

Calcolo Numerico e Matlab

Risoluzione di ODE

Silvia Falletta

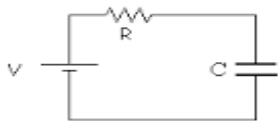
Dipartimento di Scienze Matematiche, Politecnico di Torino
silvia.falletta@polito.it

A.A. 2016/2017

Molti fenomeni reali sono modellizzati da equazioni (o sistemi di equazioni) differenziali ordinarie (ODE), cioè da equazioni che hanno come incognita una funzione che dipende da una sola variabile e che compare nell'equazione assieme alle sue derivate. Se nell'equazione compare la derivata prima, l'equazione è detta del primo ordine, se compaiono le derivate fino all'ordine m , l'equazione è detta di ordine m .

Alcuni esempi

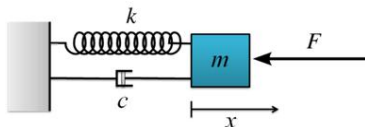
Esempio 1 Consideriamo un circuito RC in evoluzione libera, composto da una resistenza R e da un condensatore carico di capacità C . Evoluzione libera significa che il circuito non ha sorgenti esterne di tensione o di corrente, la corrente circolante è dovuta solo al movimento di cariche dovute all' energia immagazzinata nel condensatore e precedentemente fornita da una sorgente esterna. Applicando la legge di Kirchhoff delle tensioni, l'equazione del circuito è:



$$RC \frac{dv(t)}{dt} + v(t) = 0.$$

L'equazione è una **ODE del primo ordine**.

Esempio 2



L'evoluzione dinamica del sistema costituito dal modello **massa-molla-smorzatore** si può scrivere, in termini matematici, mediante l'equazione differenziale

$$m \frac{d^2 x(t)}{dt^2} + c \frac{dx(t)}{dt} + kx(t) = F(t)$$

dove $x(t)$ rappresenta lo spostamento della massa m dalla posizione di equilibrio, $\frac{dx(t)}{dt}$ la sua velocità, mentre $\frac{d^2 x(t)}{dt^2}$ rappresenta la sua accelerazione. I valori c e k rappresentano il coefficiente di smorzamento e la costante elastica della molla, rispettivamente. La funzione $F(t)$ rappresenta una forza esterna. L'equazione è una **ODE del secondo ordine**.

Esempio 3 Dalla legge di gravitazione universale di Newton, si deduce l'equazione del moto della terra attorno al sole. Indicando con \mathbf{r} il vettore che congiunge la terra al sole, si può scrivere l'equazione differenziale

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{GM}{\|\mathbf{r}\|^3}\mathbf{r}$$

La costante GM prende il nome di costante gravitazionale geocentrica della terra. La norma che si considera nel termine $\|\mathbf{r}\|^3$ è la norma euclidea del vettore \mathbf{r} . L'equazione è una **ODE del secondo ordine**, in questo caso però di tipo **vettoriale** perchè compare la derivata di un vettore. Esplicitando l'equazione componente per componente, si ottiene un sistema di equazioni differenziali del secondo ordine.

Problemi differenziali a valori iniziali

Un problema è ben posto se si assegna un dato iniziale.

Si definisce problema a valori iniziali o problema di Cauchy il seguente problema differenziale:

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

Tale espressione prende il nome di **forma canonica** del problema di Cauchy. Risolvere un problema di Cauchy significa determinare una funzione $y \in C^1$, soddisfacente l'equazione differenziale $y'(x) = f(x, y(x))$ e passante per il punto (x_0, y_0) .

Esempio

- Il problema $y'(x) = y$, $x > 0$ ammette infinite soluzioni, $y(x) = ce^x$, $c \in \mathbb{R}$;
- Il problema $\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$ ammette una ed una sola soluzione, $y(x) = e^x$;

Esistono condizioni sulla funzione f che garantiscono l'esistenza e l'unicità della soluzione di un problema di Cauchy. Noi assumeremo che tali condizioni siano soddisfatte e pertanto tratteremo solo problemi di Cauchy che ammettono una ed una sola soluzione.

Metodi numerici: classificazione

Consideriamo un problema di Cauchy posto in forma canonica

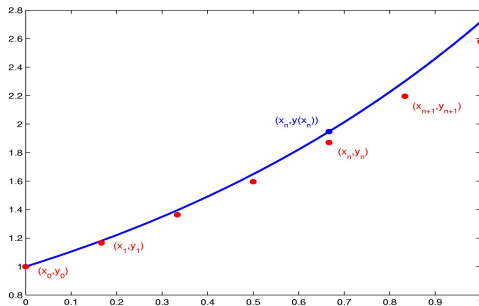
$$\begin{cases} y'(x) = f(x, y) & x > x_0 \\ y(x_0) = y_0 \end{cases}$$

e una suddivisione dell'intervallo di integrazione

$$x_0 < x_1 < x_2 < \dots < x_n < \dots$$

con $x_{n+1} = x_n + h$, $h = h(n)$ oppure h costante. Un generico metodo numerico fornisce le approssimazioni y_n della soluzione $y(x)$ nei punti x_n della partizione. Noti i valori y_n , per determinare le approssimazioni della soluzione $y(x)$ in punti diversi da x_n , si possono utilizzare le tecniche di interpolazione.

Graficamente si ha:



I metodi numerici vengono classificati nel seguente modo:

- ad un passo (one-step) $\begin{cases} \text{esplicito} \\ \text{implicito} \end{cases}$
- a più passi (multi-step) $\begin{cases} \text{esplicito} \\ \text{implicito} \end{cases}$

Si definiscono ad un passo quei metodi per i quali l'approssimazione y_{n+1} dipende unicamente dall'approssimazione y_n . Se

$$y_{n+1} = y_n + h\phi(x_n, y_n; h, f), \quad n = 0, 1, 2, \dots$$

allora il **metodo ad un passo** si dice **esplicito**; se

$$y_{n+1} = y_n + h\phi(x_n, x_{n+1}, y_n, y_{n+1}; h, f), \quad n = 0, 1, 2, \dots$$

il **metodo ad un passo** si dice **implicito**.

Si definiscono a k passi quei metodi per i quali l'approssimazione y_{n+1} dipende dalle approssimazioni precedenti $y_n, y_{n-1}, \dots, y_{n-k+1}$.
Se

$$y_{n+1} = \psi(x_n, x_{n-1}, \dots, x_{n-k+1}, y_n, y_{n-1}, \dots, y_{n-k+1}; h, f)$$

$$n = k - 1, k, k + 1, \dots$$

allora il **metodo a k passi** si dice **esplicito**; se

$$y_{n+1} = \psi(x_{n+1}, x_n, x_{n-1}, \dots, x_{n-k+1}, y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k+1}; h, f)$$

$$n = k - 1, k, k + 1, \dots$$

il **metodo a k passi** si dice **implicito**.

Esempio

- $y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$ definisce un metodo a 2 passi esplicito;
- $y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$ definisce un metodo a 1 passo implicito.

Metodi ad un passo

Costruiamo alcuni metodi ad un passo. A tal scopo consideriamo il generico problema di Cauchy in forma canonica

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

e, per semplicità, una partizione uniforme (a passo costante) dell'intervallo di integrazione

$$x_0 < x_1 < x_2 < \dots < x_n < \dots$$

con $x_{n+1} = x_n + h = x_0 + nh$, h costante.

Integrando la relazione $y'(x) = f(x, y)$ tra x_n e x_{n+1} , e considerando l'identità fondamentale del calcolo integrale, otteniamo:

$$\begin{aligned}\int_{x_n}^{x_{n+1}} y'(x) dx &= y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(x, y(x)) dx \\ \implies y(x_{n+1}) &= y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x)) dx\end{aligned}$$

Approssimiamo l'integrale al secondo membro con la formula del rettangolo:

$$\begin{aligned}\int_{x_n}^{x_{n+1}} f(x, y(x)) dx &\approx (x_{n+1} - x_n) f(x_n, y(x_n)) \\ &= hf(x_n, y(x_n))\end{aligned}$$

ed otteniamo quindi

$$y(x_{n+1}) \approx y(x_n) + hf(x_n, y(x_n))$$

Sostituiamo in quest'ultima \approx con $=$ e denotiamo con y_n l'approssimazione di $y(x_n)$ definita da

$$y_{n+1} = y_n + hf(x_n, y_n)$$

Quest'ultima espressione definisce il **metodo di Eulero esplicito**.

Comandi MATLAB

```
function [x,y] = Eulero_esp(f,x0,y0,xN,N)
%
x = linspace(x0,xN,N+1);
h = (xN-x0)/N;
y = y0*ones(1,N+1);
for n=1:N+1
    fn = f(x(n),y(n));
    y(n+1) = y(n) + h*fn;
end
```


Esempio

Applichiamo il metodo di Eulero esplicito al problema

$$\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. A tal scopo consideriamo una partizione dell'intervallo $[0, 0.2]$ con passo costante $h = 0.1$. Osserviamo che nel caso assegnato il metodo è definito da

$$y_{n+1} = y_n + hy_n = (1 + h)y_n = \dots = (1 + h)^n y_0$$

Pertanto, si ha

n	x_n	y_n
0	0	1
1	0.1	1.1
2	0.2	1.21

Poiché $\exp(0.2) = 1.2214\dots$, $|y_2 - y(x_2)| \approx 0.11e - 01$.

Analogamente, approssimando l'integrale al secondo membro con la formula del rettangolo

$$\begin{aligned}\int_{x_n}^{x_{n+1}} f(x, y(x)) dx &\approx (x_{n+1} - x_n) f(x_{n+1}, y(x_{n+1})) \\ &= hf(x_{n+1}, y(x_{n+1})),\end{aligned}$$

otteniamo

$$y(x_{n+1}) \approx y(x_n) + hf(x_{n+1}, y(x_{n+1}))$$

da cui ricaviamo

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

Quest'ultima espressione definisce il **metodo di Eulero implicito**. Osserviamo che il valore $z = y_{n+1}$ si ottiene risolvendo l'equazione, in generale non lineare,

$$g(z) := z - hf(x_{n+1}, z) - y_n = 0$$

Esempio

Applichiamo il metodo di Eulero implicito al problema

$$\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. A tal scopo consideriamo una partizione dell'intervallo $[0, 0.2]$ con passo costante $h = 0.1$. Osserviamo che nel caso (lineare) assegnato il metodo è definito da un'equazione di tipo lineare che può essere risolta analiticamente:

$$\begin{aligned} y_{n+1} &= y_n + h y_{n+1} \implies (1 - h) y_{n+1} = y_n \\ \implies y_{n+1} &= \frac{1}{1-h} y_n = \dots = \left(\frac{1}{1-h} \right)^n y_0 \end{aligned}$$

Pertanto si ha

n	x_n	y_n
0	0	1
1	0.1	1.111...
2	0.2	1.2345...

Poiché $\exp(0.2) = 1.2214\dots$, $|y_2 - y(x_2)| \approx 0.13e - 01$.

Costruiamo ora un altro metodo ad un passo, approssimando l'integrale al secondo membro con la formula del trapezio:

$$\int_{x_n}^{x_{n+1}} f(x, y(x)) dx \approx \frac{h}{2} [f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))]$$

Otteniamo

$$y(x_{n+1}) \approx y(x_n) + \frac{h}{2} [f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))]$$

Sostituiamo in quest'ultima \approx con $=$ e denotiamo con y_n l'approssimazione di $y(x_n)$ definita da

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

Quest'ultima espressione definisce il **metodo dei trapezi**.

Esempio

Applichiamo il metodo dei trapezi al problema

$$\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. A tal scopo consideriamo una partizione dell'intervallo $[0, 0.2]$ con passo costante $h = 0.1$. Osserviamo che nel caso assegnato il metodo è definito da

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}[y_n + y_{n+1}] \implies (1 - \frac{h}{2})y_{n+1} = (1 + \frac{h}{2})y_n \\ \implies y_{n+1} &= \frac{1+h/2}{1-h/2}y_n = \dots = \left(\frac{1+h/2}{1-h/2}\right)^n y_0 \end{aligned}$$

Pertanto, si ha

n	x_n	y_n
0	0	1
1	0.1	1.1052...
2	0.2	1.2216...

Poiché $\exp(0.2) = 1.2214\dots$, $|y_2 - y(x_2)| \approx 0.2e - 03$.

Altri due metodi molto usati:

il **metodo di Heun**:

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + h, y_n + hK_1) \end{cases}$$

il **metodo di Runge-Kutta a 4 stadi**

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_1) \\ K_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}K_2) \\ K_4 = f(x_n + h, y_n + hK_3) \end{cases}$$

Comandi MATLAB

```
function [x,y] = Heun(f,x0,y0,xN,N)
%
x = linspace(x0,xN,N+1);
h = (xN-x0)/N;
y = y0*ones(1,N+1);
for n=1:N+1
    K1 = f(x(n),y(n));
    K2 = f(x(n+1),y(n)+h*K1);
    y(n+1) = y(n) + h/2*(K1+K2);
end
```

Comandi MATLAB

```
function [x,y] = Runge_Kutta4(f,x0,y0,xN,N)
%
x = linspace(x0,xN,N+1);
h = (xN-x0)/N;
y = y0*ones(1,N+1);
for n=1:N+1
    K1 = f(x(n),y(n));
    K2 = f(x(n)+h/2,y(n)+h/2*K1);
    K3 = f(x(n)+h/2,y(n)+h/2*K2);
    K4 = f(x(n)+h,y(n)+h*K3);
    y(n+1) = y(n) + h/6*(K1+2*K2+2*K3+K4);
end
```


Esempio

Applichiamo il metodo di Heun e quello Runge-Kutta a 4 stadi al problema

$$\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. A tal scopo consideriamo una partizione dell'intervallo $[0, 0.2]$ con passo costante $h = 0.1$. Osserviamo che nel caso assegnato il metodo di Heun e Runge-Kutta a 4 stadi sono definiti rispettivamente da

$$\begin{cases} y_{n+1} = y_n + \frac{h}{2}(K_1 + K_2) \\ K_1 = y_n \\ K_2 = y_n + hK_1 \end{cases} \quad \begin{cases} y_{n+1} = y_n + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = y_n \\ K_2 = y_n + \frac{h}{2}K_1 \\ K_3 = y_n + \frac{h}{2}K_2 \\ K_4 = y_n + hK_3 \end{cases}$$

Pertanto, si ha

n	x_n	y_n (Heun)	y_n (R-K a 4 stadi)
0	0	1	1
1	0.1	1.1050	1.105170833...
2	0.2	1.221025...	1.221402570...

Poiché $\exp(0.2) = 1.221402758\dots$, $|y_2 - y(x_2)| \approx 0.38e - 03$ per il metodo di Heun e $|y_2 - y(x_2)| \approx 0.19e - 06$ per il metodo Runge-Kutta a 4 stadi.

Convergenza dei metodi ad un passo

Applichiamo il metodo di Eulero esplicito al problema

$$\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. Scegliamo come passo h della partizione dell'intervallo $[0, 0.2]$ valori sempre più piccoli. Ricordiamo che nel caso assegnato il metodo è definito da $y_{n+1} = (1 + h)^n y_0$. Si ha

h	n	x_n	y_n	$ y_n - y(x_n) $
0.1	2	0.2	1.21	$0.11e - 01$
0.1/2	4	0.2	1.215506...	$0.59e - 02$
0.1/4	8	0.2	1.218402...	$0.30e - 02$
0.1/10	20	0.2	1.220190...	$0.12e - 02$
0.1/100	200	0.2	1.221280...	$0.12e - 03$
0.1/1000	2000	0.2	1.221390...	$0.12e - 04$

Notiamo che al decrescere del passo h , diminuisce l'errore, ovvero aumenta l'accuratezza della approssimazione calcolata.

Introduciamo ora il concetto di convergenza. Un **metodo ad un passo** si dice **convergente** in $[a, b]$ se, qualunque sia il problema di Cauchy con f dotata di derivate parziali prime continue e limitate nella striscia $[a, b] \times \mathbb{R}$, per ogni $x \in [a, b]$ e $h = (x - a)/N$ risulta

$$\lim_{\substack{N \rightarrow \infty \\ (h \rightarrow 0)}} \|y_N - y(x)\| = 0$$

ove $\|\cdot\|$ denota una qualsiasi norma di vettore. Tutti i metodi sinora introdotti risultano convergenti; anzi è possibile dimostrare che essi risultano convergenti anche quando il passo h non è mantenuto costante.

Si dice, inoltre, che un metodo ha **ordine di convergenza p** se, qualunque sia f dotata di derivate parziali di ordine p continue e limitate nella striscia $[a, b] \times \mathbb{R}$, per ogni $x \in [a, b]$ e $h = (x - a)/N$ risulta

$$\|y_N - y(x)\| = O(h^p), \quad h \rightarrow 0$$

Osserviamo che più è elevato p , più è rapida la convergenza, più è accurata l'approssimazione. Si dimostra che:

- $p = 1$ per il metodo di Eulero esplicito;
- $p = 1$ per il metodo di Eulero implicito;
- $p = 2$ per il metodo dei trapezi;
- $p = 2$ per il metodo di Heun;
- $p = 4$ per il metodo Runge-Kutta a 4 stadi.

Esempio

Applichiamo i metodi di Eulero esplicito, di Heun e Runge-Kutta a 4 stadi al problema

$$\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 1. A tal scopo consideriamo una partizione dell'intervallo $[0, 1]$ con passo costante $h = 2^{-k}$, $k = 1, 2, \dots, 9$ e riportiamo in tabella l'errore $|y_N - y(1)|$.

Si ha

	$ y_N - y(1) $		
k	Eulero esplicito	Heun	R-K a 4 stadi
1	$4.68e - 01$	$7.77e - 02$	$9.36e - 04$
2	$2.77e - 01$	$2.34e - 02$	$7.19e - 05$
3	$1.52e - 01$	$6.44e - 03$	$4.98e - 06$
4	$8.04e - 02$	$1.69e - 03$	$3.28e - 07$
5	$4.13e - 02$	$4.32e - 04$	$2.10e - 08$
6	$2.09e - 02$	$1.09e - 04$	$1.33e - 09$
7	$1.05e - 02$	$2.75e - 05$	$8.38e - 11$
8	$5.29e - 03$	$6.89e - 06$	$5.26e - 12$
9	$2.65e - 03$	$1.73e - 06$	$3.29e - 13$

Osserviamo che per il metodo di Eulero esplicito

$|y_N - y(x_N)| = O(h)$ e, pertanto, per h sufficientemente piccolo

dimezzando il passo l'errore viene diviso per 2; per il metodo di

Heun $|y_N - y(x_N)| = O(h^2)$ e quindi dimezzando il passo l'errore

viene diviso per 4; infine, per il metodo Runge-Kutta a 4 stadi

$|y_N - y(x_N)| = O(h^4)$ e dimezzando il passo l'errore viene diviso

per 16.

La convergenza di un metodo numerico per problemi a valori iniziali si ha per $h \rightarrow 0$. L'errore decresce al diminuire del passo h . Esistono tuttavia dei metodi, che quando vengono applicati ad un certo tipo di problemi (cosiddetti "stiff"), generano un errore che, anziché diminuire, aumenta (fino ad esplodere) e decresce solo per valori di h inferiori ad una soglia h_0 , ove h_0 è un valore specifico per ogni metodo e per ogni problema. La soglia h_0 può assumere valori molto grandi ma anche molto piccoli: essa dipende dal metodo e dal problema a cui viene applicato il metodo. Per il metodo di Eulero implicito e il metodo dei trapezi h_0 è infinitamente grande qualunque sia il problema differenziale a cui viene applicato.

Esempio

Applichiamo il metodo di Eulero esplicito, implementato nella function `Eulero_esp`, al problema

$$\begin{cases} y'(x) = -10^3 y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(-10^3 x)$) per determinare un'approssimazione della soluzione in 1. A tal scopo definiamo la funzione $f(x, y)$ in una function

```
function fx = f(x,y)
fx = -10^3*y;
```

e poi digitiamo

```
>>[x,y]=Eulero_esp('f',0,1,1,400);
>>abs(exp(-10^3*x(end))-y(end))
ans =
    2.732144231480134e+070
```

```
>>[x,y]=Eulero_esp('f',0,1,1,500);  
>>abs(exp(-10^3*x(end))-y(end))  
ans =  
1
```

```
>>[x,y]=Eulero_esp('f',0,1,1,510);  
>> abs(exp(-10^3*x(end))-y(end))  
ans =  
1.377878756999681e-009
```

```
>>[x,y]=Eulero_esp('f',0,1,1,515);  
>>abs(exp(-10^3*x(end))-y(end))  
ans =  
3.769401716192881e-014
```

Quello che sta accadendo è che, sul problema specifico, i risultati numerici sono accettabili solo se h è inferiore a una soglia h_0 . Per tali valori, gli errori prodotti a ogni passo di integrazione non si propagano e quindi possiamo definire il metodo **stabile**.

Se h è troppo grande, gli errori prodotti si amplificano passo dopo passo e i risultati non sono accettabili. Per tali valori, il metodo è allora **instabile**. La situazione non è limitata solo al problema/metodo specifico: in generale, per i metodi espliciti è sempre presente una soglia, che dipende sia dal metodo usato che dal problema, e solo per h sufficientemente piccolo il metodo è stabile. Metodi impliciti come Eulero Implicito e Trapezi non hanno queste restrizioni su h ; per essi è quindi possibile prendere passi h più grandi. Questi metodi richiedono però la risoluzione, ad ogni passo, di equazioni tipicamente non lineari.

Regione di assoluta stabilità

Consideriamo il seguente problema modello:

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

con $\operatorname{Re}(\lambda) < 0$. La soluzione esatta è $y(t) = e^{\lambda(t-t_0)}$ e verifica

$$\lim_{t \rightarrow \infty} y(t) = 0$$

Per h fissato, vorremmo che il metodo numerico riproducesse una soluzione che abbia lo stesso comportamento quando il numero di passi tende a ∞ , cioè

$$\lim_{n \rightarrow \infty} y_n = 0$$

Se ciò succede, il metodo è detto **assolutamente stabile**.

Applichiamo il metodo di Eulero esplicito al problema modello.
Otteniamo

$$y_{n+1} = y_n + h\lambda y_n = (1+h\lambda)y_n = (1+h\lambda)^2 y_{n-1} = \dots (1+h\lambda)^{n+1} y_0.$$

Pertanto

$$\lim_{n \rightarrow \infty} y_n = 0 \iff |1 + h\lambda| < 1 \iff -2 < h\operatorname{Re}(\lambda) < 0$$

L'intervallo $(-2, 0)$ prende il nome di **intervallo di assoluta stabilità** del metodo di Eulero esplicito.

Applichiamo il metodo di Eulero implicito al problema modello.
Otteniamo

$$y_{n+1} = y_n + h\lambda y_{n+1} \Rightarrow y_{n+1} = \frac{1}{1 - h\lambda} y_n \Rightarrow \dots y_{n+1} = \left(\frac{1}{1 - h\lambda} \right)^{n+1} y_0.$$

Pertanto

$$\lim_{n \rightarrow \infty} y_n = 0 \iff |1 - h\lambda| > 1 \iff \forall h$$

Il metodo di Eulero implicito è stabile per ogni scelta di h .

Si può dimostrare che:

- il metodo dei Trapezi è stabile per ogni scelta di h ;
- il metodo di Heun è stabile per $-2 < h\operatorname{Re}(\lambda) < 0$;
- il metodo di Runge-Kutta a 4 stadi è stabile per $\alpha < h\operatorname{Re}(\lambda) < 0$, con $\alpha \in (-3, -2)$.

In generale, applicando un metodo ad un passo al problema modello $y'(t) = \lambda y(t)$, si ottiene una espressione del tipo

$$y_{n+1} = \mathcal{F}(h\lambda)y_n = \dots = \left(\mathcal{F}(h\lambda)\right)^{n+1} y_0$$

e, pertanto,

$$\lim_{n \rightarrow \infty} y_n = 0 \iff |\mathcal{F}(h\lambda)| < 1.$$

Si definisce quindi **regione di assoluta stabilità**

$$\mathcal{R}_a = \{h\lambda \in \mathbb{C} \text{ tali che } |\mathcal{F}(h\lambda)| < 1\}.$$

Esempio

- Per il metodo di Eulero esplicito, $\mathcal{F}(h\lambda) = 1 + h\lambda$;
- per il metodo di Eulero implicito, $\mathcal{F}(h\lambda) = \frac{1}{1-h\lambda}$;
- per il metodo di Heun, $\mathcal{F}(h\lambda) = 1 + h\lambda + \frac{1}{2}(h\lambda)^2$.

Stabilità di sistemi di equazioni differenziali

Per lo studio della stabilità di sistemi di equazioni differenziali, consideriamo il seguente problema modello

$$\begin{cases} y'(t) = Ay(t) & t > t_0 \\ y(t_0) = y_0, \end{cases}$$

dove A è una matrice di ordine m . Supponiamo che A sia diagonalizzabile e che i suoi autovalori $\lambda_1, \lambda_2, \dots, \lambda_m$ soddisfino $\operatorname{Re}(\lambda_i) < 0$, per $i = 1, \dots, m$. Siano $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ gli autovettori associati agli autovalori di A . Allora la soluzione del problema differenziale è data da

$$y(t) = c_1 e^{\lambda_1(t-t_0)} \mathbf{v}_1 + c_2 e^{\lambda_2(t-t_0)} \mathbf{v}_2 + \dots + c_m e^{\lambda_m(t-t_0)} \mathbf{v}_m,$$

con c_i tali che $\sum_{i=1}^m c_i \mathbf{v}_i = y_0$. La soluzione verifica

$$\lim_{t \rightarrow \infty} y(t) = 0.$$

Quindi, affinché il metodo risulti assolutamente stabile, occorre che

$$\lim_{n \rightarrow \infty} y_n = 0,$$

e cioè che $h\lambda_i \in \mathcal{R}_a$ per ogni $i = 1, \dots, m$. Osserviamo che, l'autovalore più grande in modulo fornisce la restrizione sulla scelta del passo di discretizzazione h . Ciò giustifica la seguente definizione:

Un sistema di equazioni differenziali di ordine m , è detto **stiff** nell'intervallo di integrazione $[t_0, t_0 + T]$ se

- eventuali autovalori con parte reale positiva sono tali che $\operatorname{Re}(\lambda_i)T$ non è grande;
- esiste almeno un autovalore con parte reale negativa tale che $\operatorname{Re}(\lambda_i)T \ll -1$.

La quantità $T \max_{\operatorname{Re}(\lambda_i) < 0} |\operatorname{Re}(\lambda_i)|$ rappresenta una misura del grado di stiffness del problema.

Esempio Consideriamo il problema

$$\begin{cases} y'(t) = Ay(t) & t \in [0, 100] \\ y(0) = y_0, \end{cases}$$

con

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

i cui autovalori sono $\lambda_1 = -100$ e $\lambda_2 = -1$. Applicando al problema il metodo di Eulero esplicito, occorre scegliere il passo di discretizzazione h tale che $h < \frac{2}{100} = 0.02$. Pertanto, per ottenere la soluzione in tutto l'intervallo di integrazione $[0, 100]$ occorre effettuare almeno $N = 100/h = 5000$ passi.

Esempio Consideriamo il problema

$$\begin{cases} y'(t) = Ay(t) & t \in [0, 10^4] \\ y(0) = y_0, \end{cases}$$

con

$$A = \begin{pmatrix} 0 & 1 \\ -10 & -11 \end{pmatrix}$$

i cui autovalori sono $\lambda_1 = -10$ e $\lambda_2 = -1$. Applicando al problema il metodo di Eulero esplicito, occorre scegliere il passo di discretizzazione h tale che $h < \frac{2}{10} = 0.2$. Pertanto, per ottenere la soluzione in tutto l'intervallo di integrazione $[0, 10^4]$ occorre effettuare almeno $N = 10^4/h = 5 \cdot 10^4$ passi.

ODE di ordine superiore al primo

Mostriamo ora come un problema a valori iniziali di ordine superiore al primo possa essere ricondotto ad un problema differenziale del primo ordine posto in forma canonica.

Consideriamo per esempio un problema del secondo ordine:

$$\begin{cases} y''(x) = g(x, y, y') & x > x_0 \\ y(x_0) = y_0 \\ y'(x_0) = y'_0 \end{cases}$$

Ponendo $z_1(x) = y(x)$, $z_2(x) = y'(x)$ e derivando ambedue i membri di queste due uguaglianze, otteniamo il seguente sistema del primo ordine nelle incognite $z_1(x)$ e $z_2(x)$, posto in forma canonica:

$$\begin{cases} z_1'(x) = z_2(x) & x > x_0 \\ z_2'(x) = g(x, z_1, z_2) \\ z_1(x_0) = y_0 \\ z_2(x_0) = y_0' \end{cases}$$

Introducendo infine le seguenti notazioni vettoriali

$$z(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \end{pmatrix}, \quad f(x, z(x)) = \begin{pmatrix} z_2(x) \\ g(x, z_1(x), z_2(x)) \end{pmatrix}, \quad z_0 = \begin{pmatrix} y_0 \\ y_0' \end{pmatrix}$$

possiamo riformulare il sistema nella seguente forma più compatta

$$\begin{cases} z'(x) = f(x, z) & x > x_0 \\ z(x_0) = z_0 \end{cases}$$

Esempio

$$\begin{cases} y''(x) - 3y'(x) + 2y(x) = 0 & x > 0 \\ y(0) = 1 \\ y'(0) = 1 \end{cases} \quad \Rightarrow \quad \begin{aligned} z_1(x) &= y(x) \\ z_2(x) &= y'(x) \end{aligned}$$

$$\begin{cases} z_1'(x) = z_2(x) & x > 0 \\ z_2'(x) = 3z_2(x) - 2z_1(x) \\ z_1(0) = 1 \\ z_2(0) = 1 \end{cases} \quad \Rightarrow \quad \begin{aligned} z(x) &= (z_1(x), z_2(x))^T \\ f(x, z(x)) &= (z_2(x), 3z_2(x) - 2z_1(x))^T \\ z_0 &= (1, 1)^T \end{aligned}$$

$$\begin{cases} z'(x) = f(x, z) & x > 0 \\ z(x_0) = z_0 \end{cases}$$

Analogamente un sistema di due equazioni differenziali del secondo ordine può essere ricondotto ad un sistema di 4 equazioni differenziali del primo ordine:

$$\left\{ \begin{array}{l} y''(x) = g_1(x, y, y', z, z') \\ z''(x) = g_2(x, y, y', z, z') \\ y(x_0) = y_0 \\ y'(x_0) = y'_0 \\ z(x_0) = z_0 \\ z'(x_0) = z'_0 \end{array} \right. \quad x > x_0$$

Ponendo $u_1(x) = y(x)$, $u_2(x) = y'(x)$, $u_3(x) = z(x)$, $u_4(x) = z'(x)$ e derivando ambedue i membri di queste quattro uguaglianze, otteniamo il seguente sistema del primo ordine nelle incognite $u_i(x)$, $i = 1, 2, 3, 4$:

$$\left\{ \begin{array}{ll} u_1'(x) = u_2(x) & x > x_0 \\ u_2'(x) = g_1(x, u_1, u_2, u_3, u_4) & \\ u_3'(x) = u_4(x) & \\ u_4'(x) = g_2(x, u_1, u_2, u_3, u_4) & \\ u_1(x_0) = y_0 & \\ u_2(x_0) = y_0' & \\ u_3(x_0) = z_0 & \\ u_4(x_0) = z_0' & \end{array} \right.$$

che in forma vettoriale diventa

$$\left\{ \begin{array}{ll} u'(x) = f(x, u) & x > x_0 \\ u(x_0) = u_0 & \end{array} \right.$$

con ovvio significato per i vettori coinvolti u , f ed u_0 .

Il generico problema differenziale a valori iniziali di ordine m è dunque riconducibile al seguente problema (in forma canonica) a valori iniziali di ordine 1:

$$\left\{ \begin{array}{l} y_1'(x) = f_1(x, y_1, y_2, \dots, y_m) \\ y_2'(x) = f_2(x, y_1, y_2, \dots, y_m) \\ \dots \quad \dots \\ y_m'(x) = f_m(x, y_1, y_2, \dots, y_m) \\ y_1(x_0) = y_{10} \\ y_2(x_0) = y_{20} \\ \dots \quad \dots \\ y_m(x_0) = y_{m0} \end{array} \right. \quad x > x_0 \quad \Longleftrightarrow \quad \left\{ \begin{array}{l} y'(x) = f(x, y) \\ y(x_0) = y_0 \end{array} \right. \quad x > x_0$$

È importante saper ricondurre un qualsiasi problema differenziale a valori iniziali ad un problema differenziale a valori iniziali di ordine 1 in forma canonica, in quanto solo a quest'ultimo è possibile applicare uno dei metodi numerici a disposizione.

Le *function* MATLAB `Eulero_espl`, `Heun` e `Runge_Kutta4` possono essere utilizzate anche per risolvere sistemi di equazioni differenziali. In questo caso, l'input `f` è il nome di una *function* in cui è definita una funzione vettoriale che dipende da due argomenti, di cui il primo è uno scalare, il secondo è un vettore colonna; `y_0` è un vettore colonna contenente i valori iniziali. In output, `y` è una matrice, il cui numero di colonne coincide con la lunghezza del vettore `x` e il cui numero di righe coincide con la dimensione del sistema. Ogni colonna di `y` contiene i valori della soluzione nel corrispondente punto definito in `x`.

Esempio

Applichiamo il metodo di Eulero esplicito al problema

$$\begin{cases} y''(x) - 3y'(x) + 2y(x) = 0 & x > 0 \\ y(0) = 1 \\ y'(0) = 1 \end{cases} \iff \begin{cases} z_1'(x) = z_2(x) & x > 0 \\ z_2'(x) = 3z_2(x) - 2z_1(x) \\ z_1(0) = 1 \\ z_2(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. A tal scopo definiamo la funzione $f(x, z(x)) = (z_2(x), 3z_2(x) - 2z_1(x))^T$ in una function

```
function fx = f(x,z)
fx = [z(2); 3*z(2)-2*z(1)];
```

e poi digitiamo

```
>>[x,y]=Eulero_espl_sist('f',0,[1;1],0.2,N);
```

dove la function `Eulero_esp1_sist` applica il metodo di Eulero esplicito al sistema ottenuto, ed è data da

Comandi MATLAB

```
function [x,y] = Eulero_esp_sist(f,x0,y0,xN,N)
%
x = linspace(x0,xN,N+1);
h = (xN-x0)/N;
m = length(y0);
y = zeros(m,N+1);
y(:,1) = y0;
for n=1:N+1
    fn = f(x(n),y(:,n));
    y(:,n+1) = y(:,n) + h*fn;
end
```

Esistono funzioni predefinite di Matlab per la risoluzione di equazioni differenziali ordinarie. **Comandi MATLAB**

- `[x,y]=solutore('f',[x_0,x_N],y_0)` risolve il problema a valori iniziali

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

In input si fornisce il nome tra apici della *function* in cui è definita la $f(x, y)$, che dipende da due argomenti. Si forniscono inoltre l'intervallo di integrazione $[x_0, x_N]$ e il valore iniziale y_0 . In output la *function* `solutore` restituisce un vettore colonna x contenente i punti nei quali la soluzione numerica è stata valutata e un vettore colonna y contenente i valori della soluzione nel corrispondente punto definito in x . Se si vogliono conoscere valutazioni della soluzione numerica in specifici punti, per esempio in x_0, x_1, \dots, x_N , occorre allora sostituire $[x_0, x_N]$ con $[x_0, x_1, \dots, x_N]$.

Con `solutore` abbiamo denominato una delle *function* presenti in MATLAB per la risoluzione di un problema a valori iniziali, per esempio, `ode45`, `ode113`, `ode23`, `ode23t`, `ode15s`, `ode23s`. Tali *function* implementano metodi di tipo esplicito e di tipo implicito. Per esempio,

- `ode45` è basata su una coppia di metodi Runge-Kutta espliciti di ordine 4 e 5;
- `ode23t` utilizza la formula dei trapezi;
- `ode15s` è utilizzata per risolvere problemi stiff.

Esempio

Applichiamo la function MATLAB ode45 al problema

$$\begin{cases} y'(x) = y & x > 0 \\ y(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. A tal scopo definiamo la funzione $f(x, y)$ in una function:

```
function fx = f(x,y)
fx = y;
```

e poi digitiamo

```

>> [x,y]=ode45('f',[0,0.2],1);
>> [x y]
ans =
           0 1.0000000000000000e+000
    5.0000000000000001e-003 1.005012520860555e+000
    1.0000000000000000e-002 1.010050167085803e+000
    1.5000000000000000e-002 1.015113064616412e+000
    2.0000000000000000e-002 1.020201340026773e+000
    2.5000000000000001e-002 1.025315120525624e+000
    .....
    1.9500000000000000e-001 1.215310986490748e+000
    2.0000000000000000e-001 1.221402758160380e+000
>> abs(exp(0.2)-y(end))
ans =
    2.098321516541546e-013

```

Osserviamo che la partizione dell'intervallo di integrazione $[0, 0.2]$ è uniforme con passo costante $h = 0.005$.

Le *function* MATLAB `ode...` possono essere utilizzate anche per risolvere sistemi di equazioni differenziali. In questo caso, in input `f` è il nome di una *function* in cui è definita una funzione vettoriale che dipende da due argomenti, di cui il primo è uno scalare, il secondo è un vettore colonna; `y_0` è un vettore colonna contenente i valori iniziali. In output, `y` è una matrice, il cui numero di righe coincide con la lunghezza del vettore `x` e il cui numero di colonne coincide con la dimensione del sistema. Ogni riga di `y` contiene i valori della soluzione nel corrispondente punto definito in `x`.

Esempio

Applichiamo la function MATLAB ode45 al problema

$$\begin{cases} y''(x) - 3y'(x) + 2y(x) = 0 & x > 0 \\ y(0) = 1 \\ y'(0) = 1 \end{cases} \iff \begin{cases} z_1'(x) = z_2(x) & x > 0 \\ z_2'(x) = 3z_2(x) - 2z_1(x) \\ z_1(0) = 1 \\ z_2(0) = 1 \end{cases}$$

(con soluzione $y(x) = \exp(x)$) per determinare un'approssimazione della soluzione in 0.2. A tal scopo definiamo la funzione $f(x, z(x)) = (z_2(x), 3z_2(x) - 2z_1(x))^T$ in una function

```
function fx = f(x,z)
fx = [z(2); 3*z(2)-2*z(1)];
```

e poi digitiamo

```
>> [x,y]=ode45('f',[0,0.2],[1 1]);
>> [x y(:,1)]
ans =
```

0	1.0000000000000000e+000
5.0000000000000001e-003	1.005012520860555e+000
1.0000000000000000e-002	1.010050167085803e+000
1.5000000000000000e-002	1.015113064616412e+000
2.0000000000000000e-002	1.020201340026773e+000
2.5000000000000001e-002	1.025315120525624e+000
.....
1.9500000000000000e-001	1.215310986490748e+000
2.0000000000000000e-001	1.221402758160380e+000

```
>> abs(exp(0.2)-y(end,1))
ans =
2.098321516541546e-013
```