Calcolo Numerico con elementi di programmazione

(A.A. 2014-2015)

Appunti delle lezioni su
equazioni differenziali ordinarie

Modello matematico: eq. differenziali ordinarie

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**):

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

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

$-k\,x$: **legge di Hooke**

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

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

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

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

Eq. differenziali ordinarie: problema di Cauchy

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

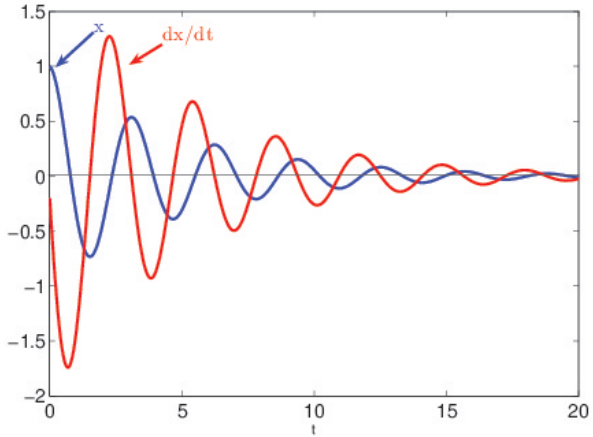
Problema di Cauchy:

$$\begin{cases} m\dfrac{d^2x}{dt^2} + b\dfrac{dx}{dt} + kx = 0 \\ x(0) = x_0 \qquad c.i. \\ \dfrac{dx}{dt}(0) = v_0 \qquad c.i. \end{cases} \Rightarrow \begin{cases} x(t) = x_m\, e^{-bt/2m}\, \cos(\omega_m\, x + \varphi_m) \\ x_m = \dfrac{x_0}{\cos\varphi_m} \\ \tan\varphi_m = \dfrac{v_0}{x_0}\dfrac{2m}{b} \end{cases}$$

Esempio:

$$\begin{cases} x(0) = 1 \\ \dfrac{dx}{dt}(0) = 0 \end{cases}$$

$$m = 250g$$
$$k = 85\ N/m$$
$$b = 70\ g/s$$



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 θ è l'angolo tra il pendolo e la verticale. Il problema è completato con le condizioni iniziali

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

Calcolare $\theta(x)$ nel caso in cui $L = 0.6m$, $\theta_0 = \pi/6$ rad, $v_0 = 0m/s$ (si assuma $g = -9.81m/s^2$)

Equazioni differenziali di ordine n

$$\begin{cases} y^{(n)}(x) = g(x, y(x), y'(x), y''(x), \ldots, y^{(n-1)}(x)) \\ \\ y^{(k)}(x_0) = y_k \qquad k = 0, 1, \ldots, n-1 \end{cases} \qquad \textbf{Condizioni iniziali}$$

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

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

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

$$\begin{cases} y_1'(x) = y_2(x) \\ y_2'(x) = y_3(x) \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ y_{n-1}'(x) = y_n(x) \\ y_n'(x) = g(x, y_1(x), y_2(x), y_3(x), \ldots, y_n(x)) \end{cases} \qquad \begin{cases} y_{10} = y_0 \\ y_{20} = y_1 \\ \ldots\ldots\ldots\ldots\ldots \\ y_{n0} = y_{n-1} \end{cases} \qquad \textbf{Condizioni iniziali}$$

Sistemi di equazioni differenziali: esempio

Il più semplice modello che descrive la competizione tra due specie è il **modello preda-predatore di Lotka-Volterra**. Si tratta di una coppia di equazioni differenziali in cui $y_1(t)$ rappresenta il **numero di prede** e $y_2(t)$ rappresenta il **numero di predatori**. In assenza di predatori le prede crescono in modo esponenziale; i predatori invece, in assenza di prede, muoiono rapidamente. L'**interazione** tra le due specie è descritta da un termine proporzionale a entrambe le popolazioni:

$$\begin{cases} \dot{y}_1 = k_1 \left(1 - \dfrac{y_2}{\mu_2}\right) y_1 \qquad t > 0 \\ \\ \dot{y}_2 = -k_2 \left(1 - \dfrac{y_1}{\mu_1}\right) y_2 \\ \\ y_1(0) = y_{10} \qquad y_2(0) = y_{20} \end{cases}$$

dove k_1, k_2, μ_1, μ_2 sono **costanti positive** e y_{10}, y_{20} sono le **condizioni iniziali**.

Nota. Se $y_{10} = \mu_1$, $y_{20} = \mu_2 \Rightarrow \dot{y}_1 = \dot{y}_2 = 0$, cioé (μ_1, μ_2) è un **punto di equilibrio**. Per alcuni valori di μ_1, μ_2 la soluzione è **periodica**.

Eq. differenziali ordinarie del primo ordine

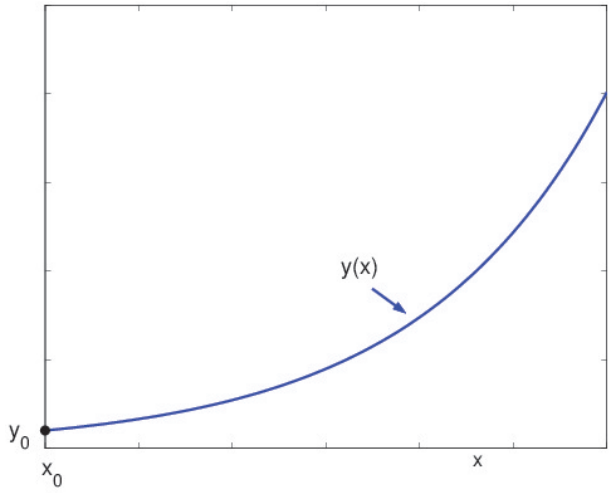
Problema di Cauchy: $\begin{cases} y'(x) = f(x, y(x)) \\ y(x_0) = y_0 \end{cases}$ **condizione iniziale**

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

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

Soluzione esatta:

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



Eq. differenziali ordinarie del primo ordine

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.

$\longrightarrow$ 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

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

chiamata **equazione logistica**.

Assumendo:

i) numero di individui al tempo al tempo iniziale $P(0) = 50976$,
ii) tasso di nascita $b = 2.9 \times 10^{-2}$,
iii) tasso di morte $d(x) = kP(x)$ proporzionale al numero di individui presenti al tempo x con $k = 1.4 \times 10^{-7}$

calcolare il numero di individui dopo 5 anni.

Crescita di una popolazione

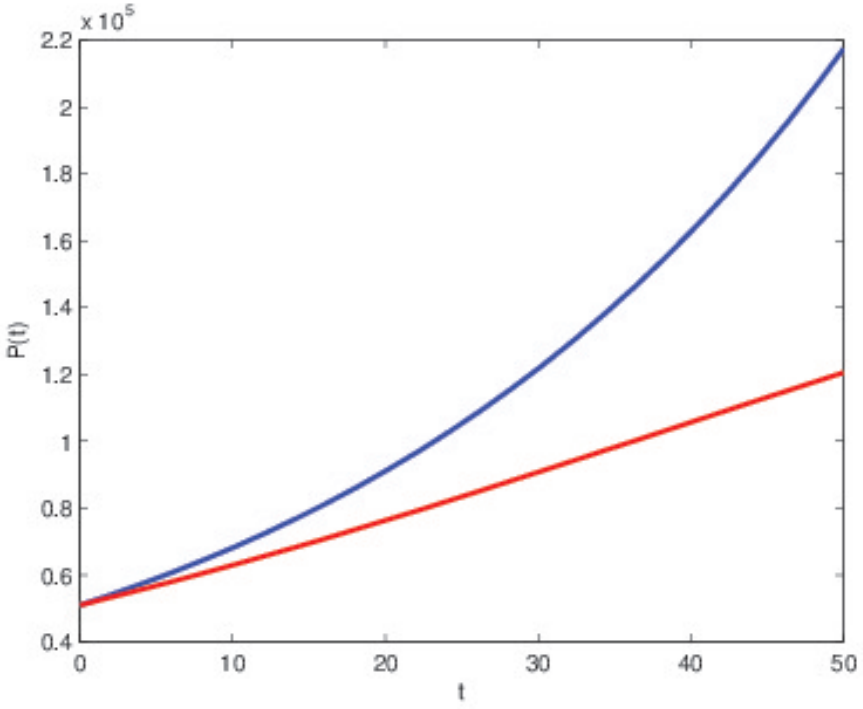
$$\begin{cases} \dfrac{dP(t)}{dt} = bP(t) \\ P(0) = 50976 \end{cases}$$

$$P(t) = P(0)\,\mathrm{e}^{bt}$$

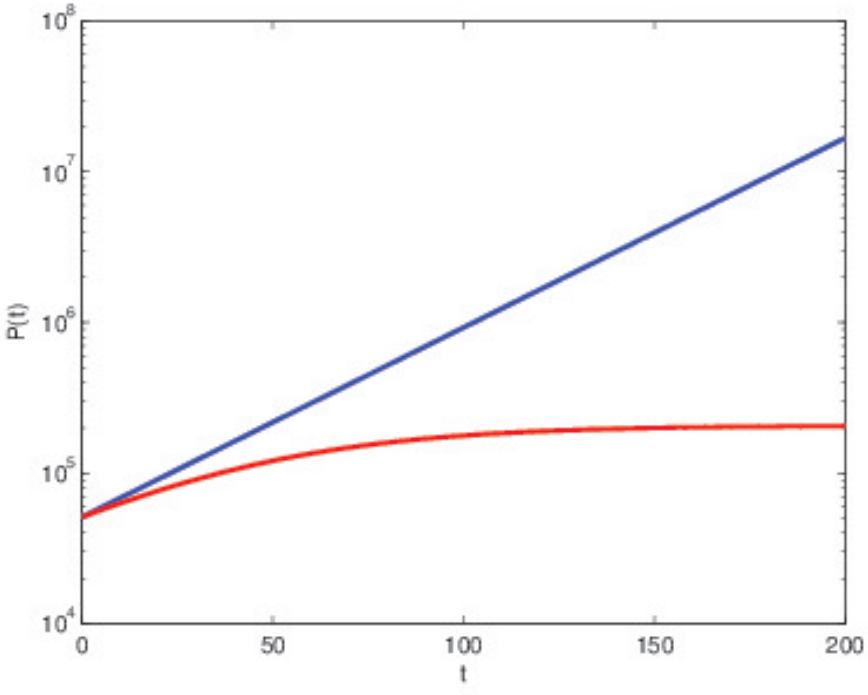
Equazione logistica

$$\begin{cases} \dfrac{dP(t)}{dt} = bP(t) - d(t)P(t) \\ P(0) = 50976 \end{cases}$$

$$P(t) = P(0)\,\mathrm{e}^{bt}\,\frac{1}{1 - \frac{k}{b}\,P(0)(1 - \mathrm{e}^{bt})}$$



(scala lineare)



(scala semilogaritmica)

Crescita esponenziale:

$$\lim_{t\to\infty} P(t) = \infty$$

Equazione logistica:

$$\lim_{t\to\infty} P(t) = \frac{b}{k}$$

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 $D \subset \mathbf{R}^2$, se esiste una **costante** $L > 0$ tale che

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

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

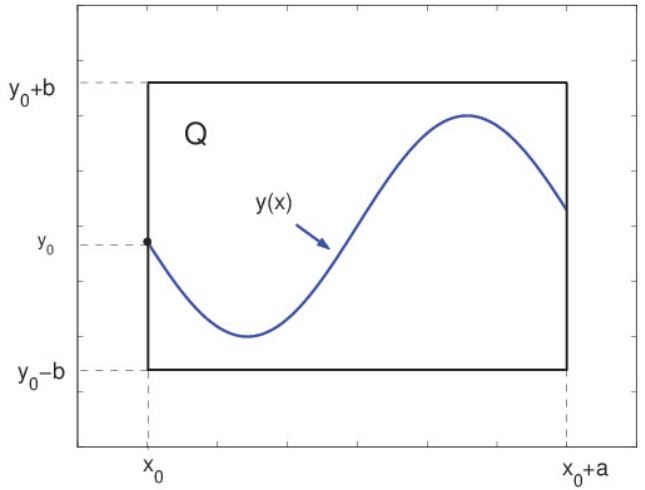
Esempio: $f(x,y) = K\,y \Rightarrow f_y(x,y) = K \Rightarrow$ è **limitata**

$\Rightarrow$ f è **lipschitziana**

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

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

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

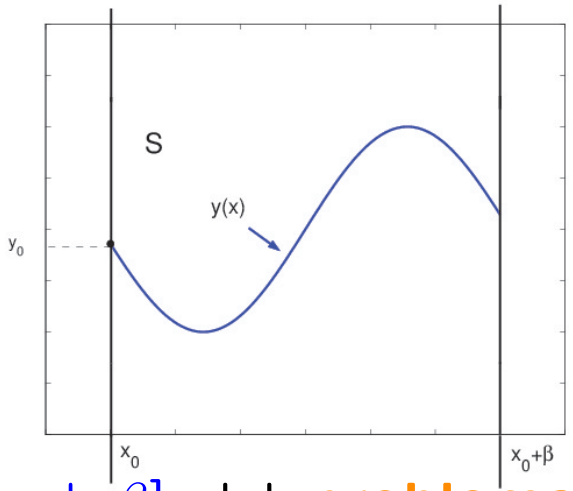


$\Rightarrow$ esiste un'**unica** soluzione $y(x) \in C^1[x_0, x_0 + \beta]$ del **problema di Cauchy** in $I = [x_0, x_0 + \beta]$ con $\beta = \min\left[a, \frac{b}{M}\right]$, $M = \max_Q |f(x, y)|$.

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

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

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**

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

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

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

dove $\varepsilon > 0$ è una prefissata **tolleranza**, purché $\delta = \delta(\varepsilon)$ 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**

$$\begin{cases} y'(x) = -y + x + 1 \quad x \in [0, 10] \\ y(0) = 1 \end{cases}$$

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

- Il seguente **Problema di Cauchy**

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

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

Metodo di Eulero

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

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

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

Sviluppo in **serie di Taylor**:

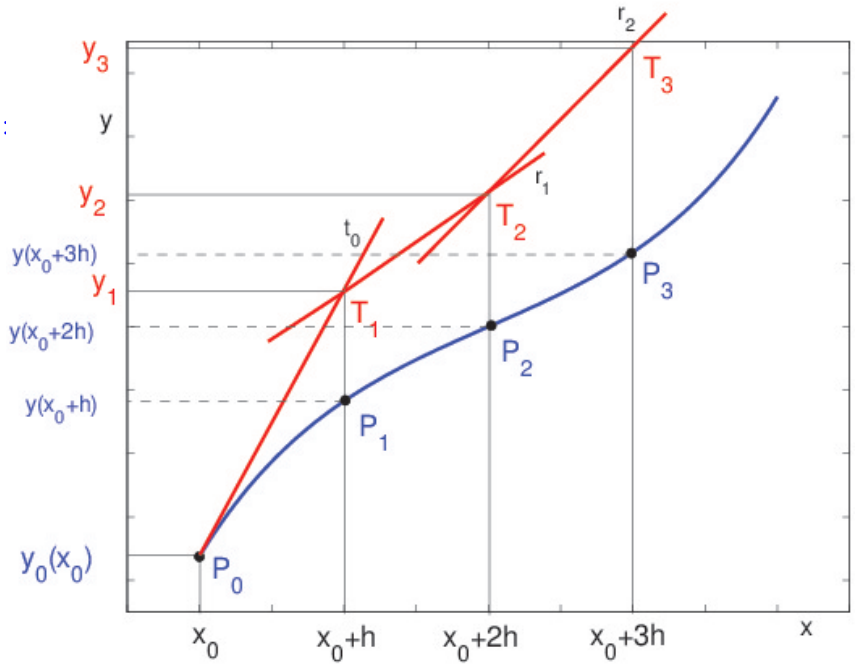
$$\begin{aligned} y(x_1) &= y(x_0 + h) = y(x_0) + y'(x_0)h + \frac{1}{2}y''(\tau_1)h^2 \\ &= y(x_0) + f(x_0, y(x_0))h + \frac{1}{2}y''(\tau_1)h^2 \\ &\qquad \tau_1 \in [x_0, x_1] \end{aligned}$$

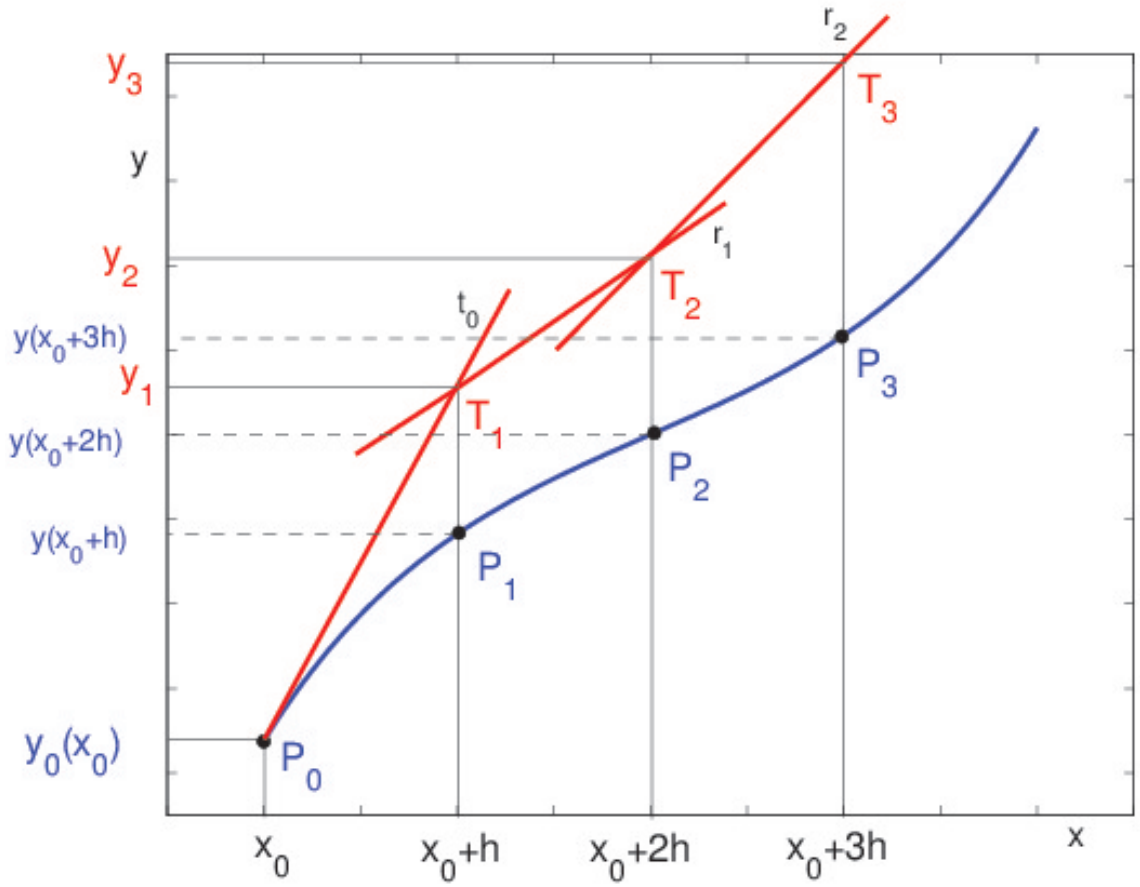
Soluzione **approssimata**:

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

Errore globale di troncamento:

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





Da un punto di vista geometrico, la determinazione di y_1 corrisponde ad approssimare l'ordinata del punto $P_1(x_1, y(x_1))$, situato sulla **curva soluzione**, con l'ordinata del punto $T_1(x_1, y_1)$ situato sulla retta t_0, tangente alla curva soluzione nel punto $P_0(x_0, y_0)$ ed avente coefficiente angolare $y'(x_0) = f(x_0, y(x_0))$.

Nota. Lo schema nasce dall'approssimare la derivata con le differenze finite

$$\frac{y_{i+1} - y_i}{h} = f(x_i, y_i)$$

Ripetendo il procedimento a partire da x_1 avremo

$$y(x_2) = y(x_1) + f(x_1, y(x_1))h + \frac{1}{2}y''(\tau_2)h^2 \qquad \tau_2 \in [x_1, x_2]$$

e la soluzione approssimata è data da

$$y_2 = y_1 + hf(x_1, y_1)$$

Da un punto di vista geometrico, l'ordinata del punto $P_2(x_2, y(x_2))$, situato sulla **curva soluzione**, è approssimata con l'ordinata di un punto $T_2(x_2, y_2)$ situato sulla retta r_1 per T_1 e di coefficiente angolare $f(x_1, y_1)$; r_1 non è tangente alla curva soluzione in P_1 e non è in generale parallela essendo $f(x_1, y_1)$ diverso da $f(x_1, y(x_1))$.
La soluzione approssimata y_2 risente sia della linearizzazione che del fatto che è calcolata a partire da y_1 che a sua volta è un valore approssimato del valore esatto $y(x_1)$.

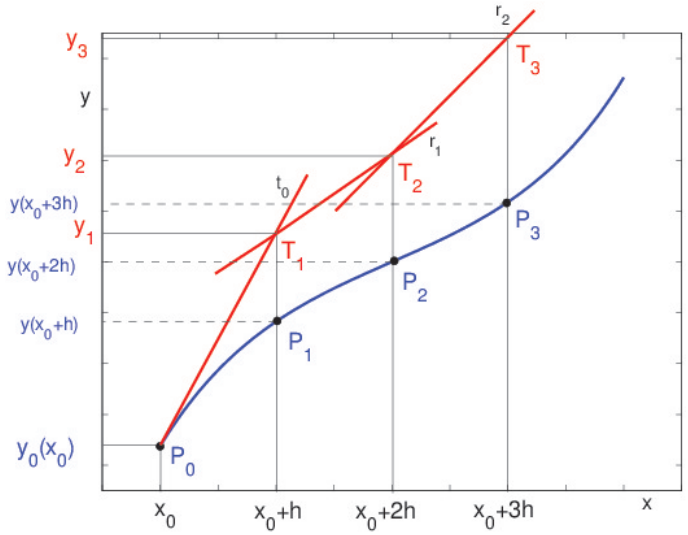
Algoritmo del metodo di Eulero

Algoritmo:

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

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**:

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

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

- 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

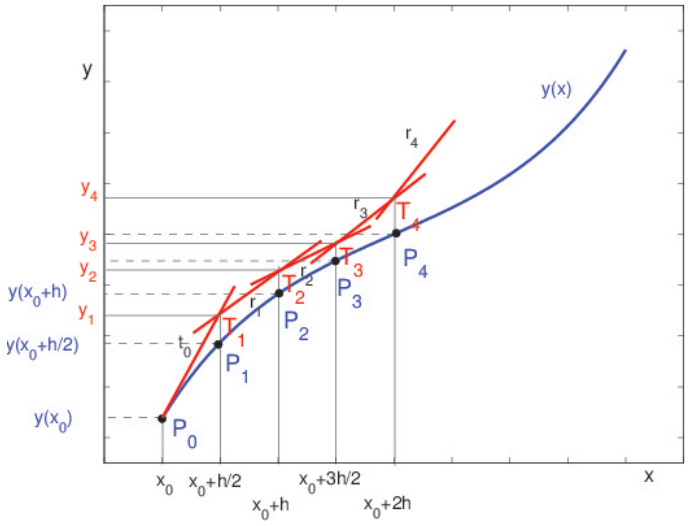
$$\lim_{h\to 0} \max_{1\le i\le n} |e_i| = 0 \quad \Leftrightarrow \quad \lim_{i\to\infty} y_i = y(\bar{x})$$

con $\bar{x} = x_0 + ih$ fissato

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

$$\lim_{h\to 0} \max_{1\le i\le n} |e_i| = 0$$

$\Rightarrow$ Il metodo di Eulero è **convergente**



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

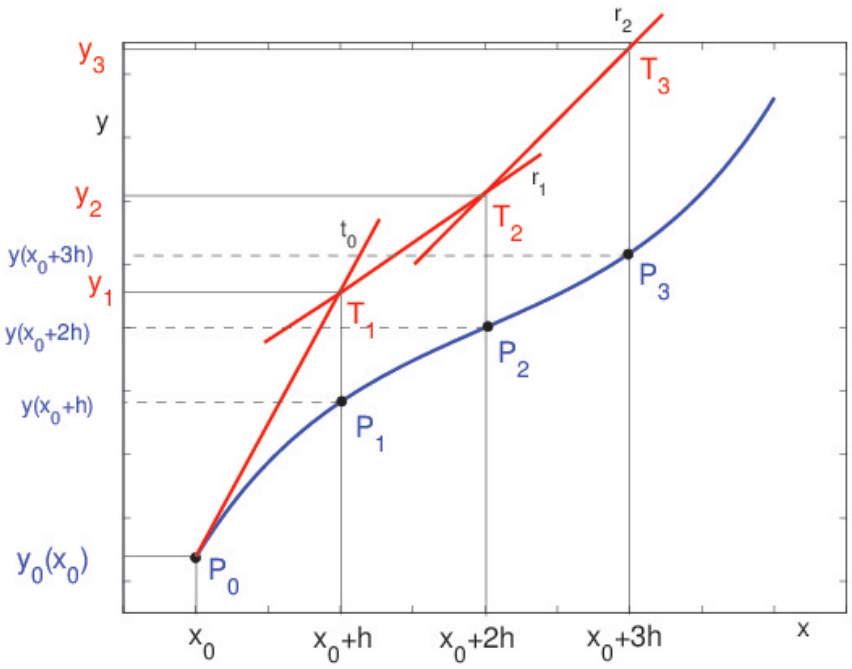
- **Consistente:**

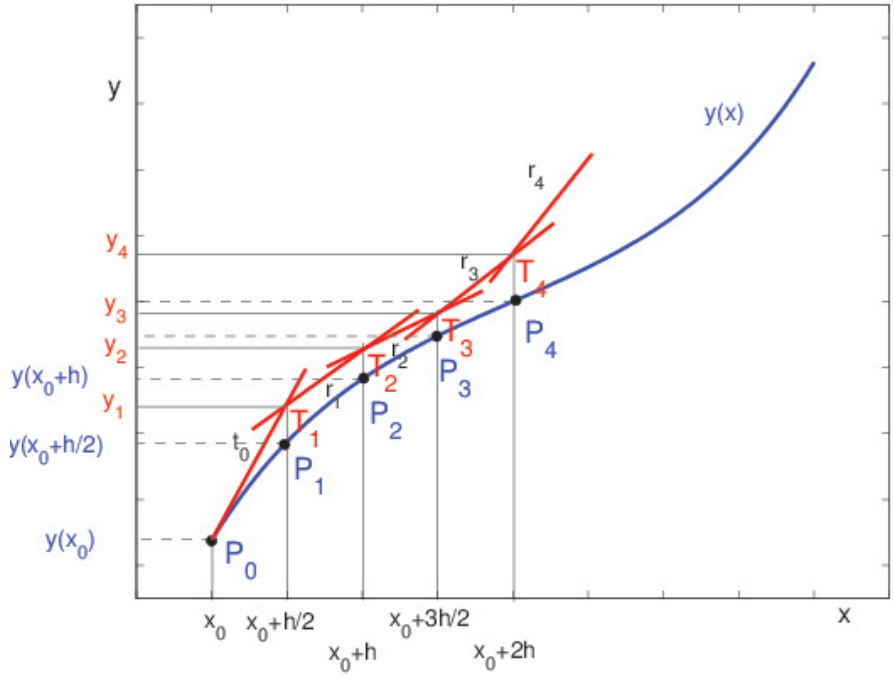
$$\lim_{h\to 0} \frac{R(x,y;h;f)}{h} = 0$$

+

- **Stabile:**

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


y3
y
y2
y(x0+3h)
y1
y(x0+2h)
y(x0+h)
y0(x0)
r2
T3
r1
T2
t0
T1
P3
P2
P1
P0
x0
x0+h
x0+2h
x0+3h
x


y
y(x)
r4
y4
r3
T4
y3
T3
y2
r2
T2
y(x0+h)
r1
T1
y1
P4
P3
P2
y(x0+h/2)
t0
P1
y(x0)
P0
x0
x0+h/2
x0+h
x0+3h/2
x0+2h
x

Esercizio 1

Dato il Problema di Cauchy

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

verificare l'estistenza e l'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 $f(x,y)$ è **lipschitziana** in $S = [0,2] \times \mathbf{R}$, cioè se esiste una costante $L > 0$ (costante di Lipschitz) tale che

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

$$f(x,y) = \begin{cases} y - x, & x < y \\ x - y, & x \geq y \end{cases} \Rightarrow f_y(x,y) = \begin{cases} 1, & x < y \\ -1, & x > y \end{cases} \quad \forall (x,y) \in S \Rightarrow$$

$|f_y| = 1,\ x \neq y \Rightarrow f_y$ risulta limitata in $x \neq y \Rightarrow f(x,y)$ è lipschitziana in S con costante di Lipschitz pari a $1 \Rightarrow$ il metodo di Eulero è convergente per $x \in [0,2]$.

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

$$x_i = x0 + i_h$$

in cui si approssimano i valori della soluzione del PC

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

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

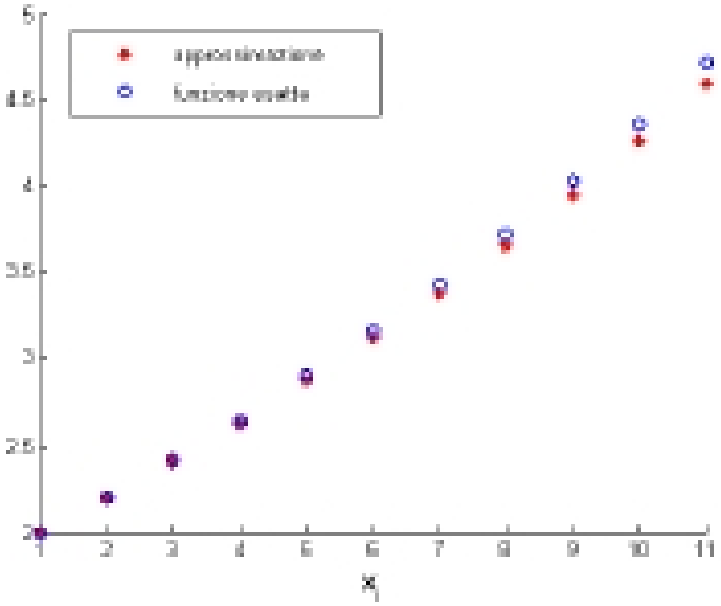
$$n = \frac{b-a}{h} = 10$$

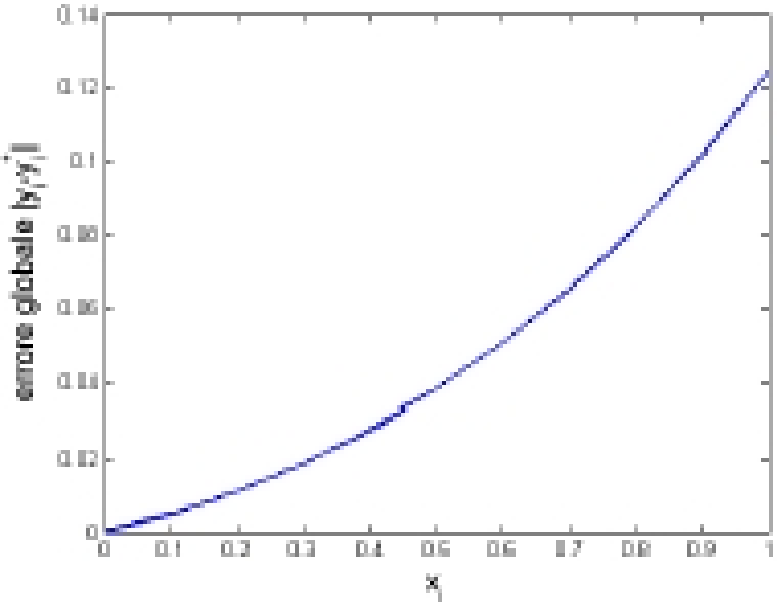
passi.

$$
\begin{aligned}
y_1 &= y_0 + hf(x_0, y_0) &&= 2 + 0.1|2 - 0| = 2.2, && x_1 = x_0 + h = 0.1 \\
y_2 &= y_1 + hf(x_1, y_1) &&= 2.2 + 0.1|2.2 - 0.1| = 2.41, && x_2 = x_0 + 2h = 0.2 \\
y_3 &= y_2 + hf(x_2, y_2) &&= 2.41 + 0.1|2.41 - 0.2| = 2.631, && x_3 = x_0 + 3h = 0.3 \\
&\vdots \\
&\vdots \\
y_{10} &= y_9 + hf(x_9, y_9) &&= \ldots = 4.5937424601, && x_{10} = x_10 + 10h = 1
\end{aligned}
$$

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 $e_i = |y_i - y_i^*|$

x_i	y_i	y_i^*	e_i
0.0	2.0000000000	2.00000000000000	0.00000000000000
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



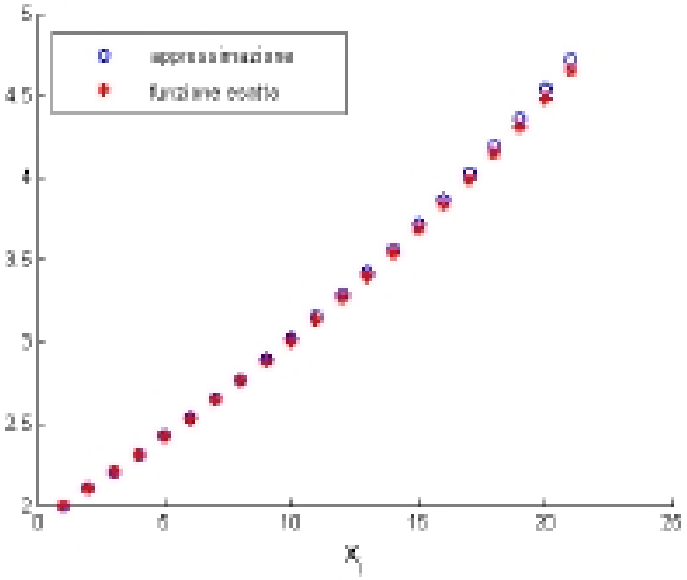


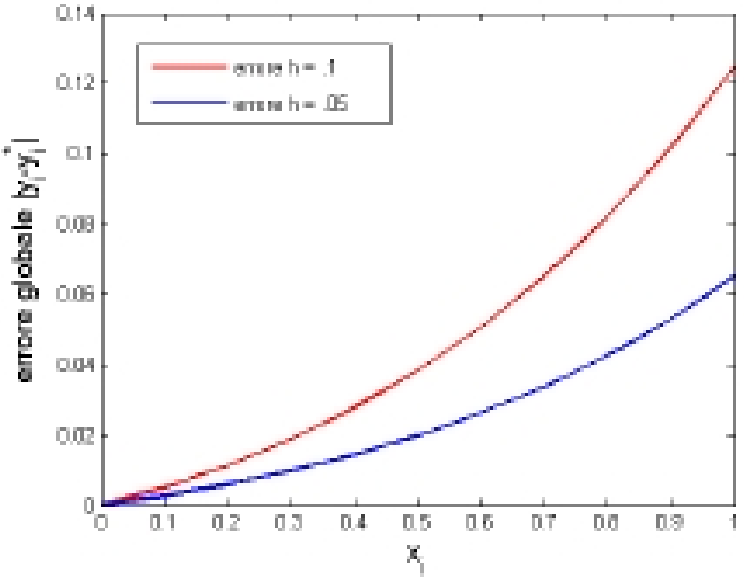
L'errore di stima cresce man mano che x_i si avvicina a 1.

Considerando il passo di discretizzazione $h = 0.05$ si eseguono 20 passi del metodo di Eulero

x_i	y_i	y_i^*	e_i
0.00	2.00000000000000	2.00000000000000	0
0.05	2.10000000000000	2.10127109637602	0.00127109637602
0.10	2.20250000000000	2.20517091807565	0.00267091807565
0.15	2.30762500000000	2.31183424272828	0.00420924272828
0.20	2.41550625000000	2.42140275816017	0.00589650816017
0.25	2.52628156250000	2.53402541668774	0.00774385418774
0.30	2.64009564062500	2.64985880757600	0.00976316695100
0.35	2.75710042265625	2.76906754859326	0.01196712593701
0.40	2.87745544378906	2.89182469764127	0.01436925385221
0.45	3.00132821597852	3.01831218549017	0.01698396951165
0.50	3.12889462677744	3.14872127070013	0.01982664392269
0.55	3.26033935811631	3.28325301786740	0.02291365975108
0.60	3.39585632602213	3.42211880039051	0.02626247436838
0.65	3.53564914232324	3.56554082901390	0.02989168669066
0.70	3.67993159943940	3.71375270747048	0.03382110803108
0.75	3.82892817941137	3.86700001661267	0.03807183720131
0.80	3.98287458838194	4.02554092849247	0.04266634011053
0.85	4.14201831780103	4.18964685192599	0.04762853412496
0.90	4.30661923369108	4.35960311115695	0.05298387746587
0.95	4.47695019537564	4.53570965931585	0.05875946394021
1.00	4.65329770514442	4.71828182845905	0.06498412331463

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





L'errore locale di troncamento è

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

Esercizio 2

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
- 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 I ordine
% con il metodo di Eulero
%
% INPUT
%   x0, y0: condizione iniziale
%   n_step: numero dei passi da eseguire
%   h:      passo di discretizzazione
%   fun:    espressione di f(x,y)
%
% OUTPUT
% T:  matrice di dimensione 2x(n_step+1)
%     la prima   riga contiene il vettore dei nodi,
%     la seconda riga contiene il vettore delle approssimazioni
%     della soluzione nei nodi
%
```

```
xi = zeros(1,n_step+1);
yi = zeros(1,n_step+1);

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, hold on
plot(0:.001:intervallo, y_true(0:.001:intervallo),'m')
for i = 1:length(h)
  n_step = 1/h(i);
  T = eulero(x0,y0,fun,h(i),n_step);
  SOL{i} = T;
  err{i} = abs(T(2,:)-y_true(T(1,:)));
  EH = [EH max(err{i})];
```

```
  plot(T(1,:),T(2,:),color(i))
end
xlabel('x_i'), ylabel('y')
legend('soluzione','0.1','0.05','0.025','0.01')

figure, hold on
for i=1:length(h)
    plot(SOL{i}(1,:),err{i},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')
```

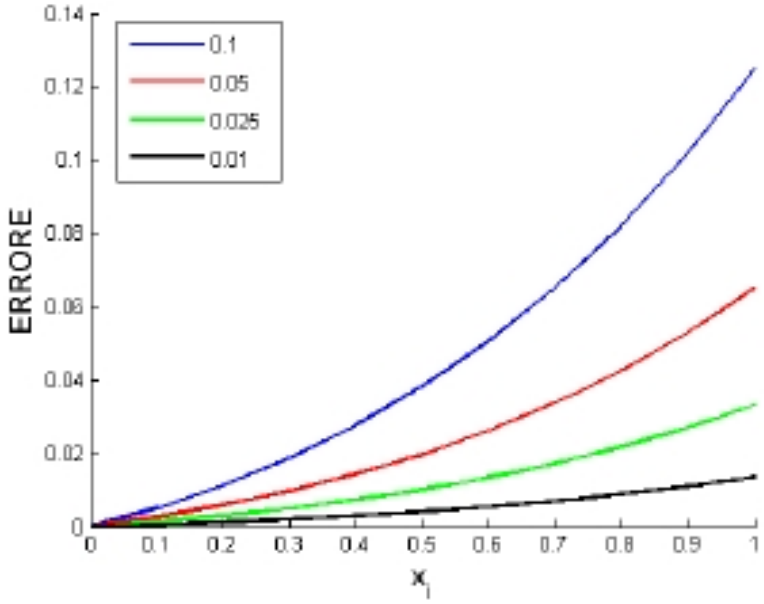
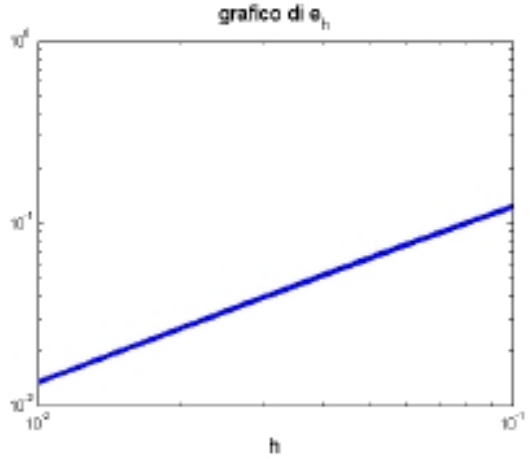


Grafico dell'**errore di approssimazione** del metodo di Eulero al variare del passo di discretizzazione h.



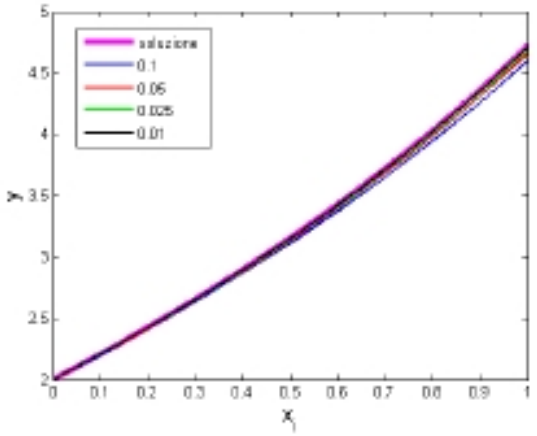


Grafico del **massimo dell'errore di approssimazione** al variare del passo di discretizzazione h e confronto delle soluzioni prodotte.

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 ($y \in C^{m+1}[x_0, x_n]$)

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

Schema di Taylor di ordine m

Trascurando l'errore e sostituendo i valori approssimati y_i 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

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

$f^{(k)}(x, y)$ indica la k-esima derivata della funzione composta, che può essere valutata ricorsivamente mediante la seguente formula

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

Nota. Per $m = 1$ si ottiene il **Metodo di Eulero.**

Metodi di Runge-Kutta

I metodi di **ordine superiore** al primo con $m \geq 2$ richiedono una certa regolarità di f e la valutazione delle sue derivate ad ogni singolo passo. Una valida alternativa è data dalla classe dei **metodi di Runge-Kutta espliciti** che 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 $[x_i, x_i + h]$.

Relazione esatta

$$y(x_i + h) = y(x_i) + h\,\Phi(x_i, y(x_i); h; f) + R(x_i, y(x_i); h; f)$$

con

$$\begin{cases} \Phi(x, y; h; f) = \displaystyle\sum_{l=1}^{r} a_l\, k_l(x, y) \\ \\ k_1(x, y) = f(x, y) \\ k_l(x, y) = f\left(x + \lambda_l\, h, y + h \displaystyle\sum_{j=1}^{l-1} b_{lj}\, k_j(x, y)\right) \qquad l = 2, \ldots, r \end{cases}$$

I **parametri** a_l, λ_l e b_{lj} sono determinati in modo che il metodo abbia **ordine** $p \Rightarrow \quad R(x_i, y(x_i); h; f) = O(h^{p+1})$

Metodi di Runge-Kutta del secondo ordine

Relazione esatta

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

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

si vogliono determinare i parametri a_1, a_2, λ in modo che il metodo sia del secondo ordine $\Rightarrow R(x, y; h; f) = O(h^{2+1})$

Sviluppo in serie di Taylor di

$$\begin{aligned} y(x+h) &= y(x) + y'(x)\, h + \frac{1}{2} y''(x)\, h^2 + 0(h^3) = \\ &= y(x) + \underbrace{f(x, y)\, h + \frac{1}{2}\left(f_x(x, y) + f_y(x, y)\, f(x, y)\right) h^2}_{h\Phi(x, y; h; f)} + O(h^3) \end{aligned}$$

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

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

Affinché il metodo sia del **secondo ordine**, il secondo membro deve ridursi a $O(h^3) \Rightarrow$ devono essere **nulli** i termini in h e h^2

$$\Downarrow$$

$$
\begin{cases}
1 - a_1 - a_2 = 0 \\
\\
\dfrac{1}{2} - a_2\,\lambda = 0
\end{cases}
\qquad \Rightarrow \quad \textbf{infinite soluzioni}
$$

Metodo di Heun

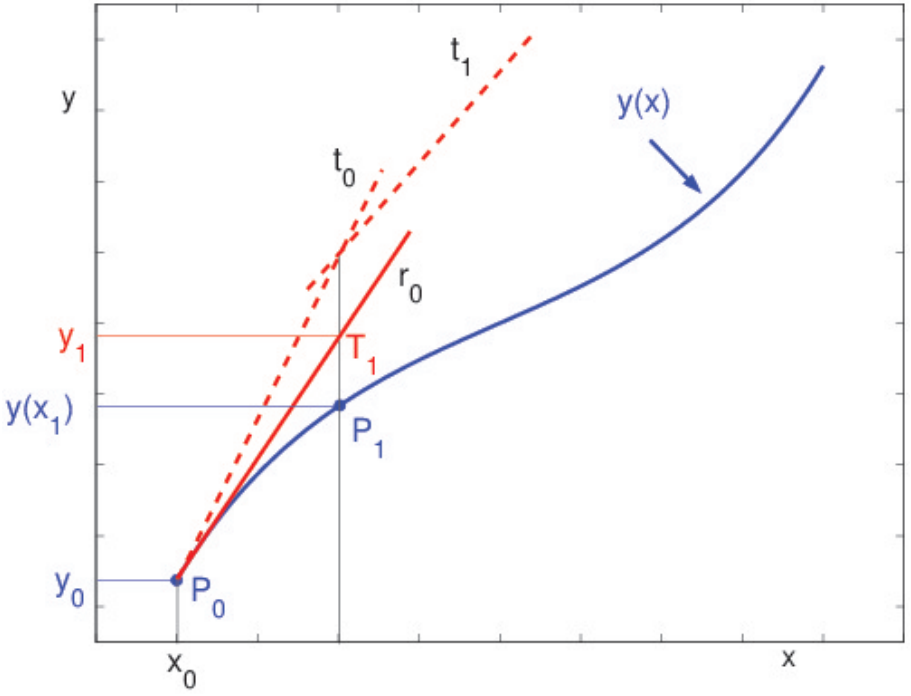
$$a_1 = a_2 = \frac{1}{2} \qquad \lambda = 1$$

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

Errore locale di trocamento:

$$R(x, y; h; f) = O(h^3)$$

$\Rightarrow$ **Secondo ordine**



Metodo di Runge-Kutta classico

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

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

Errore locale di trocamento:

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

Metodi di ordine superiore al primo

Un altro modo di derivare metodi numerici per la soluzione di equazioni differenziali ordinarie si basa sull'utilizzo di un **metodo implicito** costruito integrando **localmente** l'equazione differenziale $y'(x) = f(x, y(x))$:

$$\Rightarrow \quad y(x_{i+1}) - y(x_i) = \int_{x_i}^{x_{i+1}} y'(x)\,dx = \underbrace{\int_{x_i}^{x_{i+1}} f(x, y(x))\,dt}_{\textbf{formula di quadratura}}$$

L'integrale al secondo membro si valuta con una formula di quadratura, ad esempio:

Metodo (implicito) di Crank-Nicolson:
si ottine utilizzando la **formula dei trapezi**

$$\int_{x_i}^{x_{i+1}} f(x, y(x))\,dt \approx \frac{h}{2}\left(f(x_{i+1}, y(x_{i+1})) + f(x_i, y(x_i))\right)$$

$$\Rightarrow \quad y_{i+1} = y_i + \frac{h}{2}\left(f(x_{i+1}, y_{i+1}) + f(x_i, y_i)\right)$$

Esempio

Problema di Cauchy:

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

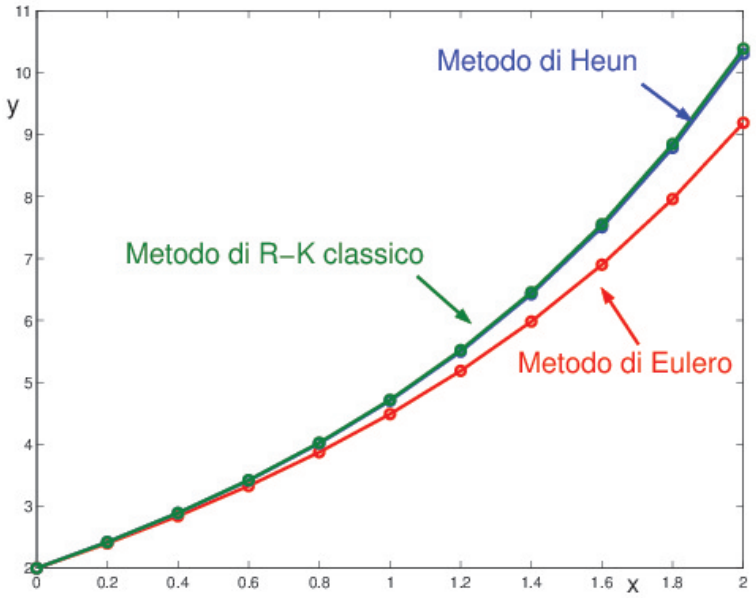


Grafico della soluzione per $h = 0.2$

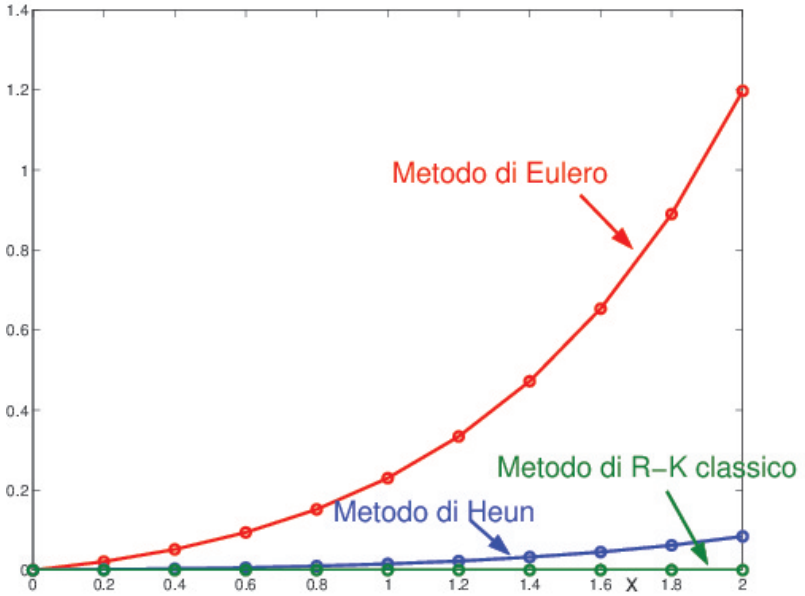


Grafico dell'errore

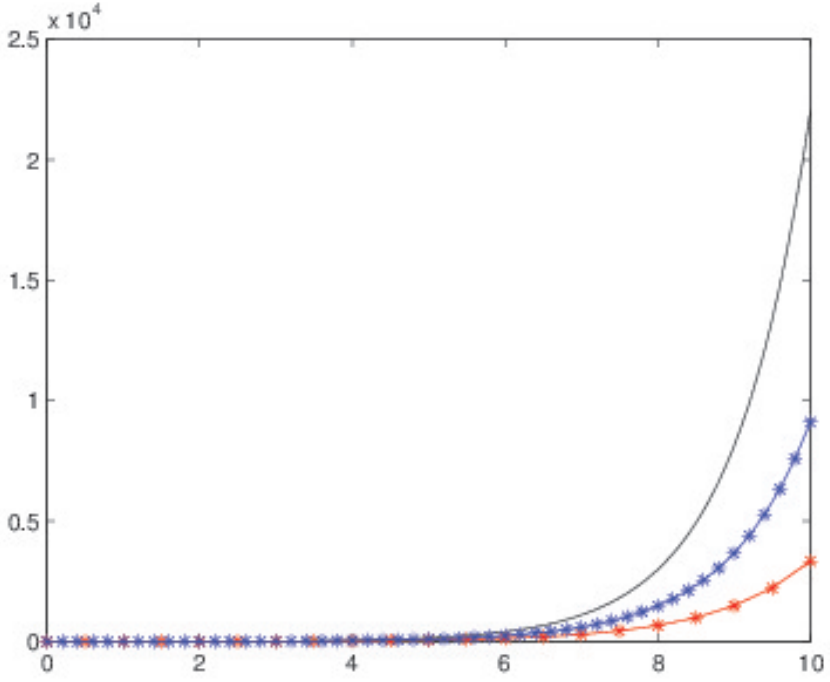


Grafico della soluzione
per $h = 0.2$ e $h = 5$

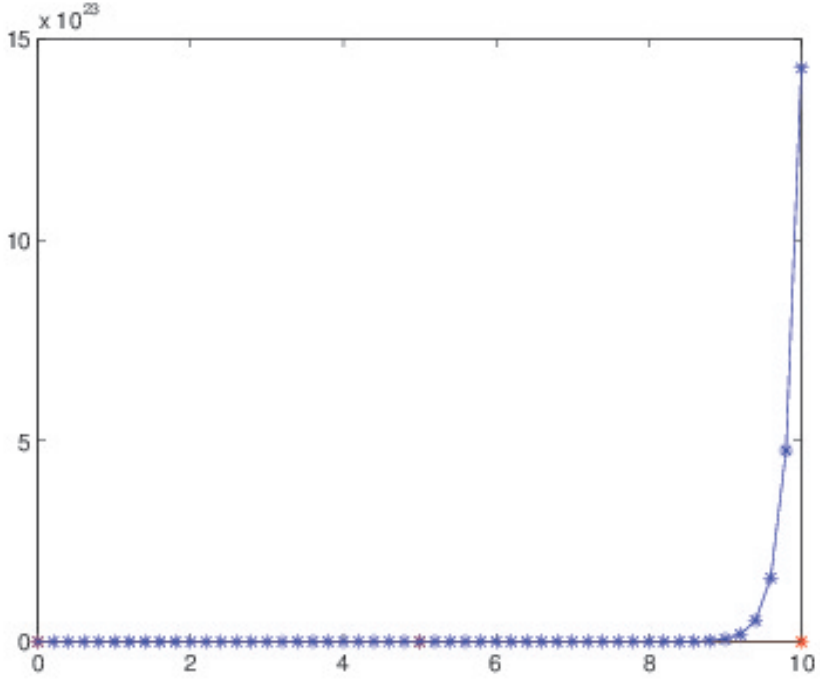


Grafico della soluzione
per $f(x, y) = 10y - x$

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:

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

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

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

Metodo di Runge-Kutta classico: $\Phi(x, y; h; f) = \frac{1}{6}\left[k_1 + 2k_2 + 2k_3 + k_4\right]$

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 $y(x)$:

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

e 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

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

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

Convergenza

A parte il primo valore $y(x_0)$ che è dato non si dispone dei valori $y(x_i) \Rightarrow$ nel calcolo dell'approssimazione y_i influiscono non solo l'omissione del termine $R(x_i, y(x_i); h; f)$ ma anche l'uso dei valori approssimati y_i al posto dei valori $y(x_i)$. Si ha quindi un **accumulo degli errori locali di troncamento** e questo porta a definire

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

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

Teorema di equivalenza di Lax.
Dato un problema differenziale lineare ben posto, uno **schema numerico alle differenze** è **convergente** se e solo se è **consistente** e **stabile**

Consistenza: l'**errore locale di troncamento** $\to 0$ per $h \to 0$

Stabilità : l'**accumularsi** degli errori locali di troncamento si mantiene **limitato** per $h \to 0$ e x fissato

Consistenza dei metodi one-step espliciti

Consistenza: $\lim_{h\to 0} \tau(x, y(x); h; f) = 0$

$$\lim_{h\to 0} \tau(x, y; h; f) = \lim_{h\to 0} \left[\frac{y(x+h) - y(x)}{h} - \Phi(x, y(x); h; f) \right]$$

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

$\Rightarrow$ Un metodo one-step esplicito è **consistente** se e solo se

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

In particolare, sono **consistenti** i metodi di **Eulero**, **Heun** e **Runge-Kutta classico**.

metodo di Eulero.

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

metodo di Heun.

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

Stabilità dei metodi one-step espliciti

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

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

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

$$\tau(x_i, y(x_i); h; f) = O(h^p) \Rightarrow |\tau(x_i, y(x_i); h; f)| \leq C_p\, h^p$$

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

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

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

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

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

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

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

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

Poiché $|e_i| \leq t_i$, $i = 0, 1, \ldots$, 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.

La stima dell'errore globale di troncamento per $x = x_i$, denota che l'errore si amplifica (per la presenza dell'esponenziale) all'aumentare della distanza di x_i da x_0.

Nota. Per **metodo di Eulero** $\Rightarrow C_1 = M/2, \quad M = \max_{x \in I} |y''(x)|$

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 $\Phi(x, y(x); h; f) \in C^0(D)$, $D = S \times [t_0, t_0 + \beta]$, $0 < h \leq \beta$, 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 η_{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**

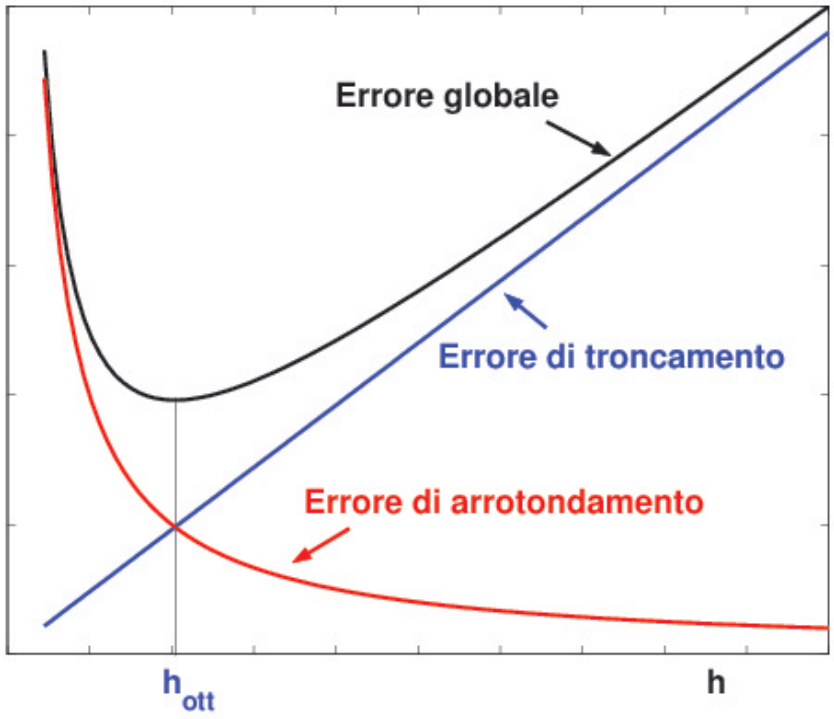
$$|e_i| \leq \frac{e^{L(x_i - x_0)} - 1}{L} \left(C_p \; h^p + \frac{\eta}{h} \right)$$

Per cui una riduzione del passo tende a far aumentare l'influenza dell'errore di arrotondamento. In particolare, per il **metodo di Eulero** si ha

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

Errore globale:

$$|e_i| \leq \frac{e^{L(x_i - t_0)} - 1}{L} \left(\frac{M}{2} h + \frac{\eta}{h} \right) \qquad M = \max_{x \in [t_0, t_0+\beta]} |y''(x)|$$



In corrispondenza del **valore ottimale** $\boxed{h_{\text{ott}} = \sqrt{\frac{2\eta}{M}}}$, dato dall'intersezione della retta $\frac{M}{2}h$ con la curva $\frac{\eta}{h}$,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:

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

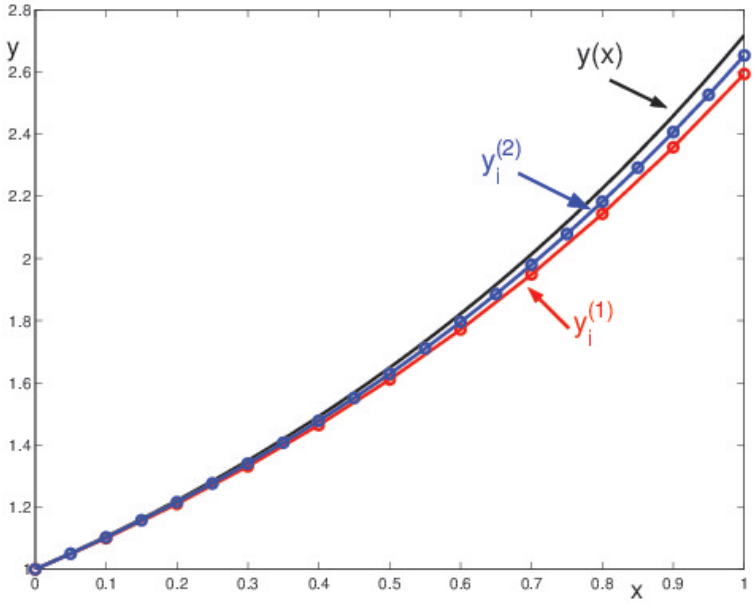
Soluzione esatta: $y(x) = e^x$

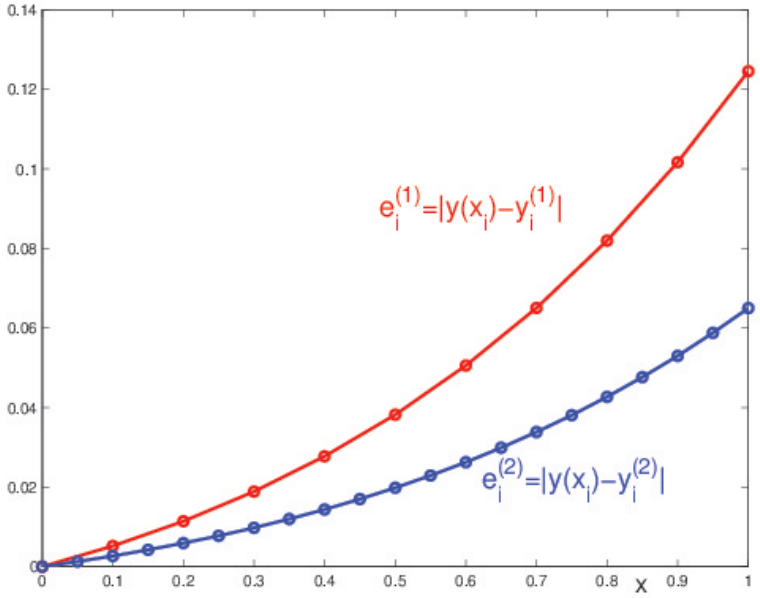
Metodo di Eulero:

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

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

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



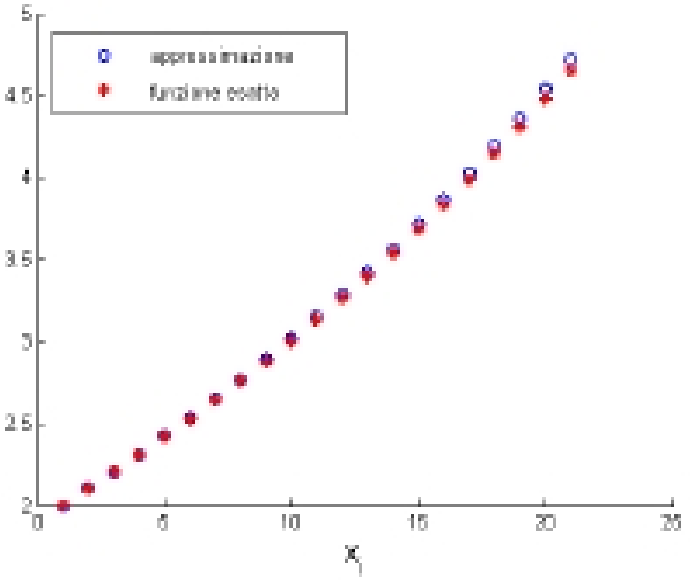


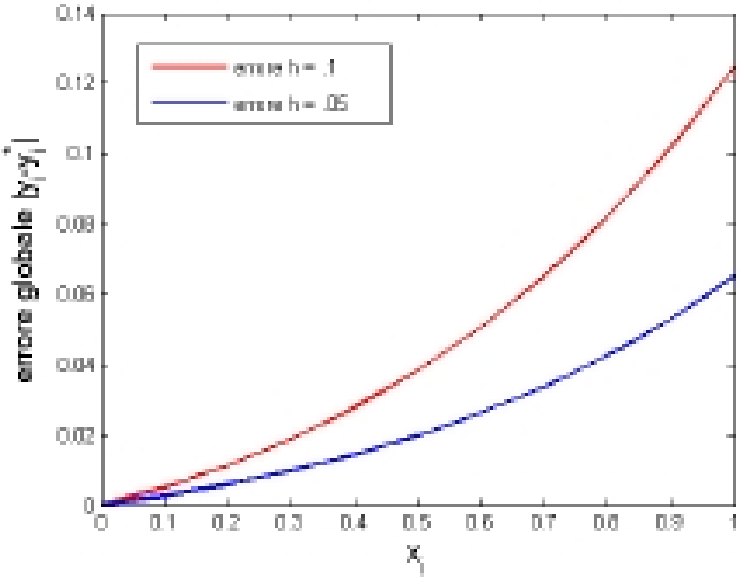
Nota. $M = \max_{x\in[0,1]} |y''(x)| = \max_{x\in[0,1]} e^x = e \simeq 2.7183$

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

Esercizio 1

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





Se volessimo stimare con quale valore del passo h l'**errore di stima** della soluzione del PC nel punto $x = 1$ è sicuramente minore di $0.5 \cdot 10^{-3}$ dovremmo considerare (trascurando gli errori di arrotondamento), per la stima dell'**errore globale di troncamento** nel punto x_i il seguente risultato

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

con $M \max_{x \in [0,1]} |y''(x)|$, L la costante di Lipschitz della funzione $f(x, y)$. Nel nostro caso poichè $x_i = 1, x_0 = 0, L = 1$ e $y''(x) = e^x$ da cui $M = e$

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

$$\Updownarrow$$

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

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

Il **metodo di Heun** è un metodo del secondo ordine, per l'**errore di troncamento globale** è dato da

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

con C_2 opportuna costante.

```
function [T]=heun(x0,y0,fun,h,n_step)
%
% [T]= heun(x0,y0,fun,h,n_step)
% Soluzione di un'equazione differenziale
% del primo ordine con il metodo di Heun
%
% INPUT
%   x0, y0: condizione iniziale
%   n_step: numero dei passi da eseguire
%   h:      passo di discretizzazione
%   fun:    espressione di f(x,y)
%
% OUTPUT
% T = matrice di dimensione 2x(n_step+1)
%     la prima riga contiene il vettore dei nodi,
%     la seconda riga contiene il vettore delle approssimazioni
%     della soluzione nei nodi
%
```

```
xi = zeros(1,n_step+1);
yi = zeros(1,n_step+1);

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];
```

```
f=@(x,y)[abs(y-x)];
y_true = @(x)[exp(x)+x+1];
h = 0.0002;
n_step = 1/h;

T_heun   = heun(0,2,f,h,n_step);
T_eulero = eulero(0,2,f,h,n_step);

y_true_val = y_true(T_heun(1,:));

ERR_heun   = abs(y_true_val-T_heun(2,:));
ERR_eulero = abs(y_true_val-T_eulero(2,:));

ERR_eulero(end)
ERR_heun(end)
```

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

Esercizio 2

Dato il Problema di Cauchy

$$\begin{cases} y'(t) = y(t) - \sin t & t \in (0,\ 1) \\ y(0) = 3 \end{cases}$$

2.1) approssimare la soluzione in $\tau = 0.1$ tramite il metodo di Eulero con passo $h_1 = 0.1$ e $h_2 = 0.05$;
2.2) sapendo che la **soluzione esatta** è data da

$$y(t) = \frac{5}{2}e^t + \frac{1}{2}\cos t + \frac{1}{2}\sin t$$

calcolare **l'errore globale** in τ nei due casi e confrontarlo con la stima teorica (si trascurino gli errori di arrotondamento);
2.3) se i valori numerici sono arrotondati alle sesta cifra decimale, calcolare il passo ottimo.

Soluzione

2.1) La funzione $f(t,y) = y - \sin t$ è definita e continua in $(0,1)$, $f_y(t,y) = 1 \Rightarrow f(t,y)$ è lipschitziana in y uniformemente rispetto a $t \Rightarrow$ esiste ed è unica la soluzione del PC.

Per $h = 0.1$ con il metodo di Eulero si ottiene

$$y_1^{(1)} = y_0 + h_1 f(t_0, y_0) = 3 + 0.1(3) = 3.3 \quad t_1 = t_0 + h_1 = 0.1$$

mentre per $h = 0.5$

$$y_1^{(2)} = y_0 + h_2 f(t_0, y_0) \qquad = 3 + 0.05(3) = 3.150000 \quad t_1^{(2)} = t_0 + h_2 = 0.05$$

$$y_2^{(2)} = y_1^{(2)} + h_2 f(t_1^{(2)}, y_1^{(2)}) \quad = 3.15 + 0.05|3.15 - \sin(0.05)| = 3.305001, \quad t_2^{(2)} = t_1^{(2)} + h_2$$

2.1) Poichè

$$y(0.1) = \frac{5}{2}e^{0.1} + \frac{1}{2}\cos(0.1) + \frac{1}{2}\sin(0.1) = 3.310346$$

l'errore globale in τ per le due soluzioni approssimate $y_1^{(1)} = 3.3$ e $y_2^{(2)} = 3.305001$ sarà rispettivamente

$$e^{(1)} = y(0.1) - y_1^{(1)} = 0.1 \cdot 10^{-1}$$

$$e^{(2)} = y(0.1) - y_2^{(2)} = 0.53 \cdot 10^{-2}$$

Per quanto riguarda una stima teorica, una stima per l'errore globale di troncamento e_i per il metodo di Eulero è data

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

con $M = \max_{t\in[0,1]} |y''(t)|$ e L la costante di Lipschitz della funzione $f(t,y)$ rispetto a y, x_i il punto in cui valutare l'errore e h il passo adottato.

Nel nostro caso $x_i - x_0 = 0.1$,

$$\begin{aligned} M &= \max_{t\in[0,1]} |y''(t)| = \max_{t\in[0,1]} |y'(t) - \cos(t)| = \\ &= \max_{t\in[0,1]} |y(t) - \sin t - \cos t| = \\ &= |y(1) - \sin(1) - \cos(1)| \approx 6.1 \end{aligned}$$

(si usa la monotonia della funzione $\frac{5}{2}e^t - \frac{1}{2}\cos(t) - \frac{1}{2}\sin(t)$).

Come costante di Lipschitz si assume

$$L = \max_{(t,y)} |f_y(t,y)| = 1$$

da cui

$$\bar{e}^{(1)} \approx 0.32 \cdot 10^{-1}$$

$$\bar{e}^{(2)} \approx 0.16 \cdot 10^{-1}$$

2.3) Supponendo che $\eta = 0.5 \cdot 10^{-}6$ si ha

$$h_{ott} = \sqrt{\frac{2\eta}{M}} \approx 0.4 \cdot 10^{-3}$$

Soluzione numerica di sistemi di equazioni differenziali

$$\begin{cases} Y'(x) = F(x, Y(x)) \\ Y(t_0) = Y_0 \end{cases} \quad \text{dove} \quad \begin{array}{l} Y(x) = [y_1(x), y_2(x), \ldots, y_n(x)]^T \\ Y_0 = [y_1(t_0), y_2(t_0), \ldots, y_n(t_0)]^T \\ F(x, Y(x)) = [f_1(x, Y(x)), \ldots, f_n(x, Y(x))]^T \end{array}$$

Metodo di Eulero:

$$\begin{cases} Y_{i+1} = Y_i + hF(x_i, Y_i) \\ \qquad\qquad (i = 0, 1, \ldots, n) \\ Y_0 = Y(t_0) \end{cases} \quad \text{dove} \quad \begin{array}{l} Y_0(X_0) = [y_1(t_0), y_2(t_0), \ldots, y_n(t_0)]^T \\ Y_i = [y_{1i}, y_{2i}, \ldots, y_{ni}]^T \\ F(x_i, Y_i) = [f_1(x_i, Y_i), \ldots, f_n(x_i, Y_i)]^T \end{array}$$

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

$$\begin{cases} y'(x) = f(x, y(x), z(x)) \\ z'(x) = g(x, y(x), z(x)) \\ y(t_0) = y_0 \\ z(t_0) = z_0 \end{cases} \Rightarrow \begin{cases} y_{i+1} = y_i + hf(x_i, y_i, z_i) \\ z_{i+1} = z_i + hg(x_i, y_i, z_i) \\ y_0 = y(t_0) \\ z_0 = z(t_0) \end{cases} \quad i = 0, 1, \ldots, n$$

Esempio 1: Modello preda-predatore

Risolvere la **coppia di equazioni differenziali non lineari**

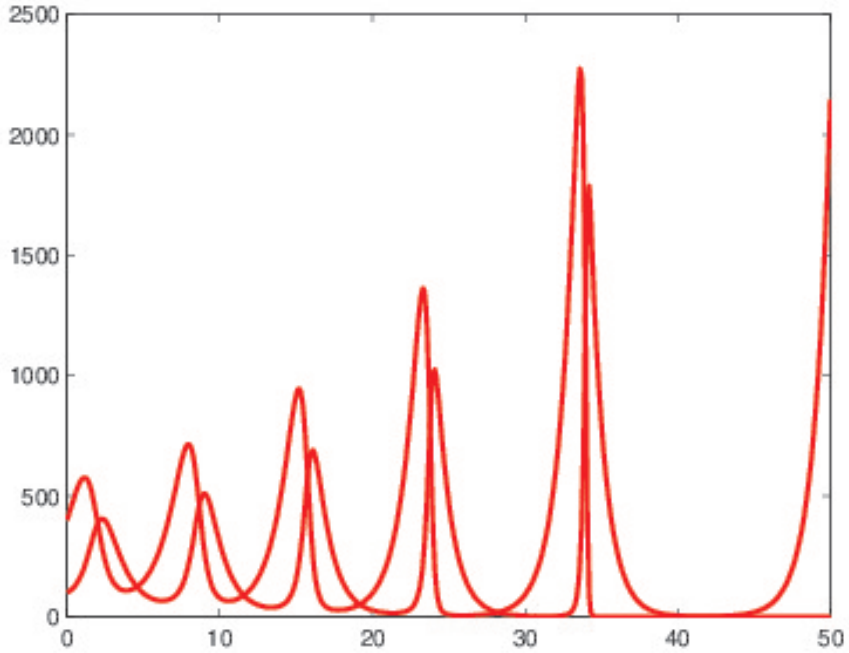
$$\begin{cases} \dot{y}_1 = k_1 \left(1 - \dfrac{y_2}{\mu_2}\right) y_1 \qquad x > 0 \\ \dot{y}_2 = -k_2 \left(1 - \dfrac{y_1}{\mu_1}\right) y_2 \\ y_1(0) = y_{10} \qquad y_2(0) = y_{20} \end{cases}$$

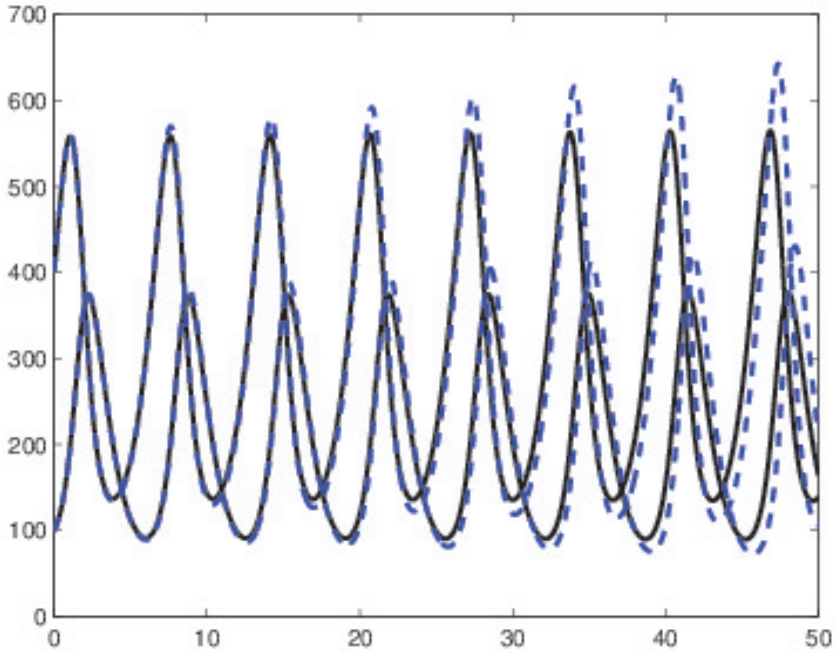
in cui $y_1(x)$ rappresenta il **numero di prede** e $y_2(x)$ rappresenta il **numero di predatori**.

Si assuma $k_1 = k_2 = 1$, $\mu_1 = 300$, $\mu_2 = 200$, $y_{10} = 400$, $y_{20} = 100$.

Nota. Per questi valori di μ_1, μ_2 la soluzione è **periodica**.

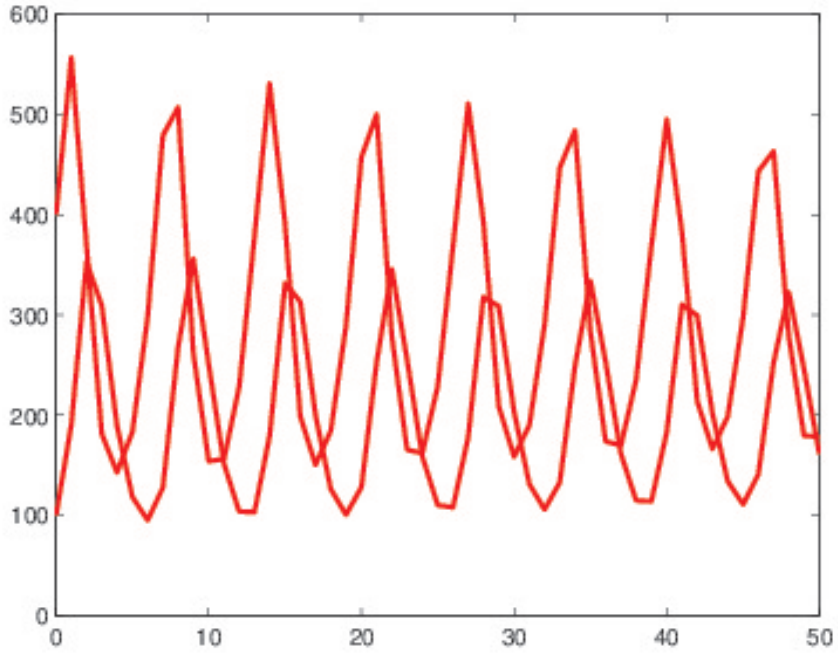
Metodo di Eulero

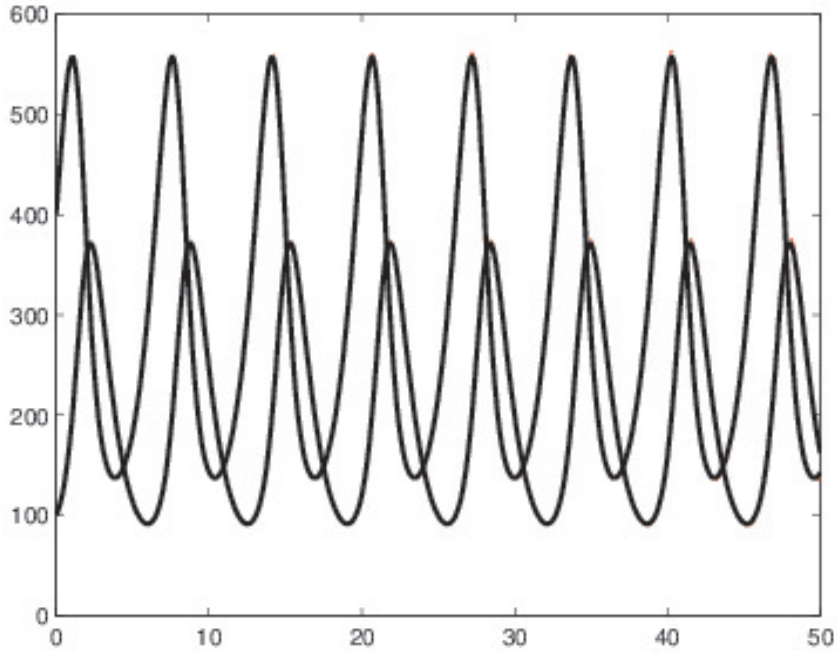




Passi: $h = 0.1$ (**grafico a sinistra**), $h = 0.001$ (grafico a destra, **linea continua**), $h = 0.01$ (grafico a destra, **linea tratteggiata**)

Metodo di Runge-Kutta classico





Passi: $h = 0.1$ (**grafico a sinistra**), $h = 0.01$ (**grafico a destra**)

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')
```

Riferimenti bibliografici

L. Gori, *Calcolo Numerico*: Cap. 9 §§ **9.1-9.6**

L. Gori, M.L. Lo Cascio, *Esercizi di Calcolo Numerico*: **6.1, 6.2, 6.4, 6.5, 6.6, 7.76, 7.80, 7.85**