

# **Analisi Numerica**

## **(A.A. 2014-2015)**

**Appunti delle lezioni:**  
**Equazioni differenziali ordinarie**

Docente Vittoria Bruni

Email: [vittoria.bruni@sbai.uniroma1.it](mailto:vittoria.bruni@sbai.uniroma1.it)

Ufficio: Via A. Scarpa,

Pal. B, I piano, Stanza n. 16

Tel. 06 49766648

Ricevimento: Martedì 9.30-10.30

Testi consigliati:

Calcolo Numerico, L. Gori, Ed. Kappa, 2006

Esercizi di Calcolo Numerico, L. Gori-M.L. Lo Cascio, F. Pitolli, Ed. Kappa, 2007

Il materiale didattico è disponibile sul sito

<http://www.sbai.uniroma1.it/users/bruni-vittoria>

nella pagina dedicata al corso [Analisi Numerica](#)

# Equazioni differenziali

Le equazioni differenziali rappresentano uno strumento efficace per la costruzione di modelli matematici per la simulazione di sistemi dinamici in diversi ambiti dell'ingegneria ma anche della biologia, della chimica, della fisica, dell'informatica e delle scienze sociali.

Sono equazioni che esprimono un legame tra una o più funzioni incognite e le loro derivate.

Se le derivate si riferiscono ad una sola variabile, allora parliamo di *equazioni differenziali ordinarie* (spesso indicate con l'acronimo **ODE**), per esempio

$$y'(t) = ay(t),$$

con  $y$  funzione incognita,  $t$  variabile indipendente e  $a$  costante; se, invece, le derivate si riferiscono a più variabili, allora parliamo di *equazioni differenziali alle derivate parziali* (in genere indicate con l'acronimo **PDE**), per esempio

$$u_t - au_{xx} = 0,$$

con  $u$  funzione incognita,  $t$  e  $x$  variabili indipendenti e  $a$  costante.

# Eq. differenziali ordinarie: modello matematico

Il moto di una **particella di massa**  $m$  attaccata all'estremità di una **molla di costante elastica**  $k$  è descritto dall'**equazione differenziale lineare del secondo ordine** (**equazione dell'oscillatore armonico semplice smorzato**):

$$m \frac{d^2x}{dt^2} + b \frac{dx}{dt} + kx = 0$$

$-b \frac{dx}{dt}$ : **forza di attrito**

$-kx$ : **legge di Hooke**

Dall'**analisi matematica** sappiamo che la soluzione "**esatta**" è

$$x(t) = x_m e^{-bt/2m} \cos(\omega_m t + \varphi_m)$$

$\omega_m = \sqrt{\frac{k}{m} - \frac{b^2}{4m^2}}$ : **pulsazione**  
dell'oscillatore

L'**ampiezza**  $x_m$  e la **fase**  $\varphi_m$  dell'oscillazione sono individuate dalle **condizioni iniziali**.

# Eq. differenziali ordinarie: problema di Cauchy

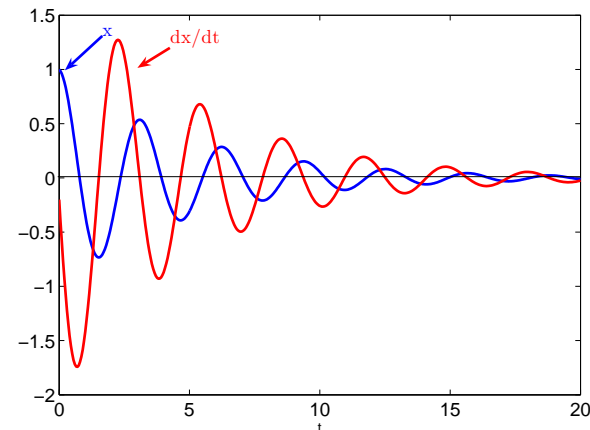
- Un **problema reale** richiede l'assegnazione delle **condizioni iniziali**.

**Problema di Cauchy:**

$$\left\{ \begin{array}{l} m \frac{d^2 x}{dt^2} + b \frac{dx}{dt} + kx = 0 \\ x(0) = x_0 \quad \text{c.i.} \\ \frac{dx}{dt}(0) = v_0 \quad \text{c.i.} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} x(t) = x_m e^{-bt/2m} \cos(\omega_m t + \varphi_m) \\ x_m = \frac{x_0}{\cos \varphi_m} \\ \tan \varphi_m = \frac{v_0}{x_0} \frac{2m}{b} \end{array} \right.$$

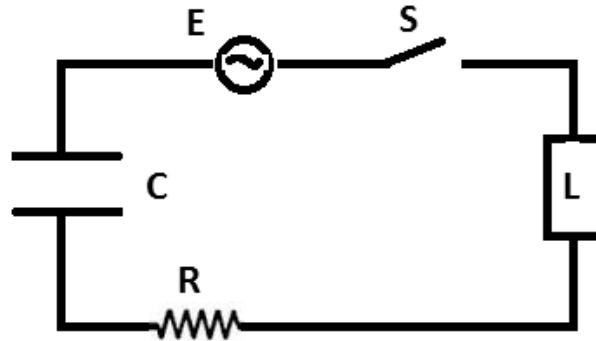
**Esempio:**

$$\left\{ \begin{array}{l} x(0) = 1 \\ \frac{dx}{dt}(0) = 0 \end{array} \right.$$



# Eq. differenziali ordinarie: modello matematico

Un modello analogo descrive un problema di origine diversa: determinare la carica  $Q(t)$  di un condensatore  $C$  nel circuito chiuso illustrato in figura, in cui  $R$  è la resistenza,  $L$  l'induttanza,  $E$  una sorgente di forza elettromotrice e  $S$  l'interruttore



In questo caso si applica la *seconda legge di Kirchhoff* che stabilisce che in un circuito chiuso la differenza di potenziale fornita attraverso la sorgente  $E$  è uguale alla somma delle differenze di potenziale misurate nel resto del circuito, ovvero

$$E(t) = RI(t) + L\frac{dI}{dt}(t) + \frac{Q(t)}{C}$$

con  $I$  l'intensità di corrente. Poichè  $I(t) = \frac{dQ}{dt}(t)$ , si ha

$$L\frac{d^2Q}{dt^2} + R\frac{dQ}{dt} + \frac{Q}{C} = E(t)$$

# Eq. differenziali ordinarie: problema di Cauchy

- Nei **problemi reali** l'espressione dell'equazione differenziale è **complicata** e, in genere, non si riesce a calcolare **esplicitamente** la soluzione.

## Esempio

Le **oscillazioni di un pendolo** possono essere descritte dall'equazione differenziale del secondo ordine **non lineare**

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin \theta = 0$$

dove  $L$  è la lunghezza del pendolo,  $g$  è l'accelerazione di gravità e  $\theta$  è l'angolo tra il pendolo e la verticale. Il problema è completato con le condizioni iniziali

$$\theta(t_0) = \theta_0 \quad \theta'(t_0) = v_0$$

**Dati:**  $g = 9.81 \text{ m/s}^2$ ,  $L = 0.6 \text{ m}$ ,  $\theta_0 = \pi/6$ ,  $v_0 = 0$

# Sistemi di equazioni differenziali: esempio

Il **modello preda-predatore** di Lotka-Volterra è il più semplice modello che descrive la competizione tra due specie. Consiste in una coppia di equazioni differenziali in cui  $y_1(t)$  rappresenta il numero di prede e  $y_2(t)$  rappresenta il numero di predatori. L'interazione tra le due specie è descritta da un termine proporzionale a entrambe le popolazioni:

$$\begin{cases} y_1'(t) = k_1 \left(1 - \frac{y_2(t)}{\mu_2}\right) y_1(t), & t > 0 \\ y_2'(t) = -k_2 \left(1 - \frac{y_1(t)}{\mu_1}\right) y_2(t) \end{cases}$$

dove  $k_1$ ,  $k_2$ ,  $\mu_1$ ,  $\mu_2$  sono costanti positive e  $y_{1,0}$ ,  $y_{2,0}$  rappresentano le condizioni iniziali.

In assenza di predatori le prede crescono in modo esponenziale; i predatori invece, in assenza di prede, muoiono rapidamente.

**Oss.** Se  $y_{1,0} = \mu_1$ ,  $y_{2,0} = \mu_2$ , allora  $y_1' = 0$  e  $y_2' = 0$ , ovvero  $(\mu_1, \mu_2)$  è un punto di equilibrio. Per particolari valori di  $\mu_1$ ,  $\mu_2$  la soluzione è periodica.



# Equazioni differenziali di ordine $n$

$$\begin{cases} y^{(n)}(x) = f(x, y(x), y'(x), y''(x), \dots, y^{(n-1)}(x)) \\ y^{(k)}(x_0) = y_k \quad k = 0, 1, \dots, n-1 \end{cases} \quad \text{Condizioni iniziali}$$

Un'equazione differenziale di ordine  $n$  può essere ricondotta a un sistema di  $n$  equazioni differenziali del primo ordine.

$$y^{(n)}(x) = f(x, \underbrace{y(x)}_{y_1(x)}, \underbrace{y'(x)}_{y_2(x)}, \underbrace{y''(x)}_{y_3(x)}, \dots, \underbrace{y^{(n-1)}(x)}_{y_n(x)})$$

$$y_1(x) = y(x)$$

$$\begin{cases} y_1'(x) = y_2(x) \\ y_2'(x) = y_3(x) \\ \dots\dots\dots \\ y_{n-1}'(x) = y_n(x) \\ y_n'(x) = f(x, y_1(x), y_2(x), y_3(x), \dots, y_n(x)) \end{cases} \quad \begin{cases} y_{10} = y_0 \\ y_{20} = y_1 \\ \dots\dots\dots \\ y_{n0} = y_{n-1} \end{cases} \quad \text{Condizioni iniziali}$$

# Eq. differenziali ordinarie del primo ordine /1

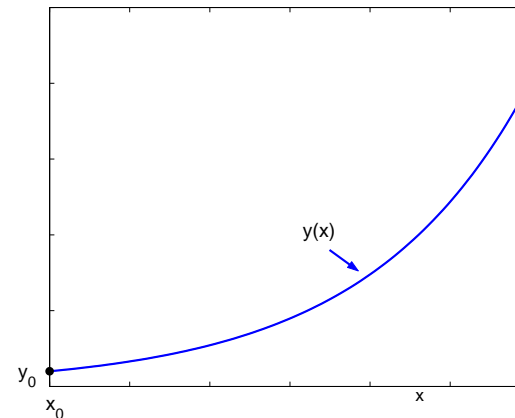
**Problema di Cauchy:**  $\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \end{cases}$  **condizione iniziale**

**Esempio:** Modello per descrivere la **crescita di una popolazione**

$$f(x, y(x)) = Ky(x) \quad K > 0 \quad \longrightarrow \quad \begin{cases} y'(x) = Ky(x) \\ y(x_0) = y_0 \end{cases}$$

**Soluzione esatta:**

$$y(x) = y_0 e^{K(x-x_0)}$$



## Eq. differenziali ordinarie del primo ordine /2

Nei **problemi applicativi**, l'espressione di  $f(x, y(x))$  è **complicata** e quindi non si può calcolare esplicitamente la soluzione del **problema di Cauchy** oppure la sua espressione è data tramite funzioni non elementari.

→ si deve ricorrere a un **metodo numerico** per approssimare  $y(x)$ .

### Esempio

Sia  $P(t)$  il **numero di individui** in una popolazione al tempo  $t$ , misurato in anni. Se il tasso medio di nascita  $b$  è costante e il tasso di morte  $d$  è proporzionale al numero di individui presenti al tempo  $t$ , allora il tasso di crescita è dato dall'equazione differenziale del primo ordine

$$\frac{dP(t)}{dt} = bP(t) - d(t)P(t), \quad d(t) = kP(t),$$

chiamata **equazione logistica**.

Assumendo  $P(0) = 50976$ ,  $b = 2.9 \times 10^{-2}$  e  $k = 1.4 \times 10^{-7}$ , calcolare il numero di individui dopo 5 anni.

# Esistenza e unicità della soluzione del problema di Cauchy

Prima di risolvere **numericamente** un'equazione differenziale bisogna essere sicuri che il **problema di Cauchy** ammetta un'**unica soluzione**  $y(x)$ .

**Definizione.** Una funzione  $f(x, y)$  si dice **lipschitziana** in  $y$  uniformemente rispetto a  $x$  in  $D \subset \mathbb{R}^2$ , se esiste una **costante**  $L > 0$  tale che

$$|f(x, y_1) - f(x, y_2)| \leq L |y_1 - y_2| \quad \forall (x, y_1), (x, y_2) \in D.$$

**Nota.** Una condizione sufficiente è che  $f_y$  esista e sia **limitata** in  $D$ .

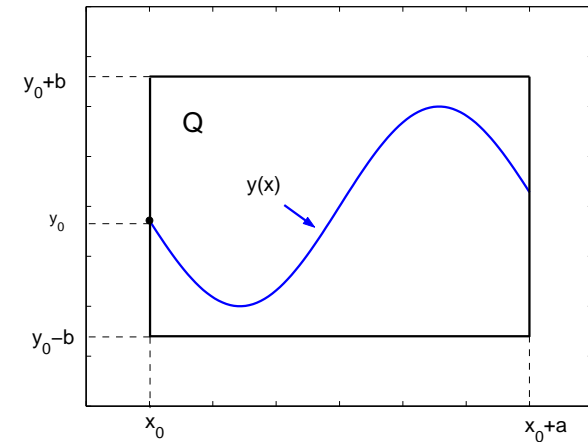
**Esempio:**  $f(x, y) = K y \rightarrow f_y(x, y) = K \rightarrow$  è **limitata**  
 $\rightarrow f$  è **lipschitziana**

Al contrario, la funzione  $f(x) = |x|$  è continua ma non  $C^1$  in un intervallo  $I$  limitato e simmetrico rispetto allo 0, in quanto la derivata non è continua in  $x = 0$ . Tuttavia, la funzione è **Lipschitziana** con  $L = 1$  in quanto  $\forall x_1, x_2 \in I$ , si ha  $||x_1| - |x_2|| \leq |x_1 - x_2|$ .

**Teorema 1.** Sia  $f$  **definita** e **continua** in  $Q$ :

$$Q := \{(x, y) \in \mathbb{R}^2 \mid x \in [x_0, x_0 + a]; y \in [y_0 - b, y_0 + b]\}$$

e sia **lipschitziana** in  $y$ , uniformemente rispetto a  $x$

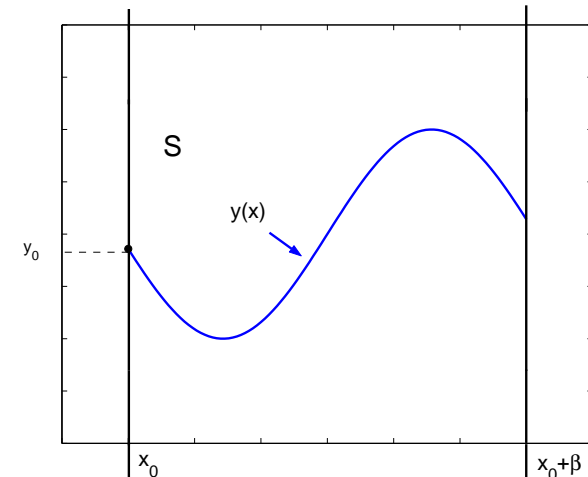


$\Rightarrow$  esiste un'**unica** soluzione  $y(x) \in C^1[x_0, x_0 + \beta]$  del **problema di Cauchy** in  $I = [x_0, x_0 + \beta]$  con  $\beta = \min \left[ a, \frac{b}{M} \right]$ ,  $M = \max_Q |f(x, y)|$ .

**Teorema 2.** Sia  $f$  **definita** e **continua** in  $S$ :

$$S := \{(x, y) \in \mathbb{R}^2 \mid x \in I = [x_0, x_0 + \beta]; y \in \mathbb{R}\}$$

e sia **lipschitziana** in  $y$ , uniformemente rispetto a  $x$



$\Rightarrow$  esiste un'**unica** soluzione  $y(x) \in C^1[x_0, x_0 + \beta]$  del **problema di Cauchy** in  $I = [x_0, x_0 + \beta]$ .

# Ben-posizione del problema di Cauchy

Per poter risolvere numericamente il problema di Cauchy è necessario anche che il problema sia **ben posto**.

**Definizione.** Il **problema di Cauchy**

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

è **ben posto** se, dette  $y(x; y_0)$  e  $y(x; y_0 + \delta)$  le soluzioni con **condizioni iniziali**  $y(x_0) = y_0$  e  $y(x_0) = y_0 + \delta$  rispettivamente, si ha

$$|y(x; y_0) - y(x; y_0 + \delta)| < \epsilon \quad x \in [a, b]$$

dove  $\epsilon > 0$  è una prefissata **tolleranza**, purché  $\delta = \delta(\epsilon)$  sia sufficientemente **piccolo**.

**Nota.** Se  $f$  soddisfa le condizioni del **Teorema 2**, il problema di Cauchy è **ben posto**.

# Esempi

- Il seguente **problema di Cauchy**:

$$\begin{cases} y'(t) = -y + t + 1 & t \in [0, 10] \\ y(0) = 1 \end{cases}$$

ammette soluzione unica in  $I = [0, 10]$ . Infatti,  $f(t, y) = -y + t + 1$  è definita e continua in  $[0, 10] \times \mathbf{R}$ . Inoltre,  $f_y(t, y) = -1, \quad \forall t \Rightarrow |f_y| = 1, \quad \forall t \in I \quad \forall y \in \mathbf{R}$

- Il seguente **problema di Cauchy**:

$$\begin{cases} y'(t) = 1 + t \sin(ty) & t \in [0, 2] \\ y(0) = 0 \end{cases}$$

ammette soluzione unica in  $I = [0, 2]$ . Infatti,  $f(t, y) = 1 + t \sin(ty)$  è definita e continua in  $[0, 2] \times \mathbf{R}$ . Inoltre,  $f_y(t, y) = t^2 \cos(ty), \quad \forall t \Rightarrow |f_y| = |t^2 \cos(ty)| \leq t^2 \leq 2^2 = 4, \quad \forall t \in I \quad \forall y \in \mathbf{R}$

# Esempio

Si consideri il seguente **problema di Cauchy**

$$\begin{cases} y'(t) = \frac{3}{2}y^{\frac{1}{3}} \\ y(0) = 0 \end{cases}$$

Le soluzioni del problema sono:

$$y(t) = 0 \quad \text{e} \quad y(t) = t^{\frac{3}{2}}$$

Inoltre, per  $a \in \mathbf{R}$ ,  $a > 0$ , la funzione

$$y_a(t) = \begin{cases} (t - a)^{\frac{3}{2}} & t > a \\ 0 & t = 0 \end{cases}$$

è soluzione del **problema di Cauchy**.

La soluzione, dunque, non è unica. Infatti,  $f(t, y)$  non è lipshitziana rispetto a  $y$ .



# Metodo di Eulero

**Problema di Cauchy:**  $\begin{cases} y'(x) = f(x, y(x)) & x \in [x_0, x_0 + \beta] \\ y(x_0) = y_0 & \text{condizione iniziale} \end{cases}$

**Discretizzazione di  $I$ :**  $x_i = x_0 + ih \quad i = 0, \dots, n \quad h = \frac{\beta}{n}$

**Metodo numerico:** I valori **esatti**  $y(x_i)$  vengono **approssimati** con i valori  $y_i$ .

Sviluppo in **serie di Taylor**:

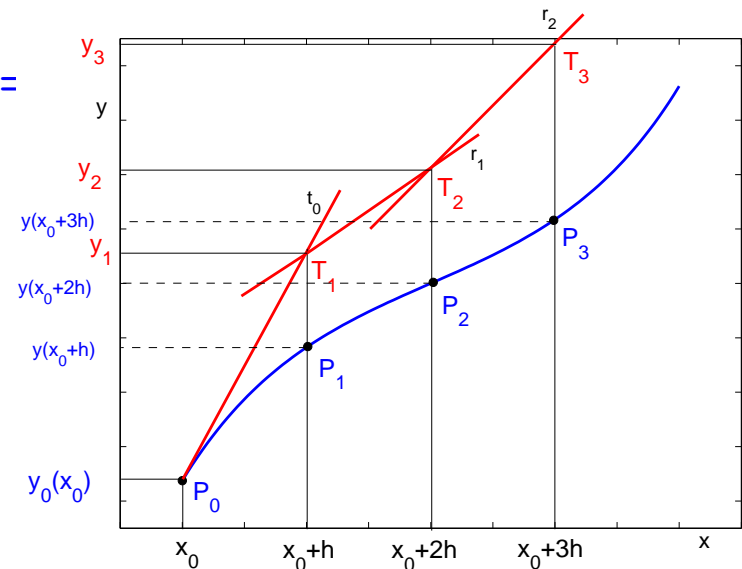
$$\begin{aligned} y(x_1) &= y(x_0 + h) = y(x_0) + y'(x_0)h + \frac{1}{2}y''(\tau_1)h^2 = \\ &= y(x_0) + f(x_0, y(x_0))h + \frac{1}{2}y''(\tau_1)h^2 \\ &\quad \tau_1 \in [x_0, x_1] \end{aligned}$$

Soluzione **approssimata**:

$$y_1 = y_0 + hf(x_0, y_0)$$

**Errore globale di troncamento:**

$$e_1(x_1) = y(x_1) - y_1 = \overline{P_1 T_1}$$



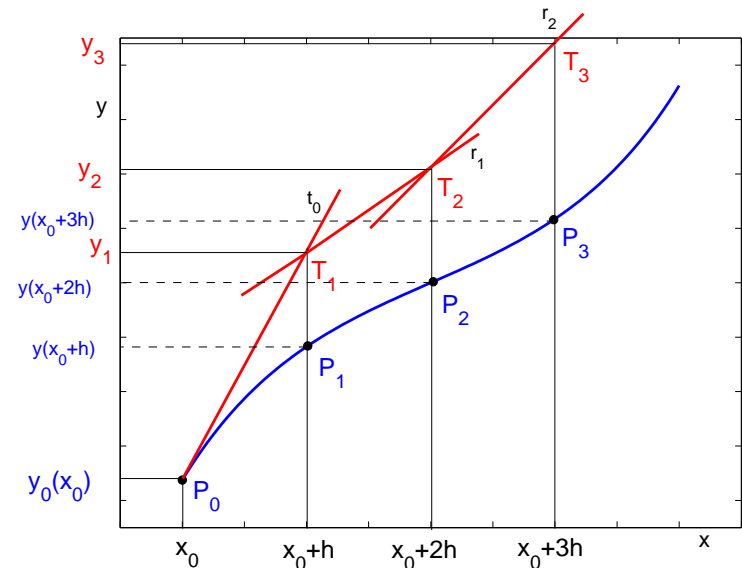
# Algoritmo del metodo di Eulero

**Algoritmo:**

$$y_{i+1} = y_i + hf(x_i, y_i) \quad i = 0, 1, \dots, n$$

**Errore globale di troncamento:**

$$e_i = y(x_i) - y_i = \overline{P_i T_i}$$



L'**errore globale di troncamento** ha due contributi:

- l'**errore locale di troncamento**, dovuto al fatto che la soluzione esatta  $y(x)$  viene **approssimata localmente** con una **retta**:

$$R(x_i, y(x_i); h; f) = \frac{1}{2}h^2 y''(\tau_i) \quad \tau_i \in [x_i, x_{i+1}]$$

$$R(x_i, y(x_i); h; f) = O(h^2) \quad \rightarrow \quad \text{Primo ordine}$$

- l'**accumularsi** degli errori locali di troncamento, per cui al generico passo  $i \geq 1$  ci si muove lungo la retta  $r_i$  che è una **approssimazione della retta tangente** alla soluzione in  $P_i = [x_i, y(x_i)]$ .

# Convergenza del metodo di Eulero

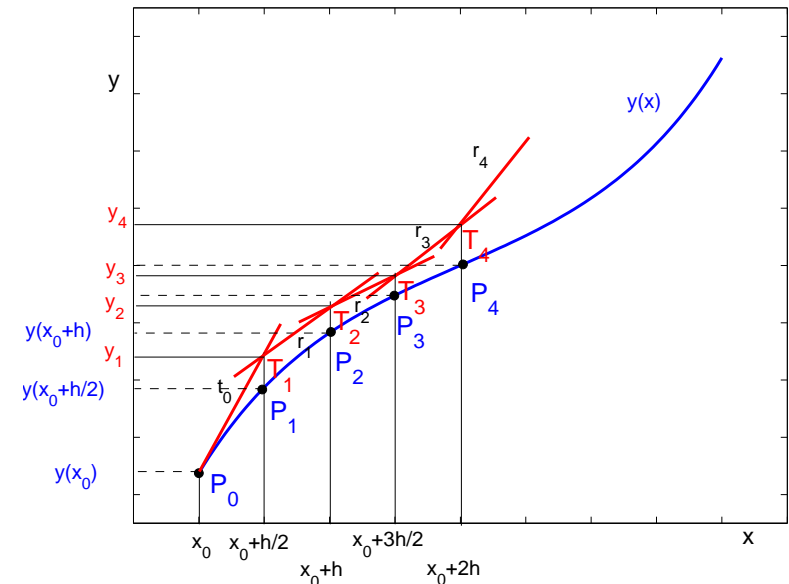
**Definizione.** Un **metodo numerico** per la soluzione approssimata di un'equazione differenziale è **convergente** se

$$\lim_{h \rightarrow 0} \max_{1 \leq i \leq n} |e_i| = 0 \quad \Leftrightarrow \quad \lim_{i \rightarrow \infty} y_i = y(\bar{x}) \quad \text{con } \bar{x} = x_0 + ih$$

Dalla figura è evidente che, se si **riduce il passo**  $h$ , si riduce anche l'**errore**  $e_i = \overline{P_i T_i}$ . Per di più

$$\lim_{h \rightarrow 0} \max_{1 \leq i \leq n} |e_i| = 0$$

⇒ Il metodo di Eulero è **convergente**



Un **metodo numerico** è **convergente** se è

- **Consistente:**

$$\lim_{h \rightarrow 0} \frac{R(x, y; h; f)}{h} = 0$$

+

- **Stabile:**

l'**accumularsi** degli errori locali di troncamento si mantiene **limitato**

## Esercizio

Dato il problema di Cauchy (PC)

$$\begin{cases} y' &= |y - x|, & x \in [0, 2] \\ y(0) &= 2 \end{cases}$$

verificare l'esistenza e unicità della soluzione in  $S = [0, 2] \times \mathbf{R}$  e approssimarla con il metodo di Eulero con passi  $h_1 = 0.1$  e  $h_2 = 0.05$  per  $x \in [0, 1]$

## Soluzione

Per verificare se vale il teorema di esistenza e unicità della soluzione bisogna verificare se  $f(x, y) = |y - x|$  è lipschitziana in  $S = [0, 2] \times \mathbf{R}$ , cioè se esiste una costante  $L > 0$  (costante di Lipschitz) tale che

$$|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|, \quad \forall (x, y_1), (x, y_2) \in S.$$

Per  $f(x, y) = |y - x|$  si ha

$$f(x, y) = \begin{cases} y - x, & x < y \\ x - y, & x \geq y \end{cases}$$

$$\Rightarrow f_y(x, y) = \begin{cases} 1, & x < y \\ -1, & x > y \end{cases} \quad \forall (x, y) \in S = [0, 2] \times \mathbf{R}.$$

Quindi,  $|f_y| = 1$   $x \neq y$  e  $|f_y| = 0$   $x = y$  (la derivata di  $f$  non è definita in  $y = x$ ). Poichè  $f_y$  risulta limitata in  $x \neq y$ , si può concludere che  $f(x, y)$  è lipschitziana in  $S$  con costante di Lipschitz pari a **1**. Dalla lipschitzianità di  $f$  segue anche che il metodo di Eulero è convergente per  $x \in [0, 2]$ .

Scegliendo il passo  $h = 0.1$ , definiamo i punti equidistanti

$$x_i = x_0 + ih$$

in cui si approssimeranno i valori della soluzione del PC, cioè

$$y_i = y_{i-1} + hf(x_{i-1}, y_{i-1}).$$

Poichè si chiede la soluzione nell'intervallo  $[0, 1]$ , sarà necessario eseguire

$$n = \frac{b - a}{h} = \frac{1}{0.1} = 10$$

passi del procedimento suddetto.

$$y_1 = y_0 + hf(x_0, y_0) = 2 + 0.1|2 - 0| = 2.2, \quad x_1 = x_0 + h = 0.1$$

$$y_2 = y_1 + hf(x_1, y_1) = 2.2 + 0.1|2.2 - 0.1| = 2.41, \quad x_2 = x_0 + 2h = 0.2$$

$$y_3 = y_2 + hf(x_2, y_2) = 2.41 + 0.1|2.41 - 0.2| = 2.631, \quad x_3 = x_0 + 3h = 0.3$$

$$y_4 = y_3 + hf(x_3, y_3) = 2.631 + 0.1|2.631 - 0.3| = 2.8641, \quad x_4 = x_0 + 4h = 0.4$$

$$y_5 = y_4 + hf(x_4, y_4) = 2.8641 + 0.1|2.8641 - 0.4| = 3.11051, \quad x_5 = x_0 + 5h = 0.5$$

⋮

⋮

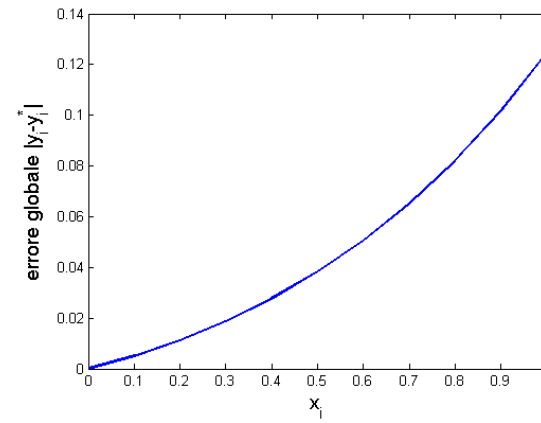
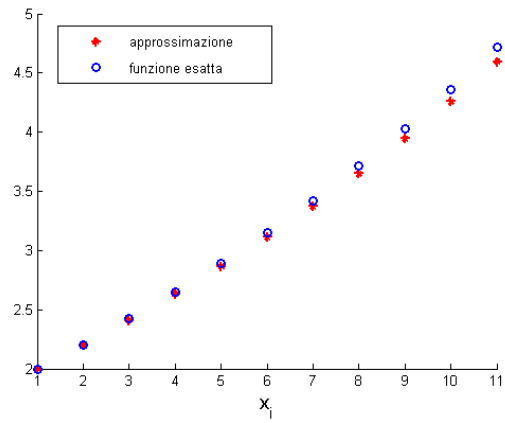
$$y_{10} = y_9 + hf(x_9, y_9) = 4.257947691 + 0.1|4.257947691 - 0.9| = 4.5937424601,$$

$$x_{10} = x_0 + 10h = 1$$



Per valutare l'accuratezza della approssimazione prodotta, confrontiamo i valori ottenuti con la soluzione esatta  $y^*(x) = e^x + x + 1$  valutando l'errore globale  $e_i = |y_i - y_i^*|$ , cioè

$x_i$	$y_i$	$y_i^*$	$e_i$
0.0	2.0000000000	2.0000000000000000	0.0000000000000000
0.1	2.2000000000	2.20517091807565	0.00517091807565
0.2	2.4100000000	2.42140275816017	0.01140275816017
0.3	2.6310000000	2.64985880757600	0.01885880757600
0.4	2.8641000000	2.89182469764127	0.02772469764127
0.5	3.1105100000	3.14872127070013	0.03821127070013
0.6	3.3715610000	3.42211880039051	0.05055780039051
0.7	3.6487171000	3.71375270747048	0.06503560747048
0.8	3.9435888100	4.02554092849247	0.08195211849247
0.9	4.2579476910	4.35960311115695	0.10165542015695
1.0	4.5937424601	4.71828182845905	0.12453936835904

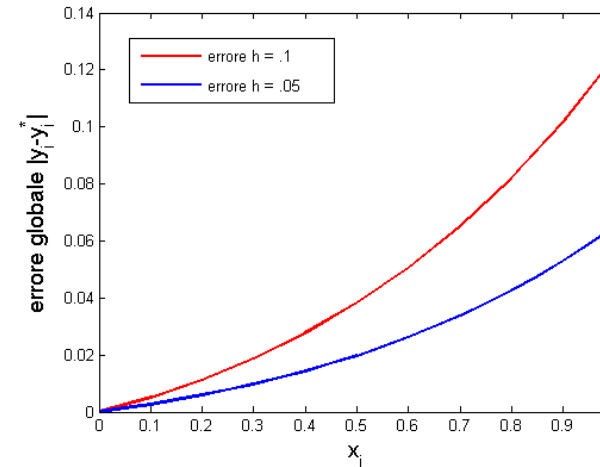
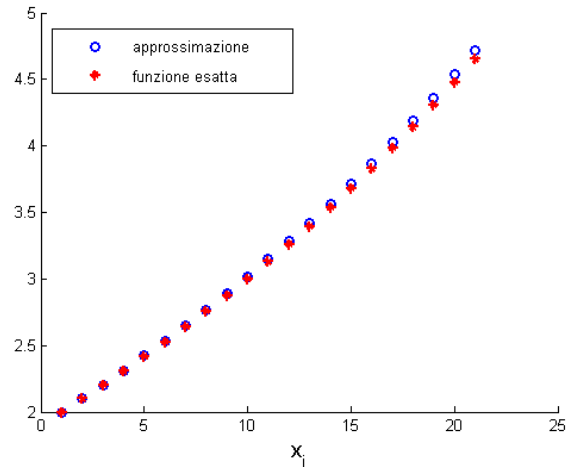


L'errore di stima cresce man mano che  $x_i$  si avvicina a 1.

Consideriamo ora il passo di discretizzazione  $h = 0.05$ . In questo caso eseguiamo 20 passi del metodo di Eulero:

$x_i$	$y_i$	$y_i^*$	$e_i$
0.00	2.0000000000000000	2.0000000000000000	0
0.05	2.1000000000000000	2.10127109637602	0.00127109637602
0.10	2.2025000000000000	2.20517091807565	0.00267091807565
0.15	2.3076250000000000	2.31183424272828	0.00420924272828
0.20	2.4155062500000000	2.42140275816017	0.00589650816017
0.25	2.5262815625000000	2.53402541668774	0.00774385418774
0.30	2.6400956406250000	2.64985880757600	0.00976316695100
0.35	2.7571004226562500	2.76906754859326	0.01196712593701
0.40	2.8774554437890600	2.89182469764127	0.01436925385221
0.45	3.0013282159785200	3.01831218549017	0.01698396951165
0.50	3.1288946267774400	3.14872127070013	0.01982664392269
0.55	3.2603393581163100	3.28325301786740	0.02291365975108
0.60	3.3958563260221300	3.42211880039051	0.02626247436838
0.65	3.5356491423232400	3.56554082901390	0.02989168669066
0.70	3.6799315994394000	3.71375270747048	0.03382110803108
0.75	3.8289281794113700	3.86700001661267	0.03807183720131
0.80	3.9828745883819400	4.02554092849247	0.04266634011053
0.85	4.1420183178010300	4.18964685192599	0.04762853412496
0.90	4.3066192336910800	4.35960311115695	0.05298387746587
0.95	4.4769501953756400	4.53570965931585	0.05875946394021
1.00	4.6532977051444200	4.71828182845905	0.06498412331463

Riducendo il passo di discretizzazione, l'errore si è ridotto di circa la metà



Infatti, **l'errore locale di troncamento** è

$$R(x_i, y(x_i); h; f) = \frac{1}{2}h^2 y''(\eta_i) \quad \eta_i \in [x_i, x_{i+1}]$$

cioè

$$R(x_i, y(x_i); h; f) = \frac{1}{2}h^2 e^{\eta_i}$$

## Esercizio

Scrivere la funzione matlab **eulero.m** che implementi il metodo di Eulero esplicito. La funzione ha come parametri di input il punto iniziale  $(x_0, y_0)$  la funzione  $f(x, y(x))$ , il passo di discretizzazione  $h$  e il numero di iterazioni da eseguire  $n\_step$ . L'output è una matrice  $T$  le cui righe contengono rispettivamente i nodi e l'approssimazione prodotta dal metodo nei nodi.

Usare la funzione per risolvere l'esercizio precedente usando i passi  $h = 0.1, 0.05, 0.025, 0.01$ . Si calcolino gli errori commessi confrontando l'approssimazione prodotta con la soluzione esatta e verificare che l'errore massimo nell'intervallo di stima è una funzione lineare rispetto ad  $h$ .

```

function [T]=eulero(x0,y0,fun,h,n_step)
%
% [T]= eulero(x0,y0,fun,h,n_step)
% Soluzione di un' equazione differenziale
% del primo ordine con il metodo di Eulero
%
% Dati di input:
% x0, y0: condizione iniziale
% n: numero dei passi da eseguire
% h: passo di discretizzazione
% fun: espressione di f(x,y)
%
% OUTPUT
% T = matrice di dimensione 2xn_step
%     la prima riga contiene il vettore dei nodi,
%     la seconda riga contiene il vettore delle approssimazioni
%     della soluzione nei nodi
%

```

```
xi(1) = x0;  
yi(1) = y0;  
  
for i = 2:n_step+1  
    yi(i) = yi(i-1) + h*fun(xi(i-1),yi(i-1));  
    xi(i) = x0 + (i-1)*h;  
end  
T = [xi;yi];
```

## Script matlab

```
y_true = @(x)[exp(x)+x+1];
x0 = 0;
y0 = 2;
fun = @(x,y)[abs(y-x)];
intervallo = 1;
h = [0.1 0.05 0.025 0.01];
color = ['b','r','g','k'];
EH = [];
figure,
for i = 1:length(h)
    n_step = 1/h(i);
    [T]=eulero(x0,y0,fun,h(i),n_step);
    err = abs(T(2,:)-y_true(T(1,:)));
    EH = [EH max(err)];
    hold on,
    plot(T(1,:),err,color(i))
    xlabel('x_i')
    ylabel('ERRORE')
end
legend('0.1','0.05','0.025','0.01')
figure, loglog(h,EH)
xlabel('h')
title('grafico di e_h')
```



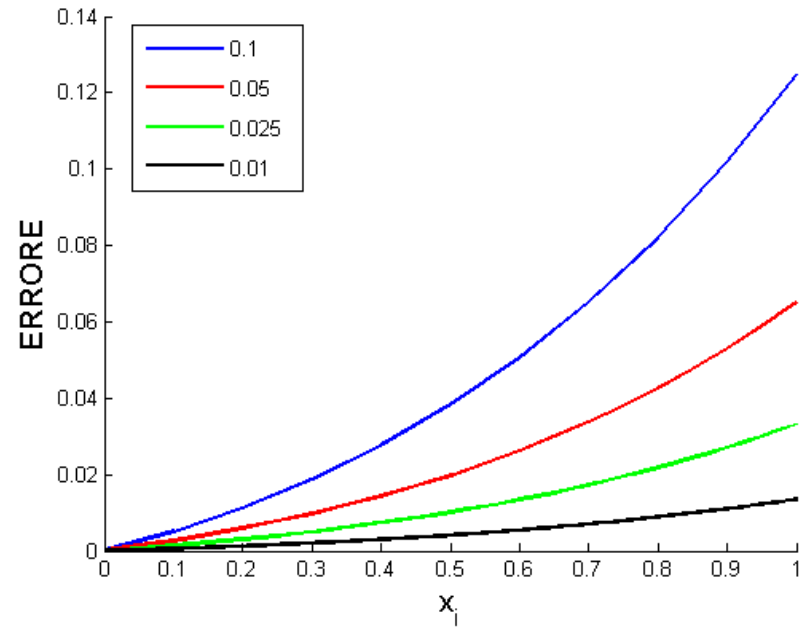


Grafico dell'errore di approssimazione del metodo di Eulero al variare del passo di discretizzazione  $h$ .

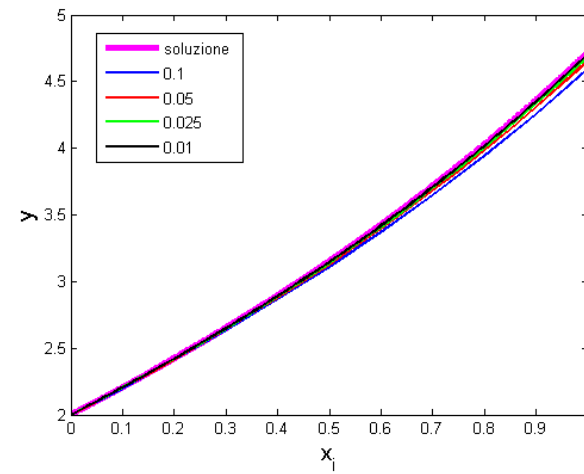
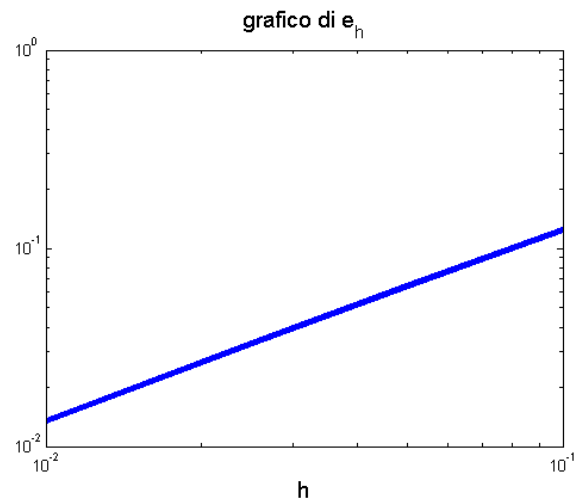


Grafico del massimo dell'errore di approssimazione al variare del passo di discretizzazione  $h$  e confronto delle soluzioni prodotte.

# Metodi di ordine superiore al primo

Sono basati sull'uso dello sviluppo in serie di Taylor di  $y(x)$  arrestato ad un certo ordine

**Sviluppo in serie di Taylor** di **ordine**  $m$  ( $y \in C^{m+1}[x_0, x_n]$ )

$$y(x_{i+1}) = y(x_i) + h \sum_{k=1}^m \frac{y^{(k)}(x_i)}{k!} h^{k-1} + \frac{h^{m+1}}{(m+1)!} y^{(m+1)}(\tau_i) \quad \tau_i \in (x_i, x_{i+1})$$

**Schema di Taylor di ordine**  $m$

Trascurando l'errore e sostituendo i valori approssimati si ottiene l'algoritmo

$$y_{i+1} = y_i + h \sum_{k=1}^m \frac{f^{(k-1)}(x_i, y_i)}{k!} h^{k-1}$$

**Errore locale di troncamento:**

$$R(x_i, y(x_i); h; f) = \frac{h^{m+1}}{(m+1)!} y^{(m+1)}(\tau_i)$$

Le derivate della  $f$  possono essere calcolate ricorsivamente da

$$\begin{cases} f^{(k)}(x, y(x)) = f_x^{(k-1)}(x, y(x)) + f_y^{(k-1)}(x, y(x)) f(x, y(x)) & k \geq 1 \\ f^{(0)}(x, y(x)) := f(x, y(x)) \end{cases}$$

## Metodi di ordine superiore al primo: Runge-Kutta

Si ottengono muovendosi lungo una **retta** che ha come coefficiente angolare una **combinazione lineare** di valori assunti da  $f(x, y)$  in punti opportuni dell'intervallo  $[x_i, x_i + h]$ .

### Metodi di Runge-Kutta del secondo ordine

#### Relazione esatta

$$y(x+h) = y(x) + h \Phi(x, y; h; f) + R(x, y; h; f) \rightarrow R(x, y; h; f) = O(h^{2+1})$$

$$\text{con } \begin{cases} \Phi(x, y; h; f) = a_1 k_1(x, y) + a_2 k_2(x, y) \\ k_1(x, y) = f(x, y) \\ k_2(x, y) = f(x + \lambda h, y + \lambda h k_1(x, y)) \end{cases}$$

## Sviluppo in serie di Taylor

$$\begin{aligned}y(x+h) &= y(x) + y'(x)h + \frac{1}{2}y''(x)h^2 + O(h^3) = \\ &= y(x) + \underbrace{f(x,y)h + \frac{1}{2}(f_x(x,y) + f_y(x,y)f(x,y))h^2}_{h\Phi(x;y;h;f)} + O(h^3)\end{aligned}$$

Si osserva che la funzione incremento così definita dipende anche da  $f_x$  e  $f_y$ . Tuttavia, poichè  $f(x,y) + f_x(x,y)\lambda h + f_y(x,y)\lambda h f(x,y)$  è lo sviluppo in serie di Taylor attorno al punto  $(x,y)$  della funzione

$$f(x + \lambda h, y + \lambda h f(x, y)),$$

si ha

$$k_2(x, y) = f(x, y) + f_x(x, y)\lambda h + f_y(x, y)\lambda h f(x, y) + O(h^2)$$

$$\begin{aligned}
R(x, y; h; f) &= y(x+h) - y(x) - h \Phi(x, y; h; f) = \\
&= h f(x, y) + \frac{h^2}{2} \left( f_x(x, y) + f_y(x, y) f(x, y) \right) + O(h^3) - \\
&- h \left( a_1 f(x, y) + \right. \\
&\quad \left. a_2 \left( f(x, y) + f_x(x, y) \lambda h + f_y(x, y) \lambda h f(x, y) + O(h^2) \right) \right) = \\
&= h f(x, y) [1 - a_1 - a_2] + h^2 f_x(x, y) \left( \frac{1}{2} - a_2 \lambda \right) + \\
&+ h^2 f_y(x, y) f(x, y) \left( \frac{1}{2} - a_2 \lambda \right) + O(h^3)
\end{aligned}$$

Affinché il metodo sia del **secondo ordine** devono essere **nulli** i termini in  $h$  e  $h^2$

$$\begin{cases} 1 - a_1 - a_2 = 0 \\ \frac{1}{2} - a_2 \lambda = 0 \end{cases}$$

che è un **sistema di 2 equazioni in 2 incognite**  $\Rightarrow$  ammette **infinite soluzioni**

# Esempio: metodo di Heun

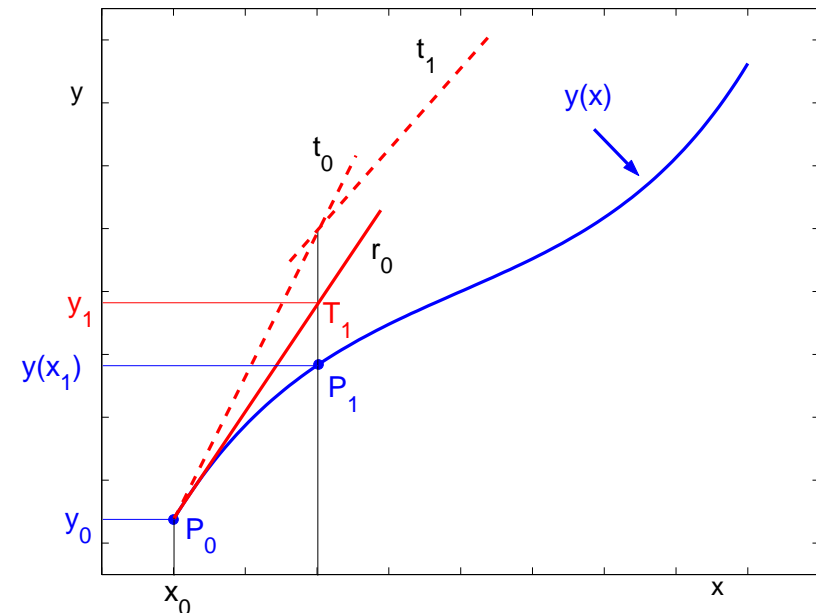
Il **metodo di Heun** si ottiene per una particolare scelta dei tre parametri  $a_1, a_2, \lambda$

$$a_1 = a_2 = \frac{1}{2} \quad \lambda = 1$$

$$\begin{cases} y_{i+1} = y_i + \frac{h}{2} (f(x_i, y_i) + f(x_i + h, y_i + hf(x_i, y_i))) & i = 0, 1, \dots, n \\ y_0 = y(x_0) \end{cases}$$

**Errore locale di truncamento:**

$$R(x, y; h; f) = O(h^3) \\ \Rightarrow \text{Secondo ordine}$$





## Esempio: metodo di Eulero modificato

Il **metodo di Eulero modificato** si ottiene per un'altra particolare scelta dei tre parametri  $a_1, a_2, \lambda$

$$a_1 = 0 \quad a_2 = 1 \quad \lambda = \frac{1}{2}$$

$$\begin{cases} y_{i+1} = y_i + h \left( f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} f(x_i, y_i)\right) \right) & i = 0, 1, \dots, n \\ y_0 = y(x_0) \end{cases}$$

# Metodo di Runge-Kutta del 4° ordine

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) & i = 0, 1, \dots, n \\ y_0 = y(x_0) \end{cases}$$

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \\ k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right) \\ k_4 = f(x_i + h, y_i + hk_3) \end{cases}$$

**Errore locale di truncamento:**

$$R(x, y; h; f) = O(h^5) \Rightarrow \text{Quarto ordine}$$

# Metodo di Runge-Kutta a r stadi

Più in generale è possibile costruire metodi di **Runge Kutta a r stadi**

$$\begin{cases} y_{i+1} = y_i + h \sum_{l=1}^r a_l k_l & i = 0, 1, \dots, n \\ y_0 = y(x_0) \end{cases}$$

con

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_l = f\left(x_i + \lambda_l h, y_i + h \sum_{j=0}^{l-1} b_{lj} k_j\right) \end{cases}$$

che risultano consistenti se  $\sum_{l=1}^r a_l = 1$ .

Inoltre, l'ordine del metodo è

$$\begin{aligned} p &= r, & r &= 1, \dots, 4 \\ p &= r - 1, & r &= 5, \dots, 7 \\ p &= r - 2, & r &\geq 8 \end{aligned}$$

# Esempio

Problema di Cauchy:

$$\begin{cases} y'(x) = y(x) - x & x \in [0, 2] \\ y(0) = 2 \end{cases} \rightarrow \text{Soluzione esatta: } y(x) = e^x + x + 1$$

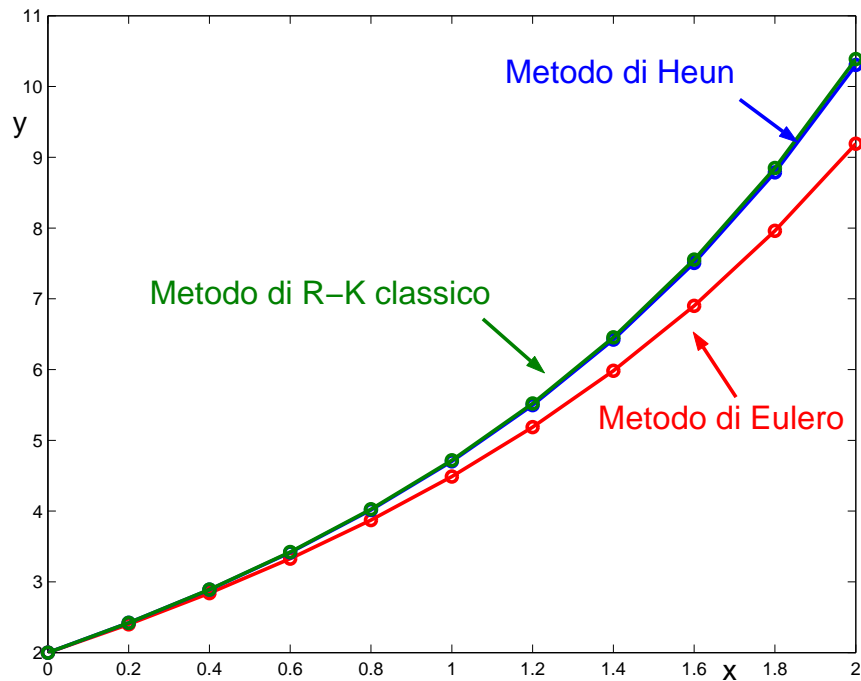


Grafico della soluzione

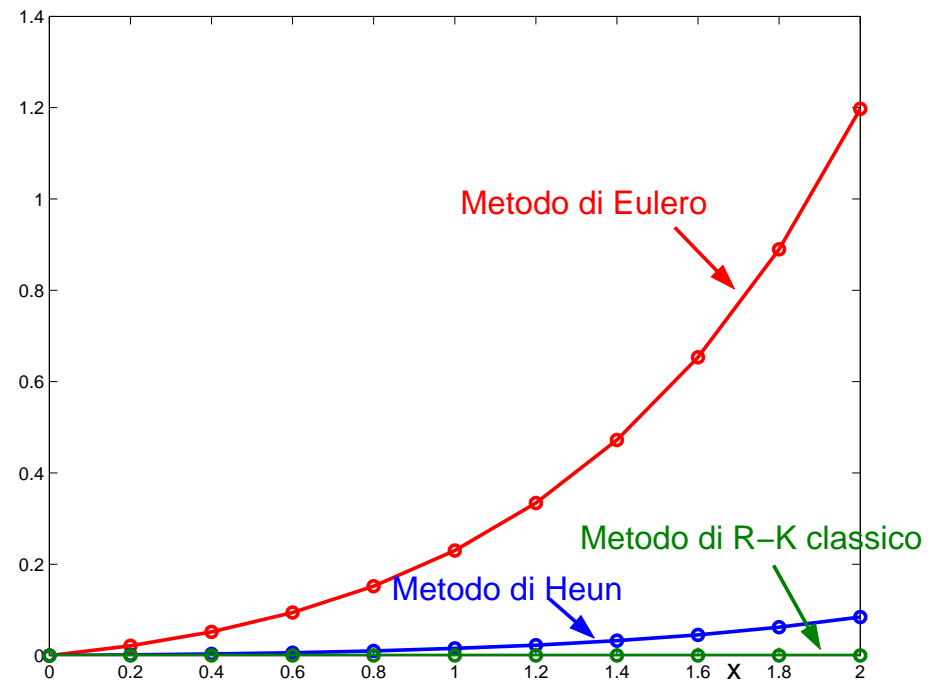


Grafico dell'errore

# Metodi one-step espliciti

I metodi di **Eulero**, **Heun** e **Runge-Kutta classico** sono tutti **metodi one-step espliciti**, cioè metodi in cui per il calcolo di  $y_{i+1}$  si utilizza **solo** il valore approssimato  $y_i$ :

**Metodi one-step espliciti:**

$$\begin{cases} y_{i+1} = y_i + h \underbrace{\Phi(x_i, y_i; h; f)}_{\text{Funzione incremento}} & i = 0, 1, \dots, n \\ y_0 = y(x_0) \end{cases}$$

**Metodo di Eulero:**  $\Phi(x, y; h; f) = f(x, y(x))$

**Metodo di Heun:**  $\Phi(x, y; h; f) = \frac{1}{2} [f(x, y(x)) + f(x + h, y(x) + hf(x, y(x)))]$

# Errore locale di troncamento

L'**errore locale di troncamento** è definito come la **differenza** tra la **soluzione esatta**  $y(x+h)$  e l'**approssimazione** fornita dallo schema numerico quando questo utilizza i valori esatti  $y(x)$ :

$$R(x, y(x); h; f) = y(x+h) - \left( y(x) + h\Phi(x, y(x); h; f) \right)$$

Rappresenta l'**errore** dovuto al fatto di aver approssimato **localmente** la soluzione esatta con una retta.

**Errore locale unitario di troncamento:**

$$\tau(x, y(x); h; f) = \frac{R(x, y(x); h; f)}{h} = \frac{y(x+h) - y(x)}{h} - \Phi(x, y(x); h; f)$$

**Ordine del metodo:** permette di valutare quanto è **accurata** l'approssimazione ottenuta. È definito come il **massimo intero**  $p$  tale che

$$R(x, y(x); h; f) = O(h^{p+1}) \Leftrightarrow \tau(x, y(x); h; f) = O(h^p)$$

ovvero  $\exists C_p$  costante :  $C_p h^{p+1}$

- Un metodo di ordine  $p$  è **esatto** per **tutti i polinomi** fino al grado  $p$ .

# Consistenza dei metodi one-step espliciti

**Consistenza:**  $\lim_{h \rightarrow 0} \tau(x, y(x); h; f) = \lim_{h \rightarrow 0} \frac{R(x, y; h; f)}{h} = 0$

I metodi one-step espliciti si ottengono **discretizzando** una relazione **esatta** del tipo:

$$y(x+h) = y(x) + h\Phi(x, y(x); h; f) + \underbrace{R(x, y(x); h; f)}$$



**Errore locale di troncamento**

$$\begin{aligned} \lim_{h \rightarrow 0} \tau(x, y; h; f) &= \lim_{h \rightarrow 0} \left[ \frac{y(x+h) - y(x)}{h} - \Phi(x, y(x); h; f) \right] \\ &= y'(x) - \lim_{h \rightarrow 0} \Phi(x, y(x); h; f) = 0 \end{aligned}$$

⇒ Un metodo one-step esplicito è **consistente** se e solo se

$$\lim_{h \rightarrow 0} \Phi(x, y(x); h; f) = y'(x) = f(x, y(x))$$

In particolare,

- il metodo di **Eulero** è **consistente**. Infatti,

$$\lim_{h \rightarrow 0} \Phi(x, y(x); h; f) = \lim_{h \rightarrow 0} f(y, y(x)) = f(x, y(x))$$

- il metodo di **Heun** è **consistente**.

Infatti,

$$\lim_{h \rightarrow 0} \Phi(x, y(x); h; f) = \lim_{h \rightarrow 0} \frac{1}{2} (f(y, y(x)) + f(x+h, y(x)+h)) = f(x, y(x))$$



# Stabilità dei metodi one-step espliciti

**Errore globale di troncamento:**  $e_i = y(x_i) - y_i$

**Convergenza:**  $\lim_{h \rightarrow 0} \max_{0 \leq i \leq n} |e_i| = 0 \Leftrightarrow \lim_{i \rightarrow \infty} y_i = y(\bar{x}) \quad \bar{x} = x_0 + ih$

$$\begin{aligned} e_{i+1} &= y(x_{i+1}) - y_{i+1} = \\ &= (y(x_i) + h\Phi(x_i, y(x_i); h; f) + R(x_i, y(x_i); h; f)) - \\ &\quad - (y_i + h\Phi(x_i, y_i; h; f)) \end{aligned}$$

- Dalla definizione di **ordine** di un metodo si ha:

$$R(x_i, y(x_i); h; f) = O(h^{p+1}) \Rightarrow |R(x_i, y(x_i); h; f)| \leq C_p h^{p+1}$$

**Esempio:** per il **metodo di Eulero**  $|R| = |\frac{1}{2}h^2 y''(x)| \leq \frac{M}{2}h^2$ , dove  $M$  è il **massimo** di  $|y''(x)|$  nell'intervallo di integrazione  $\Rightarrow C_1 = M/2$

- Poiché  $\Phi$  è una combinazione lineare di valori di  $f$ , dalla **lipshitzianità** di  $f$  segue la lipshitzianità di  $\Phi$ :

$$|\Phi(x, y_1) - \Phi(x, y_2)| \leq L|y_1 - y_2|$$

Utilizzando le **maggiorazioni** trovate e **trascurando** gli errori di arrotondamento si ha:

$$e_{i+1} = \underbrace{(y(x_i) - y_i)}_{e_i} + h \underbrace{(\Phi(x_i, y(x_i); h; f) - \Phi(x_i, y_i; h; f))}_{\leq L|y(x_i) - y_i|} + O(h^{p+1})$$

$$\Rightarrow \begin{cases} |e_{i+1}| \leq |e_i|(1 + hL) + C_p h^{p+1} & i = 0, 1, \dots \\ e_0 = y(x_0) - y_0 = 0 \end{cases}$$

Si associa alla disuguaglianza ottenuta una **equazione alle differenze** del primo ordine:

$$\begin{cases} t_{i+1} = t_i(1 + hL) + C_p h^{p+1} & i = 0, 1, \dots \\ t_0 = 0 \end{cases} \Rightarrow t_i = C_p h^p \frac{(1 + hL)^i - 1}{L}$$

Poiché  $|e_i| \leq t_i$ ,  $i = 0, 1, \dots$ , e  $(1 + \alpha)^i < e^{i\alpha}$  per  $\alpha > 0$ , si deduce che l'**errore globale** di troncamento ha la **limitazione**

$$|e_i| \leq C_p h^p \frac{e^{L(x_i - x_0)} - 1}{L}$$

$L$ : costante di Lipschitz di  $f(x, y)$      $C_p$ : costante dipendente dal metodo one-step.

# Convergenza dei metodi one-step espliciti

Un generico metodo per risolvere numericamente un'equazione differenziale è **convergente** se è **consistente** (l'errore locale unitario di troncamento tende a zero quando  $h$  tende a zero) e **stabile** (la propagazione degli errori durante lo sviluppo del metodo resta limitata).

**Teorema.** Sia  $\Phi(x, y(x); h; f) \in C^0(D)$ ,  $D = S \times [x_0, x_0 + \beta]$ ,  $0 < h \leq \beta$ , e **lipschitziana** in  $y$ . Allora un **metodo one-step esplicito** è **convergente** se e solo se è **consistente**. Inoltre, se il metodo è di **ordine**  $p$ , si ha

$$|e_i| = |y(x_i) - y_i| \leq k \cdot h^p$$

dove  $k$  è una costante indipendente da  $i$  e da  $h$ .

# Propagazione degli errori di arrotondamento

Se indichiamo con  $\eta_{i+1}$  l'**errore di arrotondamento** che si produce nel calcolo di  $y_{i+1}$  ad ogni passo, si può scrivere

$$y_{i+1} = y_i + h\Phi(x_i, y_i; h; f) + \eta_{i+1}$$

Se  $|\eta_i| \leq \eta, \forall i$ , per l'errore globale di troncamento si ha la **limitazione**

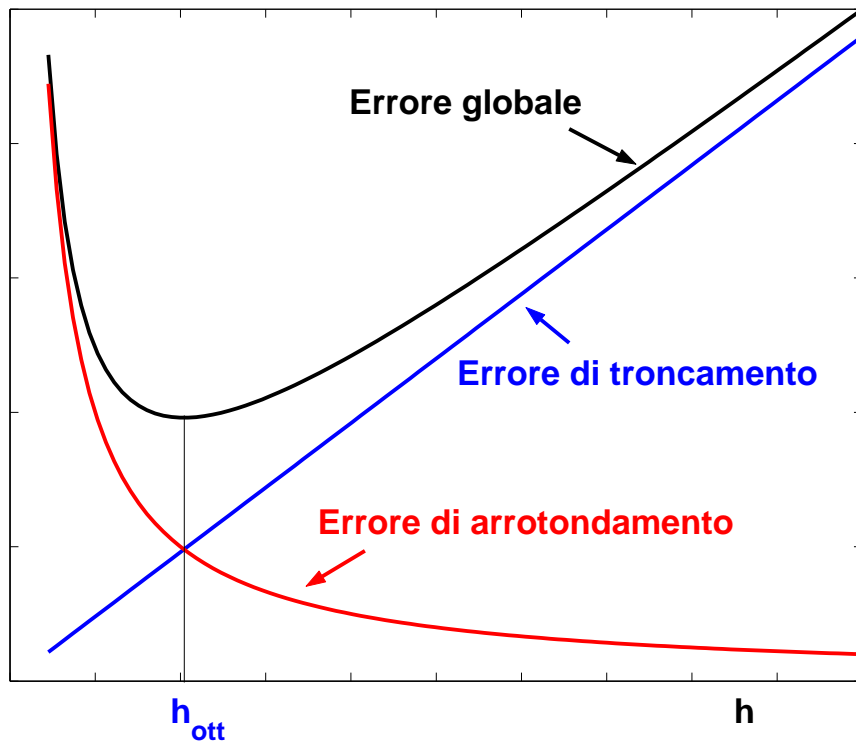
$$|e_i| \leq \frac{e^{L(x_i-x_0)} - 1}{L} \left( C_p h^p + \frac{\eta}{h} \right)$$

In particolare, per il **metodo di Eulero** si ha

$$|e_i| \leq \frac{e^{L(x_i-x_0)} - 1}{L} \left( \frac{M}{2} h + \frac{\eta}{h} \right) \quad M = \max_{x \in [x_0, x_0+\beta]} |y''(x)|$$

## Errore globale:

$$|e_i| \leq \frac{e^{L(x_i-x_0)} - 1}{L} \left( \frac{M}{2} h + \frac{\eta}{h} \right) \quad M = \max_{x \in [x_0, x_0+\beta]} |y''(x)|$$



In corrispondenza del **valore**

**ottimale**

$$h_{\text{ott}} = \sqrt{\frac{2\eta}{M}}$$

dato

dall'intersezione della retta  $\frac{M}{2}h$  e la retta  $\frac{\eta}{h}$ , l'errore di troncamento è uguale a quello di arrotondamento e la maggiorazione dell'errore ha un **minimo**:

- per  $h < h_{\text{ott}}$  predomina l'**errore di arrotondamento**;
- per  $h > h_{\text{ott}}$  predomina l'**errore di troncamento**.

# Esempio

Problema di Cauchy:  $f(x, y) = y(x)$

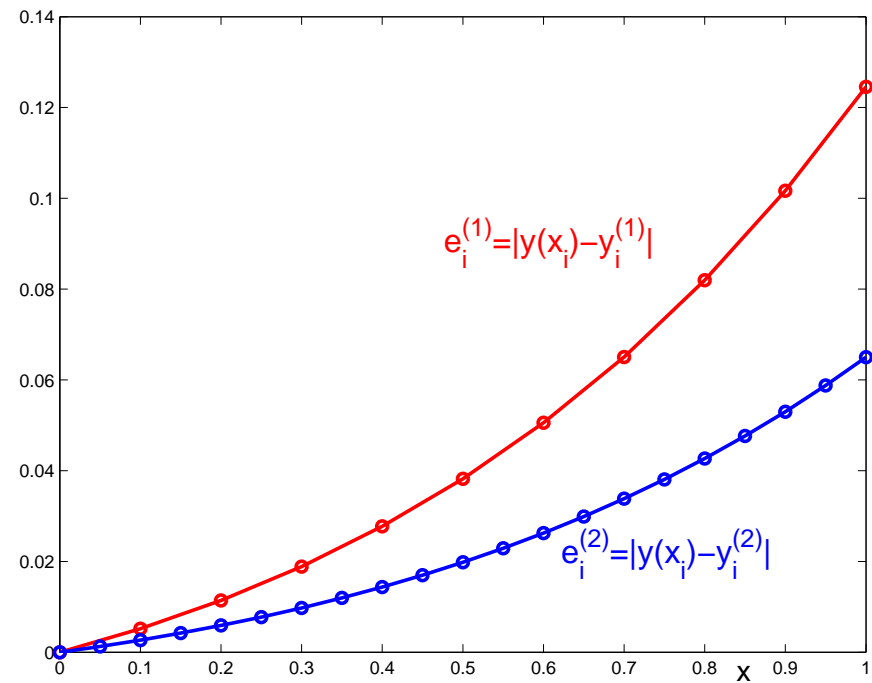
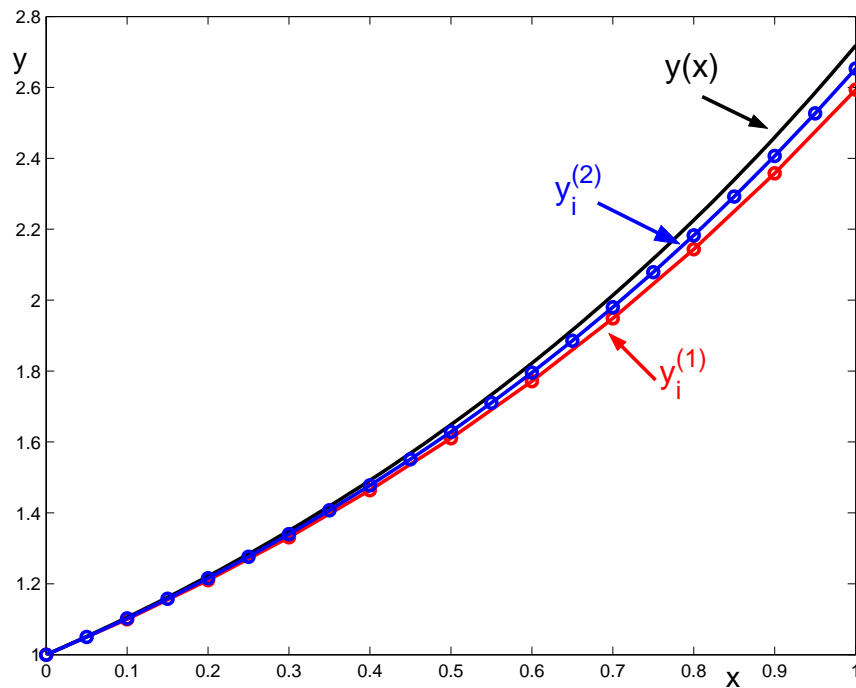
$$\begin{cases} y'(x) = y(x) & x \in [0, 1] \\ y(0) = 1 \end{cases} \quad \text{Soluzione esatta: } y(x) = e^x$$

Metodo di Eulero:

$$\begin{cases} y_{i+1} = y_i + hf(x_i, y_i) = y_i + hy_i & i = 1, 2, \dots, n \\ y_0 = 1 \end{cases}$$

1)  $x_i^{(1)} = ih_1 \quad i = 0, 1, \dots, n_1 \quad h_1 = 0.1 \quad n_1 = 10$

2)  $x_i^{(2)} = ih_2 \quad i = 0, 1, \dots, n_2 \quad h_2 = 0.05 \quad n_2 = 20$



**Nota.**  $M = \max_{x \in [0,1]} |y''(x)| = \max_{x \in [0,1]} e^x = e \simeq 2.7183$

Se  $\eta = 0.5 \cdot 10^{-14} \Rightarrow h_{\text{ott}} = \sqrt{\frac{2\eta}{M}} \simeq 6.06 \cdot 10^{-8}$

# Esercizio

(Primo esercizio della lezione)

$$\begin{cases} y' &= |y - x|, & x \in [0, 2] \\ y(0) &= 2 \end{cases}$$



## Soluzione

Ricordando che la soluzione esatta è  $y^*(x) = e^x + x + 1$ , scegliendo il passo  $h = 0.1$  e risolvendo numericamente usando il Metodo di Eulero:

$x_i$	$y_i$	$y_i^*$	$e_i =  y_i - y_i^* $
0.0	2.0000000000	2.0000000000000000	0.0000000000000000
0.1	2.2000000000	2.20517091807565	0.00517091807565
0.2	2.4100000000	2.42140275816017	0.01140275816017
0.3	2.6310000000	2.64985880757600	0.01885880757600
0.4	2.8641000000	2.89182469764127	0.02772469764127
0.5	3.1105100000	3.14872127070013	0.03821127070013
0.6	3.3715610000	3.42211880039051	0.05055780039051
0.7	3.6487171000	3.71375270747048	0.06503560747048
0.8	3.9435888100	4.02554092849247	0.08195211849247
0.9	4.2579476910	4.35960311115695	0.10165542015695
1.0	4.5937424601	4.71828182845905	0.12453936835904

Con  $h = 0.05$

$x_i$	$y_i$	$y_i^*$	$e_i$
0.00	2.000000000000000	2.000000000000000	0
0.05	2.100000000000000	2.10127109637602	0.00127109637602
0.10	2.202500000000000	2.20517091807565	0.00267091807565
0.15	2.307625000000000	2.31183424272828	0.00420924272828
0.20	2.415506250000000	2.42140275816017	0.00589650816017
0.25	2.526281562500000	2.53402541668774	0.00774385418774
0.30	2.640095640625000	2.64985880757600	0.00976316695100
0.35	2.757100422656250	2.76906754859326	0.01196712593701
0.40	2.877455443789060	2.89182469764127	0.01436925385221
0.45	3.001328215978520	3.01831218549017	0.01698396951165
0.50	3.128894626777440	3.14872127070013	0.01982664392269
0.55	3.260339358116310	3.28325301786740	0.02291365975108
0.60	3.395856326022130	3.42211880039051	0.02626247436838
0.65	3.535649142323240	3.56554082901390	0.02989168669066
0.70	3.679931599439400	3.71375270747048	0.03382110803108
0.75	3.828928179411370	3.86700001661267	0.03807183720131
0.80	3.982874588381940	4.02554092849247	0.04266634011053
0.85	4.142018317801030	4.18964685192599	0.04762853412496
0.90	4.306619233691080	4.35960311115695	0.05298387746587
0.95	4.476950195375640	4.53570965931585	0.05875946394021
1.00	4.653297705144420	4.71828182845905	0.06498412331463

Riducendo il passo di discretizzazione, l'errore si è ridotto di circa la metà .

Vogliamo ora stimare con quale valore del passo  $h$  l'errore di stima della soluzione del PC nel punto  $x = 1$  è sicuramente minore di  $0.5 \cdot 10^{-3}$ .

Supponendo di trascurare gli errore di arrotondamento, per la stima dell'errore globale di troncamento nel punto  $x_i$  si può usare il seguente risultato

$$|e_i| < \frac{M}{2} h \frac{e^{L(x_i-x_0)} - 1}{L},$$

con  $M = \max_{x \in [0,1]} |y''(x)|$ ,  $L$  la costante di Lipschitz della funzione  $f(x, y)$ .

Nel nostro caso,  $x_i = 1$ ,  $x_0 = 0$ ,  $L = 1$  e  $y''(x) = e^x$  da cui  $M = e$ .

Ne risulta che

$$e_i < \frac{e}{2} h \frac{e-1}{1} < 0.5 \cdot 10^{-3}$$



$$h < \frac{2 \cdot 0.5 \cdot 10^{-3}}{e(e-1)} = \frac{10^{-3}}{e(e-1)} \approx 2.141 \cdot 10^{-4} = 0.000214.$$

Infatti, scegliendo  $h = 0.0002$ , l'errore su  $y(1)$  è  $0.2718 \cdot 10^{-3}$ .

Verificare questo risultato usando la funzione *eulero.m*

```
f=@(x,y)[abs(y-x)];  
y_true = @(x)[exp(x)+x+1];  
h = 0.0002;  
n_step = 1/0.0002  
n_step =  
    5000  
[T]=eulero(0,2,f,h,n_step);  
y_true_val = y_true(T(1,:));  
ERR = abs(y_true_val-T(2,:));  
ERR(end)  
ans =  
    2.717783571872801e-004
```

Utilizziamo ora il **metodo di Heun**.

Essendo un metodo del secondo ordine, l'errore di troncamento globale è

$$|e_i| < C_2 h^2 \frac{e^{L(x_i - x_0)} - 1}{L},$$

con  $C_2$  opportuna costante.

```

function [T]=heun(x0,y0,fun,h,n_step)
%
% [T]= heun(x0,y0,fun,h,n_step)
% Soluzione di unequazione differenziale
% del primo ordine con il metodo di Heun
%
% Dati di input:
% x0, y0: condizione iniziale
% n: numero dei passi da eseguire
% h: passo di discretizzazione
% fun: espressione di f(x,y)
%
% OUTPUT
% T = matrice di dimensione 2xn_step
%     la prima riga contiene il vettore dei nodi,
%     la seconda riga contiene il vettore delle approssimazioni
%     della soluzione nei nodi
%

```

```
xi(1) = x0;  
yi(1) = y0;  
  
for i = 2:n_step+1  
    fi = fun(xi(i-1),yi(i-1));  
    yi(i) = yi(i-1) + h/2*(fi+ fun(xi(i-1)+h,yi(i-1)+h*fi));  
    xi(i) = x0 + (i-1)*h;  
end  
T = [xi;yi];
```



Usando lo stesso passo  $h = 0.0002$  si ha

### Dal Command Window

```
f=@(x,y)[abs(y-x)];  
y_true = @(x)[exp(x)+x+1];  
h = 0.0002;  
n_step = 1/h;  
[T]=heun(0,2,f,h,n_step);  
y_true_val = y_true(T(1,:));  
ERR = abs(y_true_val-T(2,:));  
ERR(end)  
ans =
```

```
1.811916483518417e-008
```

L'errore in corrispondenza del punto  $x = 1$  si è **ridotto di ben 4 ordini di grandezza**.

In questo caso, invece, con

```
>> h = 0.025
```

```
h =
```

```
0.0250000000000000
```

```
>> n_step = 1/h
```

```
n_step =
```

```
40
```

```
>> [T]=heun(0,2,f,h,n_step);
```

```
>> N=g(T(1,:));
```

```
>> E = abs(N-T(2,:));
```

```
>> E(end)
```

```
ans =
```

```
2.778840880699462e-004
```

si ha un errore confrontabile con quello ottenuto usando il metodo di Eulero.

xi	yi	yi*	ei =  yi*-yi
0	2.00000000000000	2.00000000000000	0
0.02500000000000	2.05031250000000	2.05031512052443	0.00000262052443
0.05000000000000	2.10126572265625	2.10127109637602	0.00000537371977
0.07500000000000	2.15287588626099	2.15288415088463	0.00000826462365
0.10000000000000	2.20515961963197	2.20517091807565	0.00001129844368
0.12500000000000	2.25813397250390	2.25814845306683	0.00001448056292
0.15000000000000	2.31181642618291	2.31183424272828	0.00001781654538
0.17500000000000	2.36622490447066	2.36624621661236	0.00002131214170
0.20000000000000	2.42137778486507	2.42140275816017	0.00002497329509
0.22500000000000	2.47729391004447	2.47732271619186	0.00002880614739
0.25000000000000	2.53399259964247	2.53402541668774	0.00003281704527
0.27500000000000	2.59149366232092	2.59153067486762	0.00003701254670
0.30000000000000	2.64981740814842	2.64985880757600	0.00004139942758
0.32500000000000	2.70898466129218	2.70903064598075	0.00004598468857
0.35000000000000	2.76901677303114	2.76906754859326	0.00005077556212
0.37500000000000	2.82993563509849	2.82999141461820	0.00005577951971
0.40000000000000	2.89176369336192	2.89182469764127	0.00006100427935
0.42500000000000	2.95452396185014	2.95459041966338	0.00006645781324
0.45000000000000	3.01824003713447	3.01831218549017	0.00007214835570
0.47500000000000	3.08293611307444	3.08301419748578	0.00007808441134
0.50000000000000	3.14863699593664	3.14872127070013	0.00008427476349
0.52500000000000	3.21536811989628	3.21545884837909	0.00009072848281
0.55000000000000	3.28315556293116	3.28325301786740	0.00009745493624
0.57500000000000	3.35202606311785	3.35213052691404	0.00010446379619
0.60000000000000	3.42200703534052	3.42211880039051	0.00011176504999
0.62500000000000	3.49312658842258	3.49324595743222	0.00011936900964

0.6500000000000000	3.56541354269203	3.56554082901390	0.00012728632187
0.6750000000000000	3.63889744799142	3.63903297596985	0.00013552797843
0.7000000000000000	3.71360860214370	3.71375270747048	0.00014410532678
0.7250000000000000	3.78957806988546	3.78973109996649	0.00015303008102
0.7500000000000000	3.86683770227944	3.86700001661267	0.00016231433324
0.7750000000000000	3.94542015661839	3.94559212718344	0.00017197056506
0.8000000000000000	4.02535891683279	4.02554092849247	0.00018201165968
0.8250000000000000	4.10668831441512	4.10688076532930	0.00019245091418
0.8500000000000000	4.18944354987375	4.18964685192599	0.00020330205224
0.8750000000000000	4.27366071472993	4.27387529396710	0.00021457923717
0.9000000000000000	4.35937681407153	4.35960311115695	0.00022629708542
0.9250000000000000	4.44662978967772	4.44686826035815	0.00023847068043
0.9500000000000000	4.53545854372894	4.53570965931585	0.00025111558691
0.9750000000000000	4.62590296311707	4.62616721098261	0.00026424786553
1.0000000000000000	4.71800394437098	4.71828182845905	0.00027788408807

$$k1 = f(x_0, y_0) = |y_0 - x_0| = |2 - 0| = 2$$

$$k2 = f(x_0+h, y_0+h \cdot k1) = |y_0+h \cdot k1 - x_0-h| = |2+0.025 \cdot 2 - 0.025| = 2.025$$

$$y_1 = y_0 + \frac{h}{2}(k1 + k2) = 2 + 0.025/2 \cdot (2 + 2.025) = 2.0503125$$

$$x_1 = x_0 + h = 0.025$$

$$k1 = f(x_1, y_1) = |2.0503125 - 0.025| = 2.0253125$$

$$k2 = f(x_1 + h, y_1 + h \cdot k1) = |y_1 + h \cdot k1 - x_1 - h| =$$

$$= |2.0503125 + 0.025 \cdot 2.0253125 - 0.025 - 0.025| = 2.0509453125$$

$$y_2 = y_1 + \frac{h}{2}(k1 + k2) =$$

$$= 2.0503125 + 0.025/2 \cdot (2.0253125 + 2.0509453125) = 2.10126572265625$$

Aumentando l'ordine del metodo numerico, per esempio scegliendo un **Runge-Kutta del 4<sup>o</sup> ordine**

```
function [T]=RK4(x0,y0,fun,h,n_step)
%
% [T]= RK4(x0,y0,fun,h,n_step)
% Soluzione di unequazione differenziale
% del primo ordine con il metodo di Runge-Kutta di ordine 4
%
% INPUT:
% x0, y0: condizione iniziale
% n: numero dei passi da eseguire
% h: passo di discretizzazione
% fun: espressione di f(x,y)
% OUTPUT
% T = matrice di dimensione 2xn_step
%     la prima riga contiene il vettore dei nodi,
%     la seconda riga contiene il vettore delle approssimazioni
%     della soluzione nei nodi
```

```
xi(1) = x0;
yi(1) = y0;
for i = 2:n_step+1
    k1 = fun(xi(i-1),yi(i-1));
    k2 = fun(xi(i-1)+h/2,yi(i-1)+h/2*k1);
    k3 = fun(xi(i-1)+h/2,yi(i-1)+h/2*k2);
    k4 = fun(xi(i-1)+h,yi(i-1)+h*k3);
    yi(i) = yi(i-1) + h/6*(k1+2*k2+2*k3+k4);
    xi(i) = x0 + (i-1)*h;
end
T = [xi;yi];
```



e usando  $h = 0.025$ , si ha

```
>> [T]=RK4(0,2,f,h,n_step);  
>> N=g(T(1,:));  
>> E = abs(N-T(2,:));  
>> E(end)
```

ans =

```
8.666188655581664e-009
```

L'errore più piccolo di quasi 5 ordini di grandezza.

Scegliendo  $h = 0.5$  (si eseguono solo due passi del metodo!!!)

```
>> h = 0.5
```

```
h =
```

```
0.5000000000000000
```

```
>> n_step = 1/h
```

```
n_step =
```

```
2
```

```
>> [T]=RK4(0,2,f,h,n_step);
```

```
>> N=g(T(1,:));
```

```
>> E = abs(N-T(2,:));
```

```
>> E(end)
```

```
ans =
```

```
9.356370527955349e-004
```

xi	yi	yi*	ei =  yi*-yi
0	2.0000000000000000	2.0000000000000000	0
0.5000000000000000	3.1484375000000000	3.14872127070013	0.00028377070013
1.0000000000000000	4.71734619140625	4.71828182845905	0.00093563705280

$$k1 = f(x_0, y_0) = 2$$

$$k2 = f(x_0 + h/2, y_0 + h/2 \cdot k1) = |2 + 0.5/2 \cdot 2 - 0.5/2| = 2.25$$

$$k3 = f(x_0 + h/2, y_0 + h/2 \cdot k2) = |2 + 0.5/2 \cdot 2.25 - 0.5/2| = 2.3125$$

$$k4 = f(x_0 + h, y_0 + h \cdot k3) = |2 + 0.5 \cdot 2.3125 - 0.5| = 2.65625$$

$$y_1 = y_0 + h/6 \cdot (k1 + 2 \cdot k2 + 2 \cdot k3 + k4) =$$

$$= 2 + 0.5/6 \cdot (2 + 2 \cdot 2.25 + 2 \cdot 2.3125 + 2.65625) = 3.1484375$$

$$x_1 = x_0 + h = 0.5$$

$$k_1 = f(x_1, y_1) = |3.1484375 - 0.5| = 2.6484375$$

$$k_2 = f(x_1 + h/2, y_1 + h/2 \cdot k_1) = |3.1484375 + 0.5/2 \cdot 2.6484375 - 0.5 - 0.5/2| = 3.060546875$$

$$k_3 = f(x_1 + h/2, y_1 + h/2 \cdot k_2) = |3.1484375 + 0.5/2 \cdot 3.060546875 - 0.75| = 3.16357421875$$

$$k_4 = f(x_1 + h, y_1 + h \cdot k_3) = |3.1484375 + 0.5 \cdot 3.16357421875 - 1| = 3.730224609375$$

$$y_2 = y_1 + h/6 * (k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) =$$

$$3.1484375 + 0.5/6 \cdot (2.6484375 + 2 \cdot 3.060546875 + 2 \cdot 3.16357421875 + 3.730224609375) =$$

$$= 4.71734619140625$$

$$x_2 = x_0 + 2 \cdot h = 1$$

## Esercizio

Si integri la seguente equazione

$$y' = 3y - 4e^{-x}, \quad y(0) = 1$$

nell'intervallo  $[0, 10]$ , usando il metodo di Runge-Kutta del quarto ordine con passo di discretizzazione  $h = 0.1$ .

Si confronti il risultato con la soluzione analitica  $y = e^{-x}$ .

Usando la funzione *RK4.m* si ha

```
>> F =@(x,y) [3*y - 4*exp(-x)];  
>> [T] = RK4(0,1,f,0.1,100);  
>> T(:,1:20:end)'  
0.0000e+000  1.0000e+000  
2.0000e+000  1.3250e-001  
4.0000e+000 -1.1237e+000  
6.0000e+000 -4.6056e+002  
8.0000e+000 -1.8575e+005  
1.0000e+001 -7.4912e+007
```

Il comportamento della soluzione ottenuta è "strano". Infatti, guardando alla soluzione analitica, ci si aspetterebbe di avere valori monotoni decrescenti delle  $y_i$ , che, invece, prima decrescono e poi crescono in valore assoluto (addirittura diventano negative!).

La soluzione generale dell'equazione differenziale è  $y(x) = Ce^{3x} + e^{-x}$  (si può verificare per sostituzione).

La condizione iniziale  $y(0) = 1$  implica  $C = 0$  e quindi la soluzione è proprio  $y = e^{-x}$ .

La divergenza della soluzione prodotta è dovuta alla propagazione degli errori di arrotondamento. Infatti, supponendo che la condizione iniziale sia affetta da un piccolo errore  $\epsilon$ , cioè  $y(0) = 1 + \epsilon$ , la soluzione analitica dell'equazione differenziale diventa

$$y(x) = \epsilon e^{3x} + e^{-x}.$$

Quando  $x$  cresce, il termine dipendente da  $\epsilon$  diventa dominante causando l'instabilità del metodo (sensibilità della soluzione alla condizione iniziale).

# Metodi impliciti

Un altro modo di derivare metodi numerici per la soluzione di equazioni differenziali ordinarie è :

$$\underbrace{\int_{x_i}^{x_i+h} y'(x) dx}_{y(x_i+h) - y(x_i)} = \int_{x_i}^{x_i+h} f(x, y(x)) dx$$

e valutare l'integrale al secondo membro con una formula di quadratura usando un numero di nodi  $N$  con passo inferiore ad  $h$ ,

$$\Rightarrow y(x_i + h) - y(x_i) = \sum_{r=1}^N c_r f(x_{i+h-l_r}, y(x_{i+h-l_r})) + R(x_i, y(x_i); h; f)$$

$\Rightarrow y_{i+h}$  è una combinazione lineare dei valori di  $f$  nei nodi e di  $y_i$

**Esempio:** usando la formula del trapezio

$$y_{i+1} = y_i + \frac{h}{2} \left( f(x_i, y_i) + f(x_{i+1}, y_{i+1}) \right) \quad \text{Metodo di Crank-Nicolson}$$

(metodo implicito del secondo ordine)



# Metodi predictor-corrector

La soluzione viene approssimata con un metodo iterativo (**corrector**) in cui l' approssimazione iniziale è ottenuta con un metodo esplicito (**predictor**). Per esempio,

$$\left\{ \begin{array}{l} \text{Predictor} : y_{i+1} = y_i + hf(t_i, y_i) \quad i \geq 0 \\ \text{Corrector} : y_{i+1}^{(r)} = y_i + \frac{h}{2} \left( f(t_{i+1}, y_{i+1}^{(r-1)}) + f(t_i, y_i) \right) \quad 1 \leq r \leq N \\ y_{i+1} = y_{i+1}^{(N)} \end{array} \right.$$

in cui è stato usato il *metodo di Eulero esplicito* come *predittore* e il *metodo di Crank-Nicolson* come *correttore*.

## Oss:

1. La correzione si può iterare  $N$  volte, con  $N$  scelto a priori oppure determinato usando un opportuno criterio di arresto.
2. Il metodo converge se  $h < \frac{2}{L}$ , con  $L$  costante di Lipschitz di  $f$ .
3. Se  $L$  è grande (*problemi stiff*) la limitazione su  $h$  può essere troppo restrittiva
4. In generale, l'ordine  $p$  di un metodo *predictor-corrector* è  $p = \min\{p_P + 1, p_C\}$ , con  $p_P$  e  $p_C$  gli ordini rispettivamente del predittore e del correttore.



# Soluzione numerica di sistemi di equazioni differenziali

$$\begin{cases} Y'(x) = F(x, Y(x)) \\ Y(x_0) = Y_0 \end{cases} \quad \text{dove} \quad \begin{aligned} Y(x) &= [y_1(x), y_2(x), \dots, y_n(x)]^T \\ Y_0 &= [y_1(x_0), y_2(x_0), \dots, y_n(x_0)]^T \\ F(x, Y(x)) &= [f_1(x, Y(x)), \dots, f_n(x, Y(x))]^T \end{aligned}$$

## Metodo di Eulero:

$$\begin{cases} Y_{i+1} = Y_i + hF(x_i, Y_i) \\ \quad \quad \quad (i = 0, 1, \dots, n) \\ Y_0 = Y(x_0) \end{cases} \quad \text{dove} \quad \begin{aligned} Y_0(x_0) &= [y_1(x_0), y_2(x_0), \dots, y_n(x_0)]^T \\ Y_i &= [y_{1i}, y_{2i}, \dots, y_{ni}]^T \\ F(x_i, Y_i) &= [f_1(x_i, Y_i), \dots, f_n(x_i, Y_i)]^T \end{aligned}$$

Caso **particolare**:  $n = 2$

$$\begin{cases} y'(x) = f(x, y(x), z(x)) \\ z'(x) = g(x, y(x), z(x)) \\ y(x_0) = y_0 \\ z(x_0) = z_0 \end{cases} \quad \rightarrow \quad \begin{cases} y_{i+1} = y_i + hf(x_i, y_i, z_i) \\ z_{i+1} = z_i + hg(x_i, y_i, z_i) \\ y_0 = y(x_0) \\ z_0 = z(x_0) \end{cases} \quad i = 0, 1, \dots, n$$

## Metodo di Heun:

$$\begin{cases} Y_{i+1} = Y_i + \frac{h}{2}[F(x_i, Y_i) + F(x_i + h, Y_i + hF(x_i, Y_i))] \\ Y_0 = Y(x_0) \end{cases} \quad i = 0, 1, \dots, n$$

dove  $Y_0 = [y_1(x_0), y_2(x_0), \dots, y_n(x_0)]^T$   
 $Y_i = [y_{1i}, y_{2i}, \dots, y_{ni}]^T$   
 $F(x_i, Y_i) = [f_1(x_i, Y_i), \dots, f_n(x_i, Y_i)]^T$

Caso **particolare**:  $n = 2$   $\begin{cases} y'(x) = f(x, y(x), z(x)) \\ z'(x) = g(x, y(x), z(x)) \\ y(x_0) = y_0 \\ z(x_0) = z_0 \end{cases}$

$$\begin{cases} y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i, z_i) + f(x_i + h, y_i + hf(x_i, y_i, z_i), z_i + hg(x_i, y_i, z_i))] \\ z_{i+1} = z_i + \frac{h}{2}[g(x_i, y_i, z_i) + g(x_i + h, y_i + hf(x_i, y_i, z_i), z_i + hg(x_i, y_i, z_i))] \\ y_0 = y(x_0) \\ z_0 = z(x_0) \end{cases} \quad i = 0, 1, \dots, n$$

**Metodo di Runge-Kutta classico (IV ordine):**

$$\left\{ \begin{array}{l} Y_{i+1} = Y_i + \frac{h}{6}[K_1 + 2K_2 + 2K_3 + K_4] \\ Y_0 = Y(x_0) \end{array} \right. \quad i = 0, 1, \dots, n \quad \text{dove} \quad \left\{ \begin{array}{l} K_1 = F(x_i, Y_i) \\ K_2 = F\left(x_i + \frac{h}{2}, Y_i + \frac{h}{2}K_1\right) \\ K_3 = F\left(x_i + \frac{h}{2}, Y_i + \frac{h}{2}K_2\right) \\ K_4 = F(x_i + h, Y_i + hK_3) \end{array} \right.$$

Caso **particolare**:  $n = 2$   $\left\{ \begin{array}{l} y'(x) = f(x, y(x), z(x)) \\ z'(x) = g(x, y(x), z(x)) \\ y(x_0) = y_0 \\ z(x_0) = z_0 \end{array} \right.$

$$\left\{ \begin{array}{l} y_{i+1} = y_i + \frac{h}{6}[k_1 + 2k_2 + 2k_3 + k_4] \\ y_0 = y(x_0) \end{array} \right. \quad \left\{ \begin{array}{l} z_{i+1} = z_i + \frac{h}{6}[t_1 + 2t_2 + 2t_3 + t_4] \\ z_0 = z(x_0) \end{array} \right. \quad (i = 0, 1, \dots, n)$$

$$\left\{ \begin{array}{l} k_1 = f(x_i, y_i, z_i) \\ k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1, z_i + \frac{h}{2}t_1\right) \\ k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2, z_i + \frac{h}{2}t_2\right) \\ k_4 = f(x_i + h, y_i + hk_3, z_i + ht_3) \end{array} \right. \quad \left\{ \begin{array}{l} t_1 = g(x_i, y_i, z_i) \\ t_2 = g\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1, z_i + \frac{h}{2}t_1\right) \\ t_3 = g\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2, z_i + \frac{h}{2}t_2\right) \\ t_4 = g(x_i + h, y_i + hk_3, z_i + ht_3) \end{array} \right.$$

## Esercizio

Si consideri la seguente equazione differenziale in  $z(t)$

$$z'' + 2(z')^2 + 4z = t$$

con condizioni iniziali

$$z(0) = 0, \quad z'(0) = 1$$

e lo si risolva numericamente usando due passi del metodo di Eulero con passo di discretizzazione  $h = 1/2$ .

Definendo il vettore

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} z \\ z' \end{pmatrix}$$

l'equazione precedente si riduce ad un sistema di due equazioni differenziali ordinarie del primo ordine

$$\begin{cases} y_1' = y_2 \\ y_2' = t - 2y_2^2 - 4y_1 \end{cases}$$

con condizioni iniziali

$$\begin{cases} y_1(0) = 0 \\ y_2(0) = 1 \end{cases}$$

e che in forma vettoriale si scrive

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, t) \quad \mathbf{y}(0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

con

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} z \\ z' \end{pmatrix}, \quad \text{e} \quad \mathbf{f} = \begin{pmatrix} z \\ t - 2y_2^2 - 4y_1 \end{pmatrix}.$$

Usando il metodo di Eulero calcoliamo

$$\mathbf{y}_{j+1} = \mathbf{y}_j + h\mathbf{f}(t_j, \mathbf{y}_j)$$



cioè

$$t_0 = 0 \quad \mathbf{y}_1 = \mathbf{y}_0 + h\mathbf{f}(t_0, \mathbf{y}_0) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 0 \\ 0 - 2 \cdot 1^2 - 4 \cdot 0 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 0 \end{pmatrix}$$

$$t_1 = 0.5 \quad \mathbf{y}_2 = \mathbf{y}_1 + h\mathbf{f}(t_1, \mathbf{y}_1) = \begin{pmatrix} 1/2 \\ 0 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 0 \\ -3/2 \end{pmatrix} = \begin{pmatrix} 1/2 \\ -3/4 \end{pmatrix}$$

```
function [T] = eulero_sistemi(x0,y0,z0,h,n_step,ffun,gfun)
% Soluzione di un sistema di due equazioni differenziali
% del primo ordine con il metodo di Eulero
%
% INPUT
% x0, y0,z0 = consizioni iniziali
% h = passo di discretizzazione
% n = numero passi
% ffun = input(funzione f(x,y,z): );
% gfun = input(funzione g(x,y,z): );
%
% OUTPUT
% T = matrice contenente [xi;yi;zi]
```

```
xi(1) = x0; yi(1) = y0; zi(1) = z0;
for i = 2:n_step+1
    x = xi(i-1);
    y = yi(i-1);
    z = zi(i-1);
    k1 = ffun(x,y,z);
    t1 = gfun(x,y,z);
    xi(i) = x + h;
    yi(i) = y + h*k1;
    zi(i) = z + h*t1;
end
figure(1)
plot(xi,yi,'r',xi,zi,'b')

T = [xi;yi;zi];
```

## Dal Command window

```
>> ffun = @(t,y,z)[z];
>> gfun = @(t,y,z)[t-2*z^2-4*y];
>> [T] = eulero_sistemi(0,0,1,.5,2,ffun,gfun);
>> disp(['      ti      ','      y1(i)      ','      y2(i)      ']), disp(T')
      ti      y1(i)      y2(i)
      0      0      1.0000
0.5000  0.5000      0
1.0000  0.5000  -0.7500
```

Usando il metodo di **RK del quarto ordine** con lo stesso passo di discretizzazione si ha

```
>> [T] = RK_sistemi(0,0,1,.5,2,ffun,gfun);
>> disp(['      ti      ','      y1(i)      ','      y2(i)      ']), disp(T')
      ti      y1(i)      y2(i)
      0      0      1.0000
0.5000  0.3148  0.3259
1.0000  0.3761  -0.0401
```

Usando il metodo di Eulero con passo di discretizzazione  $h = 0.05$ :

```
>> [T] = eulero_sistemi(0,0,1,.05,20,ffun,gfun);  
>> disp(['      ti      ', ' y1(i)  ', ' y2(i)  '   ]), disp(T')
```

ti	y1(i)	y2(i)
0	0	1.0000
0.0500	0.0500	0.9000
0.1000	0.0950	0.8115
0.1500	0.1356	0.7316
0.2000	0.1722	0.6585
0.2500	0.2051	0.5907
0.3000	0.2346	0.5273
0.3500	0.2610	0.4676
0.4000	0.2844	0.4110
0.4500	0.3049	0.3572
0.5000	0.3228	0.3060
0.5500	0.3381	0.2571
0.6000	0.3509	0.2104
0.6500	0.3614	0.1657
0.7000	0.3697	0.1232
0.7500	0.3759	0.0827
0.8000	0.3800	0.0444
0.8500	0.3823	0.0082
0.9000	0.3827	-0.0258
0.9500	0.3814	-0.0574
1.0000	0.3785	-0.0865

Mentre il metodo di RK con passo  $h = 0.05$  restituisce

```
>> [T] = RK_sistemi(0,0,1,.05,20,ffun,gfun);  
>> disp(['      ti      ','      y1(i)      ','      y2(i)      ']), disp(T')  
      ti      y1(i)      y2(i)  
      0      0      1.0000  
0.0500    0.0476    0.9057  
0.1000    0.0907    0.8211  
0.1500    0.1298    0.7440  
0.2000    0.1652    0.6730  
0.2500    0.1972    0.6070  
0.3000    0.2260    0.5452  
0.3500    0.2518    0.4870  
0.4000    0.2748    0.4319  
0.4500    0.2950    0.3797  
0.5000    0.3128    0.3300  
0.5500    0.3281    0.2827  
0.6000    0.3411    0.2378  
0.6500    0.3519    0.1950  
0.7000    0.3606    0.1544  
0.7500    0.3674    0.1160  
0.8000    0.3723    0.0798  
0.8500    0.3754    0.0458  
0.9000    0.3769    0.0142  
0.9500    0.3768   -0.0149  
1.0000    0.3754   -0.0415
```

# Esercizio

Consideriamo l'**oscillatore armonico** descritto dall'equazione differenziale del **secondo ordine** ( $n = 2$ ):

$$u''(x) + 2\alpha u'(x) + \beta^2 u(x) = 0$$



$$u''(x) = -2\alpha u'(x) - \beta^2 u(x) := g(x, u(x), u'(x))$$

$$\left\{ \begin{array}{l} u''(x) = g(x, \underbrace{u(x)}_{y(x)}, \underbrace{u'(x)}_{z(x)}) \\ u(x_0) = u_0 \\ u'(x_0) = u_1 \end{array} \right. \rightarrow \left\{ \begin{array}{l} y(x) = u(x) \\ y'(x) = z(x) \\ z'(x) = g(x, y(x), z(x)) \\ y(x_0) = y_0 \\ z(x_0) = y_1 \end{array} \right.$$

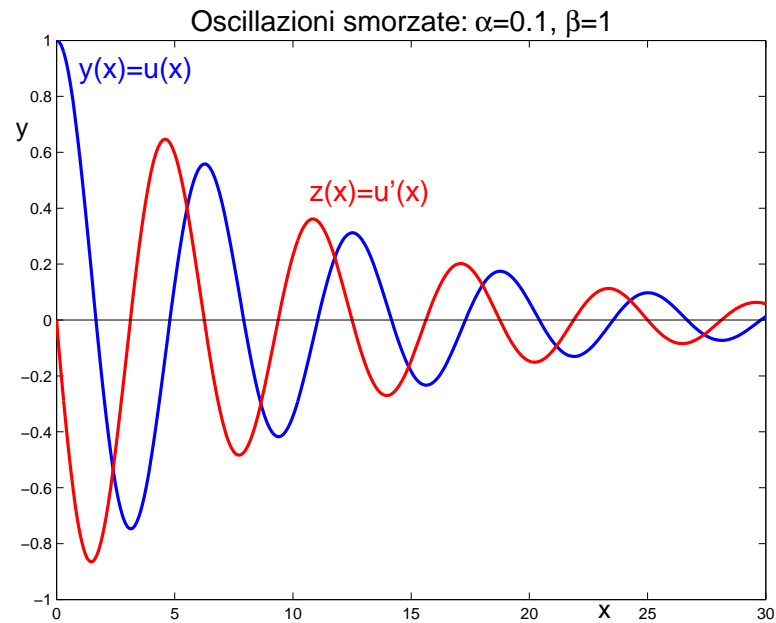
Per risolvere il sistema differenziale ottenuto, si possono applicare i metodi già visti per la soluzione dei sistemi (ad esempio, il metodo di Runge-Kutta classico).

# Script MATLAB

```
% Soluzione di un sistema di due equazioni differenziali
% del primo ordine con il metodo di Runge-Kutta classico
%
% Input
%
clear
x0 = input('x0: ');
y0 = input('y0: ');
z0 = input('z0: ');
h = input('passo: ');
n = input('numero passi: ');
ffun = input('funzione f(x,y,z): ');
f = inline(ffun,'x','y','z');
gfun = input('funzione g(x,y,z): ');
g = inline(gfun,'x','y','z');
%
% Algoritmo
%
xi(1) = x0; yi(1) = y0; zi(1) = z0;
for i = 2:n
    x = xi(i-1);
    y = yi(i-1);
    z = zi(i-1);
    k1 = f(x,y,z);
    t1 = g(x,y,z);
```

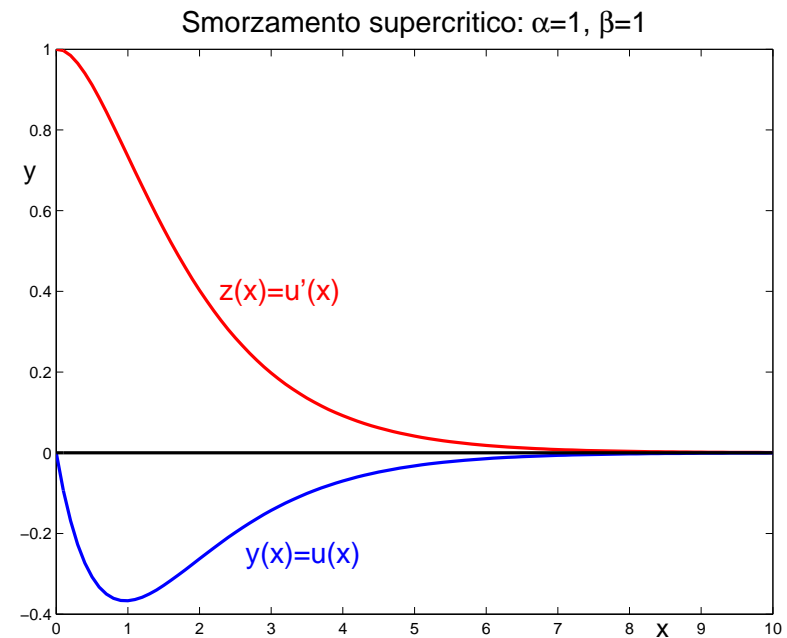


```
k2 = f(x+0.5*h,y+0.5*h*k1,z+0.5*h*t1);
t2 = g(x+0.5*h,y+0.5*h*k1,z+0.5*h*t1);
k3 = f(x+0.5*h,y+0.5*h*k2,z+0.5*h*t2);
t3 = g(x+0.5*h,y+0.5*h*k2,z+0.5*h*t2);
k4 = f(x+h,y+h*k3,z+h*t3);
t4 = g(x+h,y+h*k3,z+h*t3);
xi(i) = x + h;
yi(i) = y + h*(k1+2*k2+2*k3+k4)/6;
zi(i) = z + h*(t1+2*t2+2*t3+t4)/6;
end
figure(1)
plot(xi,yi,'r',xi,zi,'b')
```



### Dati del problema

$x_0$ : 0  
 $y_0$ : 1  
 $z_0$ : 0  
 passo: 0.2  
 numero passi: 150  
 funzione  $f(x,y,z)$ : 'z'  
 funzione  $g(x,y,z)$ : '-2\*0.1\*z-y'



### Dati del problema

$x_0$ : 0  
 $y_0$ : 1  
 $z_0$ : 0  
 passo: 0.2  
 numero passi: 150  
 funzione  $f(x,y,z)$ : 'z'  
 funzione  $g(x,y,z)$ : '-2\*z-y'

## Esercizio

1. Utilizzare il programma per approssimare la soluzione del problema del **pendolo non lineare**. Confrontare graficamente la soluzione approssimata ottenuta con quella del problema lineare.
2. Risolvere il sistema di equazioni differenziali del **modello predatore-predatore**, per i seguenti valori dei parametri  $k_1 = k_2 = 1$ ,  $\mu_1 = 300$ ,  $\mu_2 = 200$ ,  $y_{10} = 400$ ,  $y_{20} = 100$ , usando il metodo di Eulero con passi 0.1, 0.01 e 0.001, il metodo di Runge Kutta del quarto ordine con passi 0.1 e 0.01 e il metodo di Crank-Nicolson con passi 0.01 e 0.001. Confrontare i risultati

### Riferimenti bibliografici

L. Gori, *Calcolo Numerico*: Cap. 9 §§ 9.1-9.6, 9.14 (esclusi metodi di Adams-Bashfort, Milne e Numerov)

L. Gori, M.L. Lo Cascio, F. Pitolli, *Esercizi di Calcolo Numerico*: 6.1, 6.2, 6.3, 6.4, 6.5, 6.7, 7.76, 7.80, 7.85

# Esercizi

# Esercizio 1

Dato il problema di Cauchy

$$\begin{cases} y' &= 1 + y^2, & x \in [0, 1] \\ y(0) &= 0 \end{cases}$$

dopo aver verificato che  $y = \tan(x)$  è la soluzione esatta,

1. risolvere numericamente usando il metodo di Eulero esplicito con passo  $h = \frac{1}{2}$  e produrre una stima per  $y(1)$
2. scegliendo un passo  $h = \frac{1}{2^p}$  e trascurando l'influenza degli errori di arrotondamento, stabilire per quale valore di  $p$  l'errore globale di troncamento per  $y(1)$  diventa minore di  $0.5 \cdot 10^{-4}$ .

## Soluzione

Si osserva che se  $y^* = \tan(x)$ , allora  $(y^*)' = \frac{1}{\cos(x)^2}$ .

Usando la relazione fondamentale della trigonometria si ha

$$(y^*)' = \frac{\cos^2(x) + \sin^2(x)}{\cos^2(x)} = 1 + \tan^2(x) = 1 + (y^*)^2.$$

Quindi,  $y^*$  è soluzione del problema di Cauchy.

La soluzione è unica in  $[0, 1] \times \mathbf{R}$ : Infatti, poichè risulta  $|f_y| = |2y| = |2 \tan x| \leq 2 \tan(1)$ , la funzione  $f(x, y) = 1 + y^2$  è Lipschitziana rispetto a  $y$ , uniformemente rispetto a  $x$  nel dominio considerato.

Il metodo di Eulero produce approssimazioni della soluzione nei punti  $x_i = x_0 + ih$ .

Se  $h = 1/2$  e  $x_0 = 0$ , la stima per  $y(1)$  si ottiene dopo **due iterazioni**:

$$x_1 = 1/2 \qquad y(x_1) = y(0) + h(1 + y^2(0)) = 1/2$$

$$x_2 = 1 \qquad y(x_2) = y(x_1) + \frac{1}{2}(1 + y^2(x_1)) = \frac{1}{2} + \frac{1}{2} \left(1 + \frac{1}{4}\right) = \frac{9}{8} = 1.1250$$

Poichè  $y^*(1) = 1.5574$ , l'**errore relativo** è  $\frac{|y^*(1) - y(1)|}{y^*(1)} = \frac{1.5574 - 1.1250}{1.5574} = 0.2776$ .



Una stima per l'errore globale di troncamento  $e_i$  per il metodo di Eulero è data

$$|e_i| < \frac{M}{2} h \frac{e^{L(x_i - x_0)} - 1}{L},$$

con  $M = \max_{x \in [0,1]} |y''(x)|$ ,  $L$  la costante di Lipschitz della funzione  $f(x, y) = 1 + y^2$  rispetto a  $y$ ,  $x_i$  il punto in cui valutare l'errore e  $h$  il passo adottato.

In questo caso  $h = \frac{1}{2^p}$ ,  $x_i - x_0 = 1 - 0 = 1$ ,

$$\begin{aligned} M &= \max_{x \in [0,1]} |2yy'| = \max_{x \in [0,1]} |2y(x)(1 + y(x)^2)| = \\ &= |2 \tan(1)(1 + \tan^2(1))| = 10.6699 \end{aligned}$$

(è stata usata la monotonia di  $\tan(x)$  nell'intervallo  $[0, 1]$ )

mentre come costante di Lipschitz si assume

$$L = \max_{(x,y)} |f_y(x, y)| = \max_{(x,y)} |2y| = |2y(1)| = 3.1148.$$

Quindi,

$$p > \log_2 \left( \frac{10.6699 e^{3.1148} - 1}{2} \frac{1}{3.1148 \cdot 0.5 \cdot 10^{-4}} \right) = 19.49,$$

ovvero  $p = 20$ , e quindi  $h = \frac{1}{2^{20}} = 9.536743164062500 \cdot 10^{-7}$ .  
Se non si trascurano gli errori di arrotondamento, l'errore di arrotondamento si maggiora nel modo seguente

$$|e_i| < \left( \frac{M}{2} h + \frac{\eta}{h} \right) \frac{e^{L(x_i - x_0)} - 1}{L},$$

dove  $\eta$  è tale che  $|\eta_i| \leq |\eta|$ ,  $\forall i$ .

Allora è possibile definire il passo ottimo  $h_{ott}$  che rende minimo l'errore totale (troncamento + propagazione), cioè

$$h_{ott} = \sqrt{\frac{2\eta}{M}}$$

Supponendo che  $\eta = 0.5 \cdot 10^{-15}$ , si ha

$$h_{ott} = \sqrt{\frac{10^{-15}}{10.6699}} = 9.680991201464672 \cdot 10^{-9}$$

## Esercizio 2

Si vuole risolvere il problema di Cauchy

$$\begin{cases} y'(x) = y + \cos(x), & x \in (-2, 0), \\ y(-2) = 0, \end{cases}$$

con il metodo di Eulero.

- 2.1)** Sapendo che  $y(x) \in C^{(3)}([-2, 0])$  e che  $|y^{(p)}(x)| \leq 4$ ,  $p = 0, 1, 2, 3$ , per  $x \in [-2, 0]$ , determinare quale passo di integrazione garantisce che l'errore globale di troncamento in  $x = 0$  sia inferiore a  $0.5 \cdot 10^{-5}$ .
- 2.2)** Sapendo che ad ogni passo si ha un'errore di arrotondamento sulla 5 cifra decimale, individuare il passo ottimo.

## Soluzione

**2.1)** Bisogna individuare un passo  $h$  tale che

$$|e_i| \leq \frac{h}{2} M \frac{e^{(0-x_0)L} - 1}{L} \leq 0.5 \cdot 10^{-5},$$

dove  $M \geq |y''(x)|$  (e quindi  $M = 4$ ) e  $L$  è la costante di Lipschitz di  $f$  rispetto a  $y$ . Poichè  $f(x, y) = y + \cos x$  e

$$\left| \frac{\partial f(x, y)}{\partial y} \right| = |1| \Rightarrow L = 1,$$

si ha

$$\frac{h}{2} 4 \frac{e^2 - 1}{1} \leq 0.5 \cdot 10^{-5},$$

da cui si ottiene

$$h \leq \frac{0.5 \cdot 10^{-5}}{2(e^2 - 1)} \approx 3.9129 \cdot 10^{-7}.$$

Scegliendo  $h = 1 \cdot 10^{-7}$ , valore di poco sopra alla maggiorazione ottenuta, sono necessari  $2 \cdot 10^7$  passi del metodo di Eulero per arrivare all'estremo destro dell'intervallo di integrazione.

2.2) Il passo ottimo è dato da

$$h_{ott} = \sqrt{\frac{2\eta}{M}} = \sqrt{\frac{2 \cdot 0.5 \cdot 10^{-5}}{4}} \approx 0.0016.$$

Il passo individuato al punto precedente, essendo molto più piccolo di  $h_{ott}$ , può produrre un errore di propagazione elevato.

## ESERCIZIO 3

Si consideri il problema di Cauchy

$$\begin{cases} y' = y - 2\sin(x), & 0 \leq x \leq 5 \\ y(0) = 1 \end{cases} .$$

- 3.1)** Verificare che la soluzione esiste ed è unica in  $D = [0, 5] \times \mathbf{R}$  e che la funzione  $y(x) = \cos(x) + \sin(x)$  è la soluzione esatta.
- 3.2)** Supponendo di voler risolvere numericamente il problema di Cauchy con il metodo di Eulero,
- posto il passo di discretizzazione  $h = \frac{5}{2^m}$  e trascurando gli errori di arrotondamento, stabilire per quale valore di  $m \in \mathbf{N}$  l'errore globale di troncamento per  $y(5)$  diventa minore di  $0.5 \cdot 10^{-3}$ ;
  - supponendo di introdurre ad ogni passo un errore di arrotondamento inferiore a  $0.5 \cdot 10^{-3}$ , calcolare il passo di discretizzazione ottimo e confrontarlo con quello ottenuto al passo precedente.

## ESERCIZIO 4

Si consideri il seguente problema di Cauchy (PC):

$$\begin{cases} y'(t) = y(t) - t^2 + 1 & t \in [0, 2] \\ y(0) = 1/2 \end{cases}$$

- 4.1** Scelto un passo  $h = \frac{1}{3^m}$ , con  $m \in \mathbf{N}$  e sapendo che  $\max_{0 \leq t \leq 2} |y''(t)| \leq 1.7$ , stabilire per quali valori di  $m$  l'errore globale massimo della soluzione numerica del (PC) calcolata con il metodo di Eulero, trascurando gli errori di arrotondamento, risulta non superiore a  $0.5 \cdot 10^{-3}$ .
- 4.2** Confrontare il passo  $\bar{h}$  più grande calcolato al punto precedente con il valore del passo ottimo  $h_{ott}$  calcolato quando si considera un errore massimo di arrotondamento pari a  $0.5 \cdot 10^{-9}$ .

## ESERCIZIO 5

**5.1** Illustrare il problema della soluzione numerica di una equazione differenziale con particolare riferimento al metodo di Eulero esplicito.

**5.2** Stabilire se il metodo di Eulero è adatto ad approssimare la soluzione  $z(y) = \left(\frac{2}{3}y\right)^{\frac{3}{2}}$ , per  $y \geq 0$ , del seguente problema di Cauchy

$$\begin{cases} z' = z^{\frac{1}{3}} \\ z(0) = 0. \end{cases}$$