modo abbastanza preciso nell'intervallo con 10.9 o in modo piu' approssimativo.

$$n \geq 6, \qquad N \geq 12$$

Regola di Simpson in Matlab

L'integrale puo' essere calcolato in Matlab con la regola di Simpson definendo una nuova funzione `simpson`.

```
function int=simpson(fun,a,b,n)
h=(b-a)/n; x=a:h/2:b; y=fun(x);
int=h/6*(sum(y(1:2:end-2)+4*y(2:2:end-1)+y(3:2:end)));
```

La funzione puo' essere richiamata nel modo seguente.

```
>> f = @(x) exp(cos(x));
>> simpson(f,-1,1,2)
```

Equazioni differenziali

$f(x,y(x))$	x_0	y_0
$xy(x)$	0	1

A		b	c
$\begin{bmatrix}1/4\\1/4\end{bmatrix}$	$\begin{bmatrix}-1/4\\-5/12\end{bmatrix}$	$\begin{bmatrix}1/4\\3/4\end{bmatrix}$	$\begin{bmatrix}0\\2/3\end{bmatrix}$

Table 6: Valori dall'esercizio 6 della prova di esame del 26 agosto 2014.

Dato il seguente problema ai valori iniziali

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \end{cases}$$

e il metodo di Runge-Kutta definito nel tableau

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

calcolare l'ordine del metodo numerico, scriverlo esplicitamente e fare un passo con $h = \frac{1}{2}$.

Metodi di Runge-Kutta

I metodi RK sono metodi per l'approssimazione numerica di soluzioni di equazioni differenziali ordinarie e assumono la forma generale

$$y_{k+1} = y_k + \sum_{i=1}^{s} b_i K_i \quad \text{con} \quad K_i = hf(x_k + c_i h,\ y_k + \sum_{j=1}^{s} a_{ij} K_j)$$

dove s rappresenta il numero degli stadi e h il passo.

I coefficienti a_{ij}, b_i e c_i, che caratterizzano completamente un metodo Runge-Kutta, vengono generalmente raccolti in un tableau.

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array}$$

Se $a_{ij} = 0$ per $j \geq i$ allora il metodo RK e' esplicito.

$$y_{k+1} = y_k + \frac{1}{4}K_1 + \frac{3}{4}K_2$$

$$\begin{cases} K_1 = \frac{1}{2}x_k(y_k + \frac{1}{4}K_1 - \frac{1}{4}K_2) \\ K_2 = \frac{1}{2}(x_k + \frac{1}{3})(y_k + \frac{1}{4}K_1 - \frac{5}{12}K_2) \end{cases}$$

Ordine di un metodo

Il metodo deve anzitutto essere consistente, ovvero deve valere

$$\sum_{j=1}^{s} a_{ij} = c_i \quad \text{o in forma matriciale} \quad A\mathbb{1} = c$$

con $\mathbb{1}$ il vettore colonna i cui elementi sono costituiti dal numero 1.

Il metodo si dice di ordine 1 se vale la condizione di consistenza e

$$\sum_{j=1}^{s} b_j = 1 \quad \text{o equivalentemente} \quad b \cdot \mathbb{1} = 1$$

Il metodo si dice di ordine 2 se, oltre alle condizioni precedenti, vale

$$\sum_{j=1}^{s} b_j c_j = \frac{1}{2} \quad \text{o equivalentemente} \quad b \cdot c = \frac{1}{2}$$

Si dice di ordine 3 se, oltre alle condizioni precedenti, valgono

$$\sum_{j=1}^{s} b_j c_j^2 = \frac{1}{3} \text{ e } \sum_{i,j=1}^{s} b_i a_{ij} c_j = \frac{1}{6} \quad \text{o} \quad b \cdot c^2 = \frac{1}{3} \text{ e } b \cdot (Ac) = \frac{1}{6}$$

Infine il metodo si dice di ordine 4 se valgono anche

$$\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4}, \sum_{i,j=1}^{s} b_i c_i a_{ij} c_j = \frac{1}{8}, \sum_{i,j=1}^{s} b_i a_{ij} c_j^2 = \frac{1}{12} \text{ e } \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k = \frac{1}{24}$$

o in modo equivalente

$$b \cdot c^3 = \frac{1}{4}, bc \cdot (Ac) = \frac{1}{8}, b \cdot (Ac^2) = \frac{1}{12} \text{ e } (A^T b) \cdot (Ac) = \frac{1}{24}$$

Le condizioni derivano dall'analisi dello sviluppo in serie di Taylor della soluzione esatta y_{k+1} in un intorno di t_k ampio h, e dal confronto con quello della soluzione approssimata, ottenuta eseguendo un passo del metodo RK a partire dalla soluzione esatta y_k.

Il metodo associato al tableau della tabella 6 non e' consistente. Con un passo del metodo, dalla condizione iniziale $y(0) = 1$

$$x_0 = 0, \; y_0 = 1$$

$$\begin{cases} K_1 = 0 \\ K_2 \approx 0.156 \end{cases}$$

$$y_1 = 1 + \frac{1}{4}K_1 + \frac{3}{4}K_2 \approx 1.117$$

Si calcoli l'ordine del metodo multistep definito dalle tabelle e lo si scriva esplicitamente per il seguente problema ai valori iniziali.

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \end{cases}$$

$f(x,y(x))$	x_0	y_0
$x + y(x)$	0	0

α_{-1}	α_0	α_1	α_2	α_3
3	1	-1	-1	-2

β_{-1}	β_0	β_1	β_2	β_3
1/2	19/3	0	5	1/6

Table 7: Valori dall'esercizio 6 della prova di esame del 30 luglio 2014.

Metodi multistep

Il generico metodo multistep lineare assume la forma

$$\sum_{i=-1}^{p} \alpha_i y_{k-i} = h \sum_{i=-1}^{p} \beta_i f(x_{k-i}, y_{k-i})$$

$$\alpha_{-1} y_{k+1} + \alpha_0 y_k + \ldots + \alpha_p y_{k-p} = h[\beta_{-1} f_{k+1} + \beta_0 f_k + \ldots + \beta_p f_{k-p}]$$

Per il metodo e la ODE indicati dai valori della tabella 7 si ha

$$y_{k+1} = \frac{2(19h-3)}{3(6-h)} y_k + \frac{2}{6-h} y_{k-1} + \frac{2(5h+1)}{6-h} y_{k-2} + \frac{h+12}{3(6-h)} y_{k-3} + \\ + \frac{h}{6-h} x_{k+1} + \frac{38h}{3(6-h)} x_k + \frac{10h}{6-h} x_{k-2} + \frac{h}{3(6-h)} x_{k-3}$$

La consistenza e' verificata se vale la condizione

$$\sum_{i=-1}^{p} \alpha_i = 0$$

Il metodo si dice di ordine 1 se vale la condizione di consistenza e

$$\sum_{i=-1}^{p} i\alpha_i + \beta_i = 0$$

Si dice di ordine 2 se, oltre alle condizioni precedenti, vale

$$\sum_{i=-1}^{p} i^2\alpha_i + 2i\beta_i = 0$$

In generale, un metodo multistep lineare si dira' di ordine o se, oltre alla condizione di consistenza, per ogni valore di $0 < r \leq o$ e' verificata la relazione

$$\sum_{i=-1}^{p} i^r\alpha_i + ri^{r-1}\beta_i = 0$$

Il metodo associato ai valori della tabella 7 e' di ordine 4.