SIS Scuola Interdipartimentale delle Scienze, dell'Ingegneria e della Salute

DiST Dipartimento di Scienze e Tecnologie

Laurea Magistrale in STN

Applicazioni di Calcolo Scientifico e Laboratorio di ACS

(12 cfu)

prof. Mariarosaria Rizzardi

Centro Direzionale di Napoli – Isola C4
stanza: n. 423 – Lato Nord, 4° piano
tel.: 081 547 6545
email: mariarosaria.rizzardi@uniparthenope.it

Argomenti trattati

- **Calcolo Numerico in MATLAB:**
  - **Ordinary Differential Equations (ODE)**
    - **Initial Value Problems (IVP)**
    - **Boundary Value Problems (BVP)**

Equazioni Differenziali Ordinarie (ODE)

Una **ODE** (**O**rdinary **D**ifferential **E**q.) è un'equazione contenente una o più derivate di una funzione incognita $y(t)$, di una sola variabile indipendente t.

Se le variabili indipendenti sono più di una si parla di **PDE** (**P**artial **D**ifferential **E**q.).

L'**ordine dell'equazione differenziale** è il massimo ordine di derivata che compare nell'equazione.

Esempio

solo t come variabile indipendente

$$y'(t)+2y(t)-3t=0 \qquad \textbf{ODE}$$

t e x come variabili indipendenti

$$\left.\begin{aligned} & y_t(t,x)-y_x(t,x)=1 \\ & \frac{\partial y}{\partial t}(t,x)-\frac{\partial y}{\partial x}(t,x)=1 \end{aligned}\right\} \quad \textbf{PDE}$$

Una **ODE** si dice in **forma normale** quando è risolta rispetto alla derivata massima:

$$y^{(n)}(t)=\phi(t,\, y,\, y',\, \ldots,\, y^{(n-1)})$$

Equazioni Differenziali Ordinarie (ODE)

In un problema a valori iniziali o di Cauchy (**Initial Vaue Problem - IVP**) l'ODE è risolta partendo da uno stato iniziale noto o *condizioni iniziali* (IC - *Initial conditions*). Di solito la variabile indipendente rappresenta il tempo t: quindi la soluzione dell'ODE descrive un fenomeno evolutivo, al variare del tempo.

Esempio: Legge del raffreddamento di Newton

$$\begin{cases} y'(t) = \alpha(y - T_{\text{amb}}), & t \in]0, t_{\max}[\\ y(0) = y_0 & -1 \le \alpha \le 0 \end{cases}$$

"Il tasso di perdita di calore di un corpo è proporzionale alla differenza di temperatura tra il corpo e l'ambiente circostante."

soluzione analitica: $y(t) = T_{\text{amb}} - e^{\alpha t}(T_{\text{amb}} - y_0)$

In un IVP, a partire dalle condizioni iniziali la soluzione numerica è calcolata iterativamente: nel primo passo, queste consentono di iniziare il processo, in ogni passo successivo viene calcolata la soluzione in un nuovo istante a partire dai risultati calcolati in un certo numero di passi precedenti.

In un problema a valori al contorno (**Boundary Vaue Problem - BVP**) la soluzione dell'ODE descrive un fenomeno stazionario (steady state) date delle condizioni alla frontiera (BC - *Boundary Conditions*). Di solito, la variabile indipendente rappresenta lo spazio x.

Esempio: Soluzione steady-state dell'eq. del calore

$$\begin{cases} y''(x) = 0, & x \in]0, L[\\ y(0) = y_0, \quad y(L) = y_L \end{cases}$$

soluzione analitica: $y(t) = y_0 - x(y_0 - y_L)/L$

ODE

Una ODE del 1° ordine è detta

- **esplicita** se l'equazione è del tipo: $y'(t) = \phi(t,y)$;
- **implicita** se l'equazione è del tipo: $\phi(t,y,y') = 0$.

Se $\partial\phi/\partial y'$ risulta non singolare, allora si può rendere esplicita l'ODE; quando invece $\partial\phi/\partial y'$ è singolare, l'equazione è detta DAE (differential-algebraic equation).

Analogamente per ODE di ordine superiore.

Esempi

$$y' + 2y - 3x = 0 \iff y' = -2y + 3x$$ ODE esplicita

$$y = \log[25 + (y')^2]$$ ODE implicita

$$(y')^2 + 4xy' + 2x^2 = 2y$$ ODE implicita

IVP: Teor. di esistenza ed unicità

Per un Problema di Cauchy $\begin{cases} y'(t) = f(t,y), & t \in]a,b[\\ y(a) = y_0 \end{cases}$

il **Teor. di Cauchy-Peano*** assicura l'esistenza e l'unicità della soluzione nelle ipotesi:

* detto anche *Teor. di Picard-Lindelöf* o *di Cauchy-Lipschitz*

- ★ $f(t,y)$ continua $\forall t \in]a,b[, \forall y \in \mathbb{R}$
- ★ $f(t,y)$ derivabile rispetto a y
- ★ $\partial f/\partial y$ limitata.

$$\| f(t,y_1) - f(t,y_2) \| \le L \| y_1 - y_2 \|$$

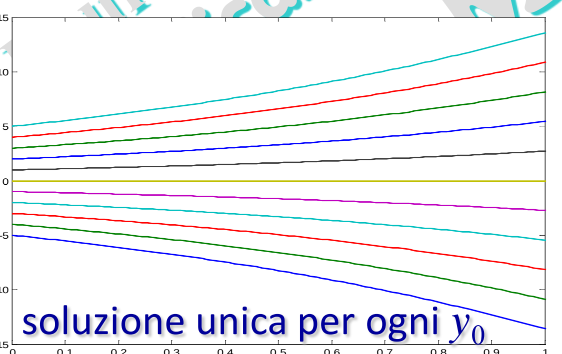
cond. di Lipschitz di f rispetto a y

Esempi

$$\begin{cases} y' = y & t \in]0,1[\\ y(0) = y_0 \end{cases}$$

continue e derivabili in $\mathbb{R}^2$

$$f(t,y) = y, \quad \frac{\partial f}{\partial y}(t,y) = 1$$



$$\begin{cases} y' = 2\sqrt{|y|} \\ y(0) = 0 \end{cases}$$

non è limitata per $y=0$

$$f = 2\sqrt{|y|}, \quad \frac{\partial f}{\partial y} = \frac{\text{signum}(y)}{\sqrt{|y|}}$$

$\forall c \ge 0$ le infinite funzioni definite come

$y_c(t) = 0$ per $t \le c$

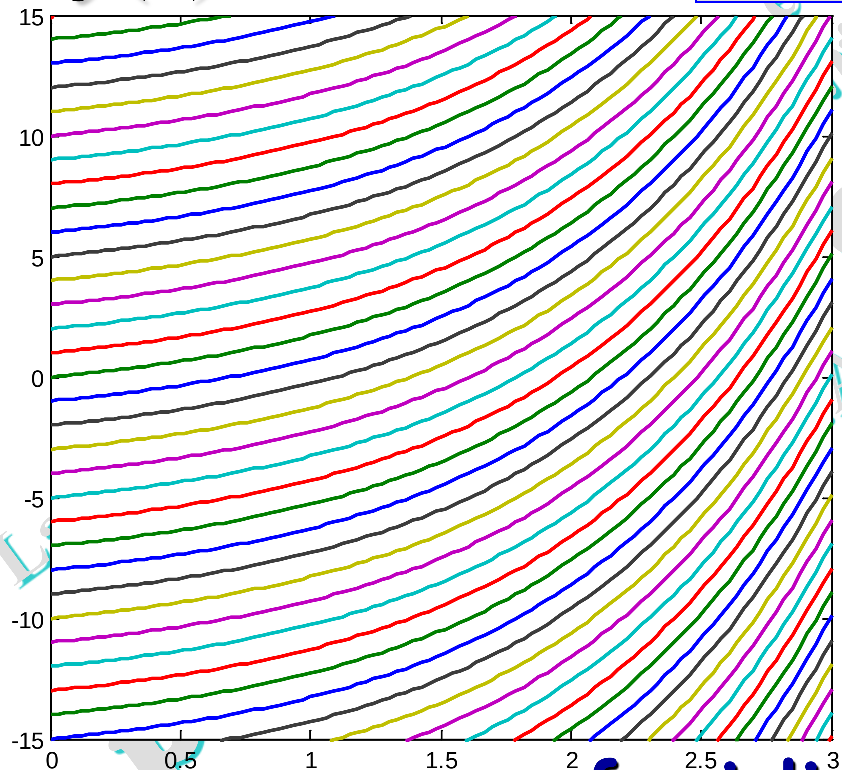
$y_c(t) = (t-c)^2$ per $t \ge c$

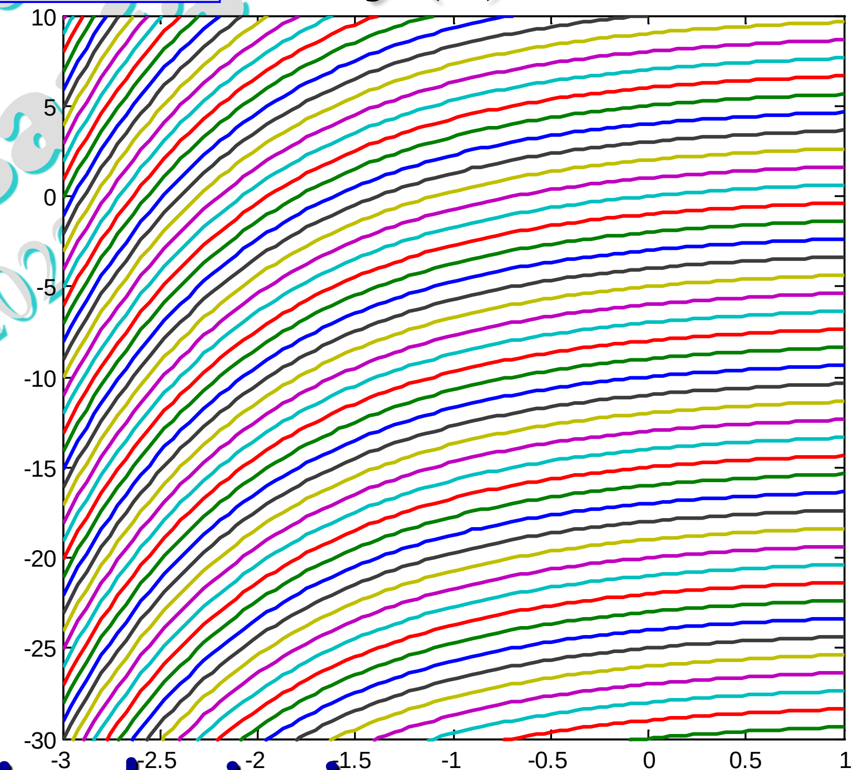
sono soluzioni dell'IVP

Una ODE per cui vale il *T. di Cauchy-Peano* ha ∞ soluzioni: ciascuna è individuata da una particolare condizione iniziale

equazione	famiglia di soluzioni
$y' = \pm y$	$y(x) = C\, e^{\pm x}$
$y' = e^{\pm x}$	$y(x) = C \pm e^{\pm x}$

$y(x) = C + e^{+x}$ ⟵ **costante additiva** ⟶ $y(x) = C - e^{-x}$



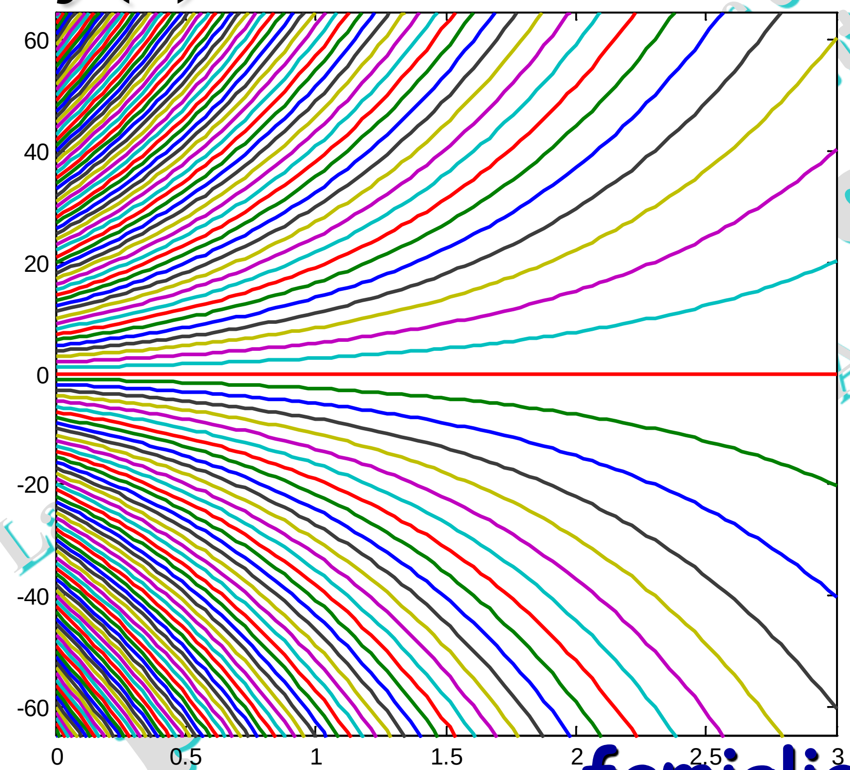


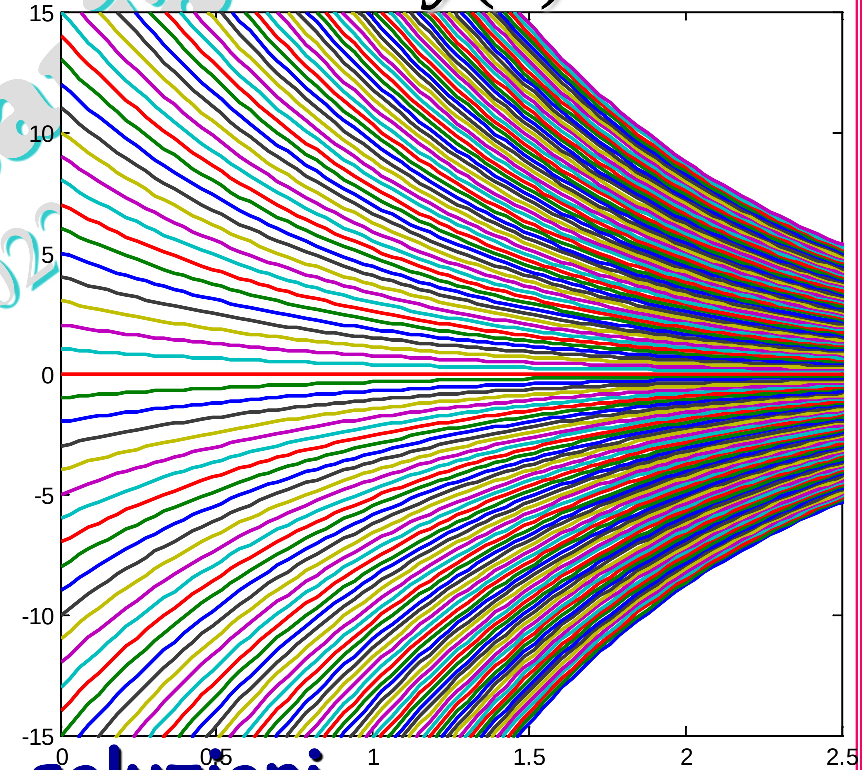
famiglia di soluzioni

Una ODE per cui vale il *T. di Cauchy-Peano* ha ∞ soluzioni: ciascuna è individuata da una particolare condizione iniziale

equazione	famiglia di soluzioni
$y' = \pm y$	$y(x) = C\, e^{\pm x}$
$y' = e^{\pm x}$	$y(x) = C \pm e^{\pm x}$

$y(x) = C\, e^{+x}$ ⟵ **costante moltiplicativa** ⟶ $y(x) = C\, e^{-x}$





famiglia di soluzioni

Sistemi di ODE

Il Problema di Cauchy per un sistema di ODE del 1° ordine appare come:

$$\begin{cases} \underline{y}'(t) = \underline{f}(t,\underline{y}), & t \in]a,b[\\ \underline{y}(a) = \underline{y}_0 \end{cases}$$

dove

vettori

$$\underline{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{pmatrix} \qquad \underline{y}_0 = \begin{pmatrix} y_{0_1} \\ y_{0_2} \\ \vdots \\ y_{0_n} \end{pmatrix} \qquad \underline{f}(t,\underline{y}) = \begin{pmatrix} f_1(t,y_1,y_2,\ldots,y_n) \\ f_2(t,y_1,y_2,\ldots,y_n) \\ \vdots \\ f_n(t,y_1,y_2,\ldots,y_n) \end{pmatrix}$$

Esempio: problema predatore-preda

$x = x(t)$ numero prede al tempo t

$y = y(t)$ numero predatori al tempo t

x_b, x_d tassi costanti di accrescimento e morte naturale prede ($x_b > x_d$)

$x'(t) = \alpha x - \beta xy$, $\alpha = x_b - x_d$ (α tasso di crescita), $\beta > 0$: dove αx=crescita esponenziale popolazione prede in assenza di predatori e βxy=velocità di predazione sulle prede.

$y'(t) = \delta xy - \gamma y$, $\gamma > 0$, $\delta > 0$: δxy=crescita della popolazione di predatori e γy=tasso di perdita dei predatori per morte naturale o emigrazione; esso porta ad un decadimento esponenziale in assenza di prede.

Equazioni di Lotka-Volterra

$$\begin{cases} x' = \alpha x - \beta xy \\ y' = -(\gamma y - \delta xy) \\ x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases}$$

ODE

Un'**equazione differenziale di ordine superiore al primo ed in forma normale** può ricondursi ad un ***sistema di equazioni differenziali del prim'ordine***.

$$y^{(p)} = \Psi(x, y, y', y'', \ldots, y^{(p-1)}) \quad \Longleftrightarrow \quad \begin{cases} z'_1 = z_2(x) \\ z'_2 = z_3(x) \\ \ldots \\ z'_p = \Psi(x, z_1, z_2, z_3, \ldots, z_p) \end{cases}$$

avendo posto

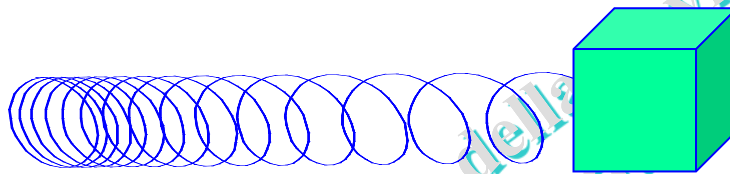
derivate successive

$$\begin{aligned} z_1(x) &= y(x) \\ z_2(x) &= y'(x) \\ z_3(x) &= y''(x) \\ &\ldots \\ z_p(x) &= y^{(p-1)}(x) \end{aligned}$$

soluzione $y(x) = z_1(x)$

Gli ODE Solver di MATLAB risolvono solo equazioni e sistemi di equazioni differenziali del 1° ordine.

Esempio: oscillatore armonico



ODE del 2° ordine

$$\begin{cases} y'' = -\dfrac{\Phi y' + qy}{m} \\ y(0) = \alpha \\ y'(0) = \beta \end{cases}$$

m massa del corpo

q, Φ costanti di proporzionalità

Sistema di ODE del 1° ordine

$$\begin{cases} z_1' = z_2 \\ z_2' = -\dfrac{\Phi z_2 + q z_1}{m} \\ z_1(0) = \alpha \\ z_2(0) = \beta \end{cases}$$

soluzione voluta

$y = z_1(t)$

ODE

Risoluzione numerica di un Problema di Cauchy

Problema: $\begin{cases} y' = f(x,y) & x \in (a, b) \\ y(a) = y_0 \end{cases}$

1. Si discretizza l'intervallo $[a,b]$:

$$a = x_0 < x_1 < x_2 < \ldots < x_n = b;$$

2. Si discretizza la **ODE** (cioè si sostituisce al problema continuo uno discreto);

3. Si determinano i valori y_i, $i = 1, 2, \ldots, n$, soluzioni del problema discreto ed approssimazioni dei valori esatti nei nodi:

$$y_i \approx y(x_i), \quad i = 1, 2, \ldots, n;$$

Risoluzione numerica di un Problema di Cauchy mediante il metodo delle differenze finite

1. Si discretizza l'intervallo $[a,b]$ in un insieme di *punti di griglia* (o *di maglia*) equispaziati o non:
$$a = x_0 < x_1 < x_2 < \ldots < x_n = b$$
2. Si discretizza la **ODE** sostituendola con una **equazione alle differenze**;
3. Si determinano i valori y_i, $i = 1, 2, \ldots, n$, soluzioni del problema discreto ed approssimazioni dei valori esatti nei nodi $y_i \approx y(x_i)$

punti di griglia x_i **equispaziati**: con ampiezza di maglia costante $h = (b-a)/n$

punti di griglia x_i **non equispaziati**: con ampiezza di maglia variabile $h_i = (x_{i+1} - x_i)$

Un'**equazione alle differenze di ordine n** è un'equazione del tipo:
$$F(k, y_k, y_{k+1}, \ldots, y_{k+n}) = 0 \ \forall k$$
che coinvolge una variabile indipendente intera k (discreta) ed una variabile dipendente y_k (*soluzione dell'eq. alle differenze*), che pertanto è una funzione discreta.

Esempio: metodo di Eulero (o m. delle tangenti)

Problema:
$$\begin{cases} y' = f(x,y) & x \in (a,b) \\ y(a) = y_0 \end{cases}$$

1 Si discretizza l'intervallo $[a,b]$: $x_k = a + k\,h, \quad k = 0,1,2,\ldots,n$

2 Si discretizza la derivata mediante il rapporto incrementale:

$$y'(x) \approx [y(x+h) - y(x)]/h \qquad \text{(con } h \text{ opportuno)}$$

equazione alle differenze

$$y(x+h) \approx y(x) + h\,f(x,y) \quad \Leftrightarrow \quad y_{k+1} = y_k + hf(x_k, y_k)$$

Formula di Eulero

3 Si applica la formula del metodo di Eulero ad ogni punto della discretizzazione a partire dal valore noto di y_0:

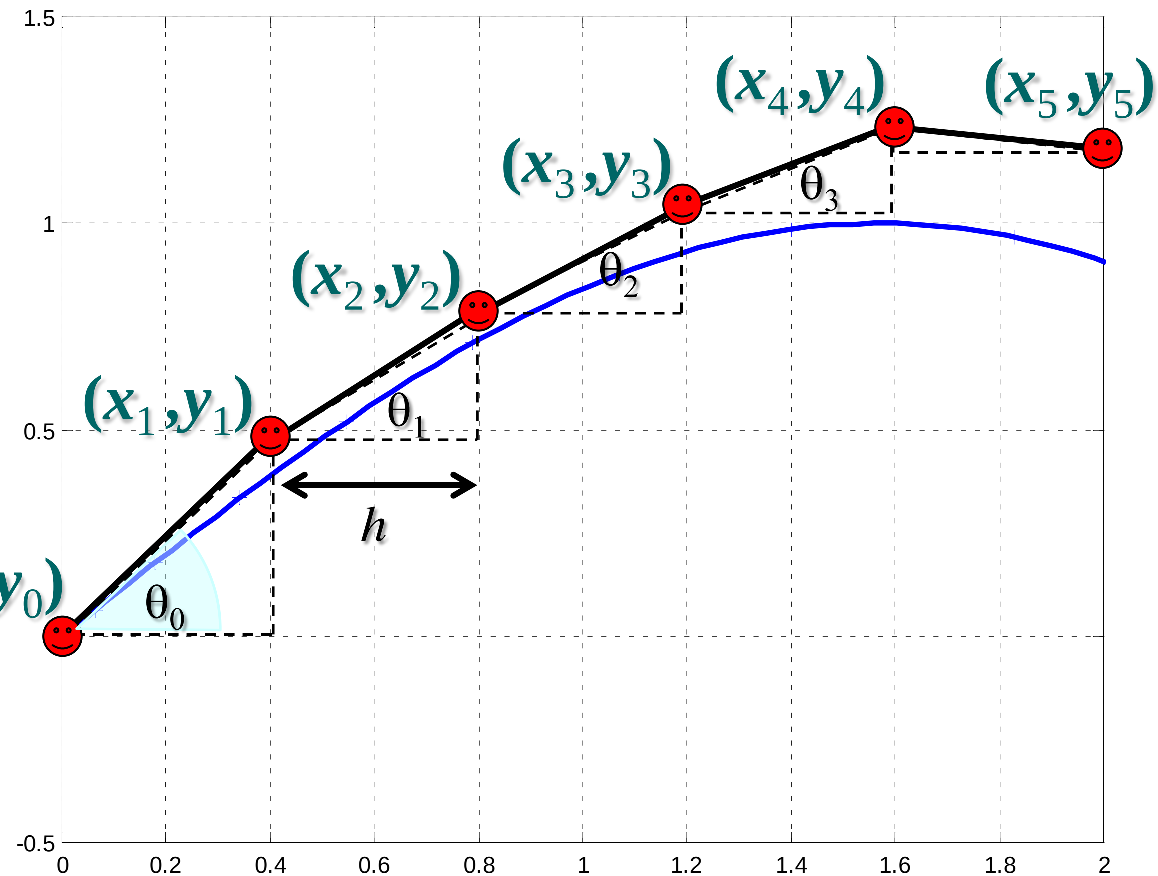
$$y_{k+1} = y_k + h\,\boxed{f(x_k, y_k)} \leftarrow \boxed{\tan(\theta_k)}$$

$$y_1 = y_0 + h\,f(x_0, y_0)$$

$$y_2 = y_1 + h\,f(x_1, y_1)$$

$$y_3 = y_2 + h\,f(x_2, y_2)$$

$\ldots$



l'errore aumenta

Interpretazione geometrica del metodo di Eulero

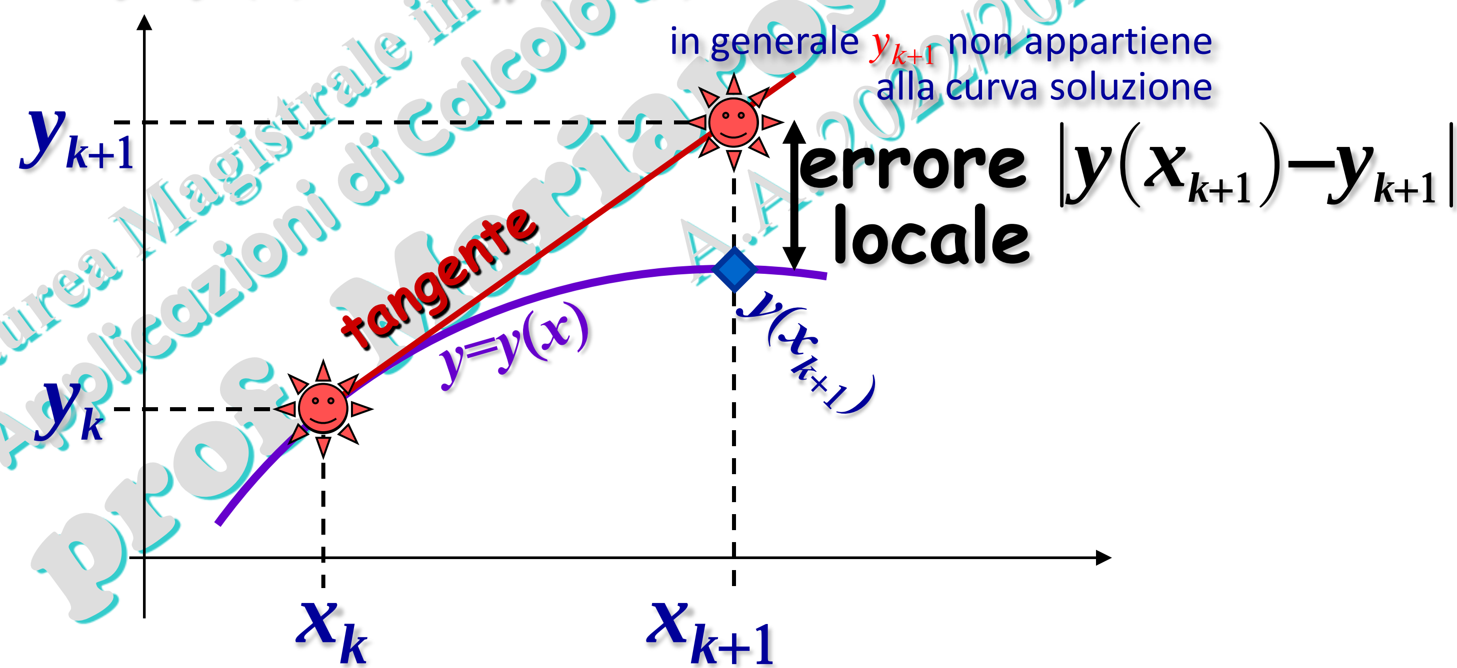
$$\boxed{y_{k+1} = y_k + h\,f(x_k, y_k)}$$

In un passo di applicazione del metodo: y_{k+1} è l'ordinata del punto, di ascissa x_{k+1}, sulla **retta**

$$y = y_k + y'(x_k, y_k)(x - x_k)$$

Serie di Taylor troncata ai termini del 1° ordine

passante per (x_k, y_k) e parallela alla tangente* alla curva $y=y(x)$ in $(x_k, y(x_k))$.



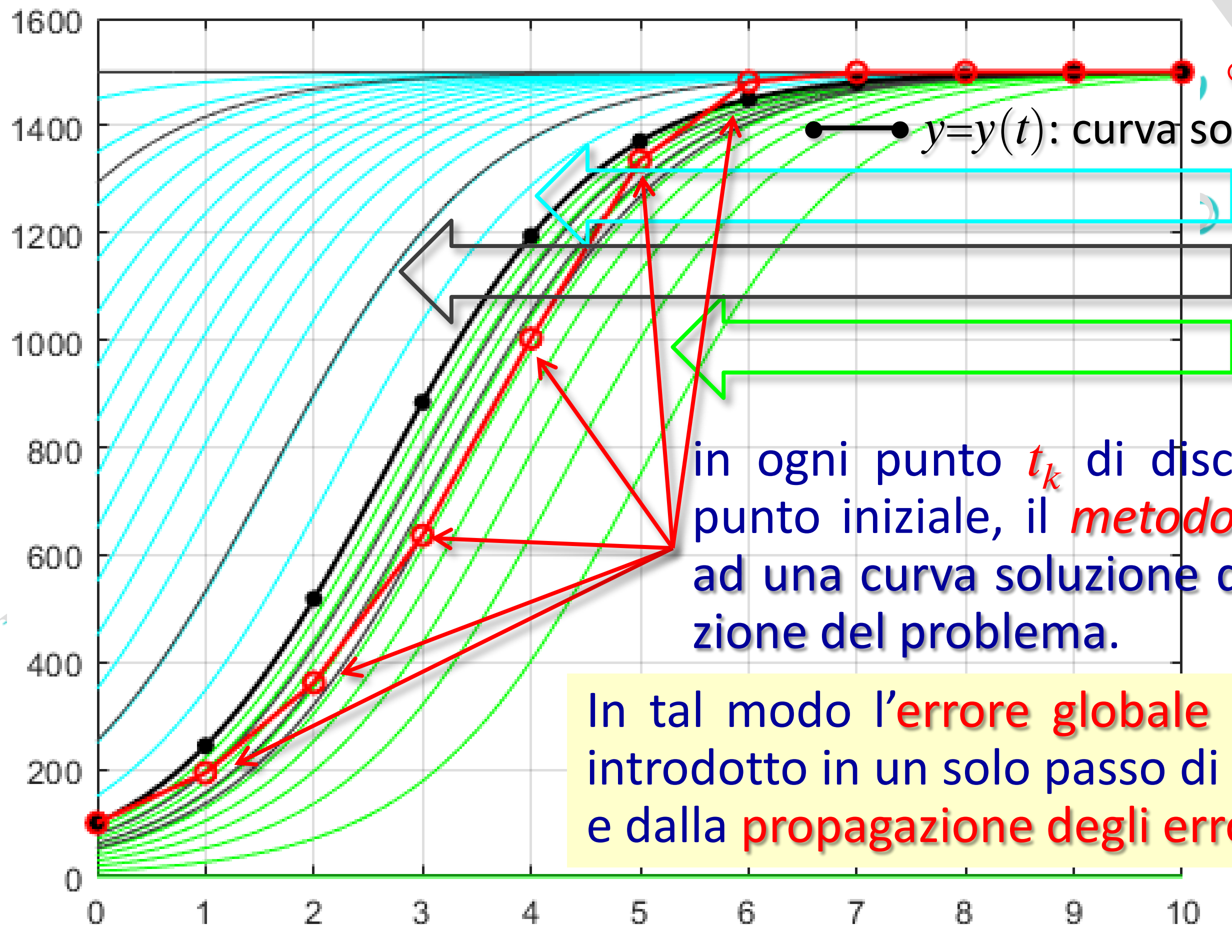
* è proprio la tangente se il punto sta sulla curva

Esempio: metodo di Eulero applicato all'eq. di Verhulst per il modello logistico di crescita di una popolazione

$$\begin{cases} y' = ry\left(1-\dfrac{y}{K}\right), & t\in\left]0,t_{\max}\right] \\ y(0)=P_0 \end{cases}$$

$y(t)$: misura la popolazione al tempo t
$P_0=100$: popolazione iniziale
$r=1$: tasso di crescita della popolazione
$K=1500$: termine asintotico della popolazione, detto "*carrying capacity*" (*capacità portante*)

soluzione analitica: $y(t)=\dfrac{K}{1+q\,e^{-rt}},\quad q=\dfrac{K}{P_0}-1$



altre curve della famiglia di soluzioni dell'ODE

in ogni punto t_k di discretizzazione, successivo al punto iniziale, il *metodo di Eulero* usa la tangente ad una curva soluzione dell'ODE **diversa** dalla soluzione del problema.

In tal modo l'errore globale dipende dall'errore locale introdotto in un solo passo di applicazione della formula e dalla propagazione degli errori dei passi precedenti.

Esempio: metodo di Eulero

problema: $\begin{cases} y' = y & x \in (0,3) \\ y(0) = 2 \end{cases}$ soluzione analitica: $y(x) = 2e^x$

1 Si discretizza l'intervallo $[0, 3]$: $x_0 = 0$, $x_k = kh$, $k = 1,2,\ldots,n$; $h = 3/n$

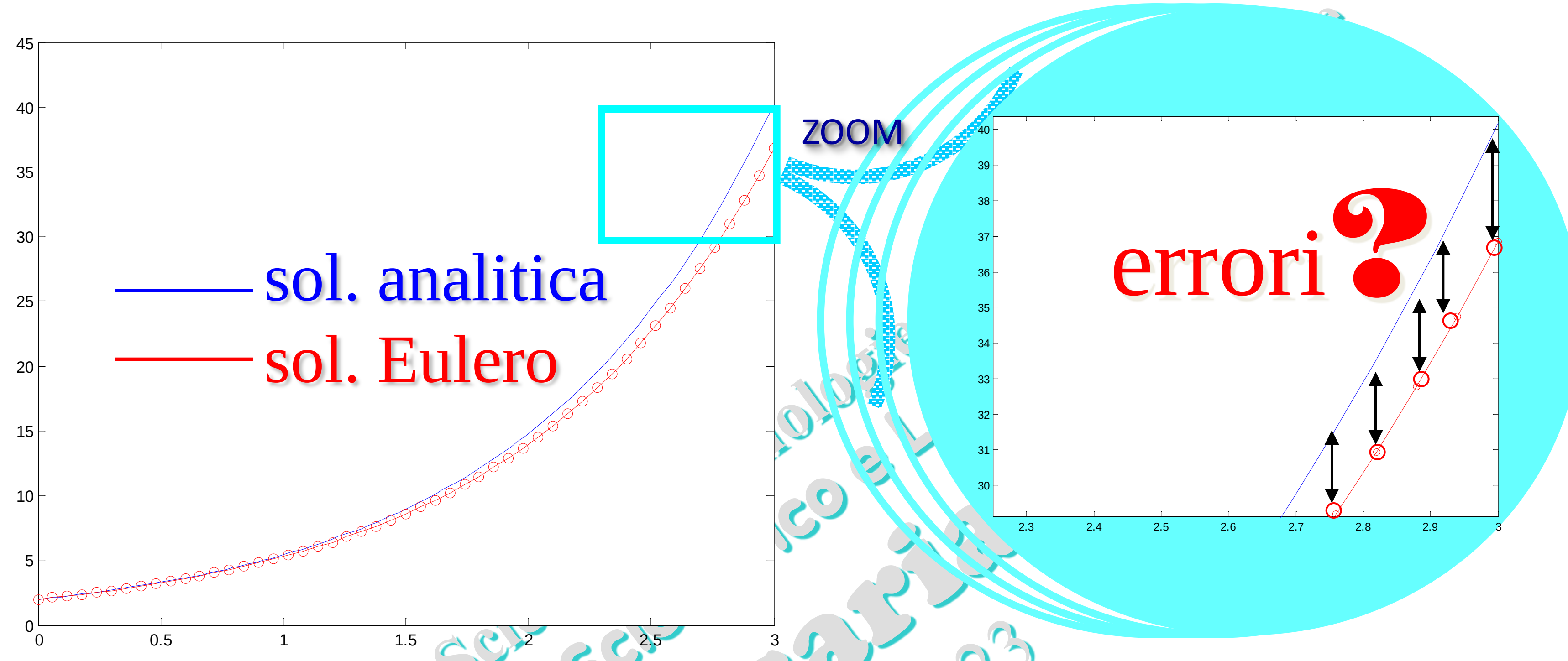
2 Si discretizza la derivata mediante il rapporto incrementale:

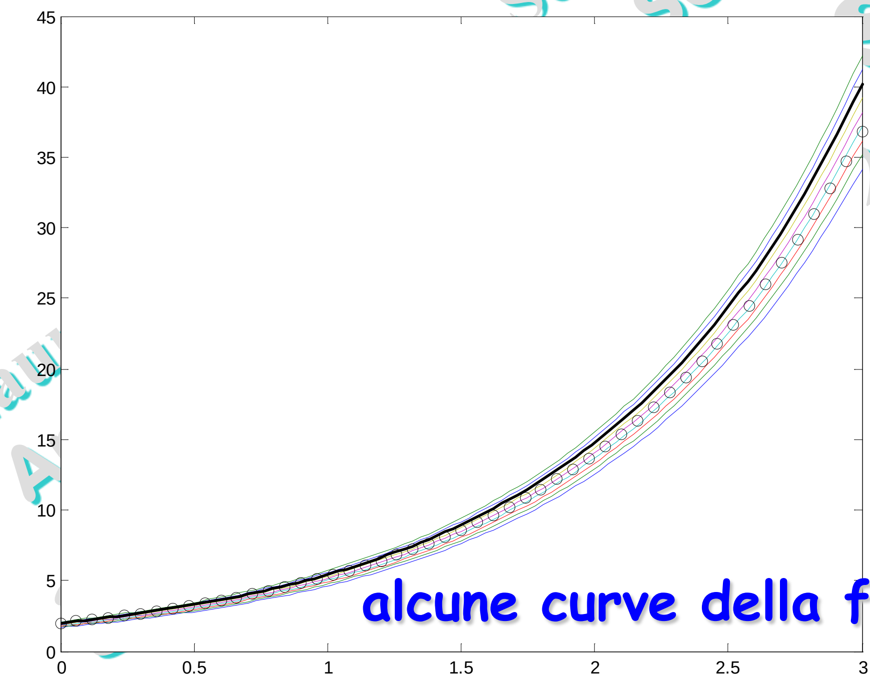
$$y'(x) \approx [y(x+h) - y(x)]/h \quad \Rightarrow \quad y(x_k + h) = y(x_k) + hf(x_k, y_k)$$

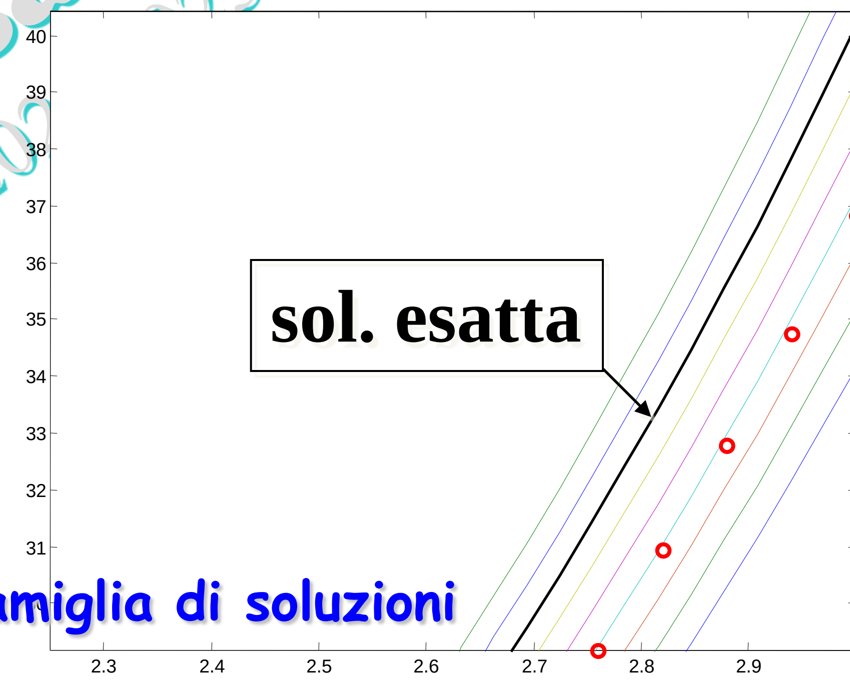
3 $$\boxed{y_{k+1} = y_k + hy}$$

```
fxy=@(x,y) y;
tspan=[0 3];
y0=2;
h=0.1;
[xi,yi]=Eulero(fxy,tspan,y0,h);
```

```
function [t,y] = Eulero(f,tspan,y0,h)
    t = (tspan(1):h:tspan(2))';
    y = zeros(size(t));
    y(1) = y0;
    for i=1:numel(y)-1
        y(i+1) = y(i) + h*f(t(i),y(i));
    end
end
```







alcune curve della famiglia di soluzioni

A causa degli errori, **ogni** punto della ***soluzione numerica*** (x_k, y_k) non appartiene alla curva della soluzione esatta, ma giace su un'altra curva della famiglia di soluzioni: quella corrispondente ad un'altra condizione iniziale

$$y_0^* \neq y_0$$

È come se (x_k, y_k) appartenesse alla soluzione della stessa ODE ma con una diversa condizione iniziale.

Pertanto lo studio dell’errore nella soluzione numerica di un ***Problema di Cauchy*** si può ricondurre alla **dipendenza** della soluzione della ODE **dalla condizione iniziale**.

Il Teorema Fondamentale del Calcolo Integrale

$$y(x_k+h)=y(x_k)+\int_{x_k}^{x_k+h} f(t,y(t))\,\mathrm{d}t$$

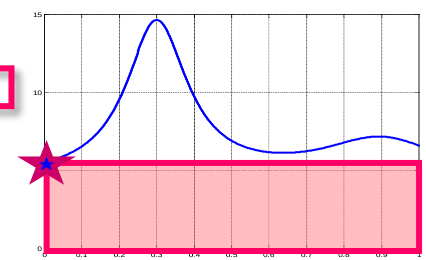
sta alla base di molti metodi numerici per le ODE: ciascuno è determinato dalla particolare formula di quadratura scelta per approssimare l'integrale.

Esempi

Il **metodo di Eulero** si ottiene applicando all'integrale la formula di quadratura rettangolare

$$\int_{x_k}^{x_{k+1}} f(t,y(t))\,\mathrm{d}t \approx hf(x_k,y_k)$$

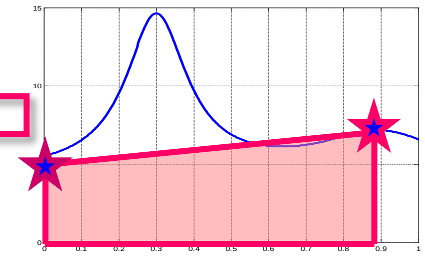
$$y_{k+1}=y_k+hf(x_k,y_k)$$



Il **metodo Trapezoidale** si ottiene applicando all'integrale la formula di quadratura trapezoidale

$$\int_{x_k}^{x_{k+1}} f(t,y(t))\,\mathrm{d}t \approx \frac{h}{2}\left[f(x_k,y_k)+f(x_{k+1},y_{k+1})\right]$$

$$y_{k+1}=y_k+\frac{h}{2}\left[f(x_k,y_k)+f(x_{k+1},y_{k+1})\right]$$



Classificazione dei metodi per le ODE

metodi one-step Il valore y_{k+1} dipende solo dal valore precedente y_k	**metodi multi-step** Il valore y_{k+1} dipende da più valori precedenti $y_k, y_{k-1}, \ldots, y_{k-p+1}$ (p passi)
metodi espliciti	**metodi impliciti**
metodi non adattativi	**metodi adattativi**

Esempi

Il **metodo di Eulero** $y_{k+1} = y_k + hf(x_k, y_k)$
è one-step, esplicito e non adattativo

Il **metodo backward Eulero (BE)** $y_{k+1} = y_k + hf(x_{k+1}, y_{k+1})$
è one-step, implicito e non adattativo

Il **metodo del punto medio** $y_{k+1} = y_{k-1} + 2hf(x_k, y_k)$
è two-step, esplicito e non adattativo

Es.: metodo Backward Eulero (BE) …

… al problema: $\begin{cases} y' = y & x \in (0,3) \\ y(0) = 2 \end{cases}$

[soluzione: $y(x) = 2e^{x}$]

1 Si discretizza l'intervallo $[0, 3]$: $x_0 = 0$, $x_k = kh$, $k=1,2,\ldots,n$; $h=3/n$

2 Si applica la formula del metodo Backward Eulero:

$$y_{k+1} = y_k + hf(x_{k+1}, y_{k+1}) \quad\Rightarrow\quad y_{k+1} = y_k + hy_{k+1}$$

$f(x,y) = y$

3 $$\boxed{y_{k+1} = \frac{y_k}{1-h}}$$

in questo caso BE diventa esplicito!

Cosa comporta un "metodo implicito"?

Esempio: problema non lineare $y' = -\dfrac{1}{2\sqrt{y}}$

metodo di Eulero $y_{k+1} = y_k - \dfrac{h}{2\sqrt{y_k}}$

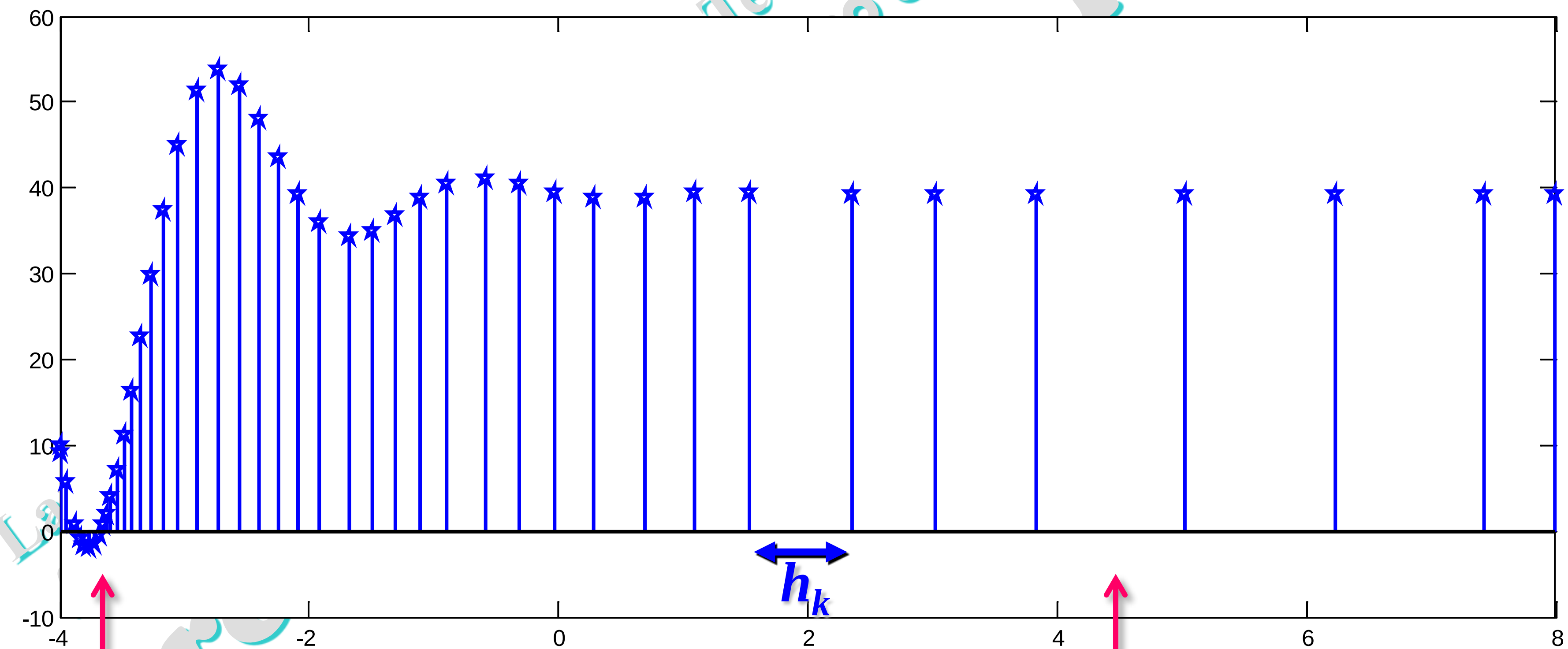
ad ogni passo si calcola un nuovo valore direttamente.

metodo BE $y_{k+1} = y_k - \dfrac{h}{2\sqrt{y_{k+1}}}$

ad ogni passo per calcolare un nuovo valore, è necessario risolvere un'**equazione non lineare in** y_{k+1}.

Che significa "metodo adattativo"?

Tutti gli **ODE Solver di MATLAB** usano un **metodo adattativo**: il passo $h_k = x_{k+1} - x_k$ varia con l'andamento della funzione

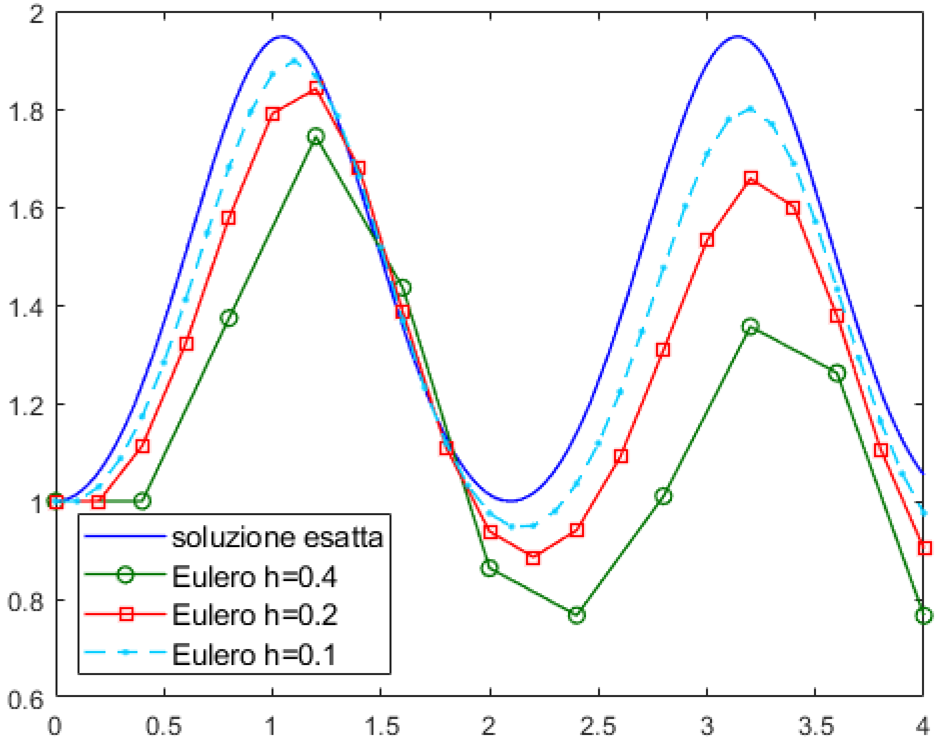


passo più piccolo dove la funzione varia rapidamente, passo più grande dove la funzione varia meno

ordine di un metodo

Esempio: metodo di Eulero applicato al problema

$$y' = y\sin 3x \quad x \in (0,4)$$
$$y(0) = 1$$



soluzione analitica:

$$y(x) = e^{\frac{1-\cos(3x)}{3}}$$

L'errore globale si riduce come h

$$E = |y_k - y(x_k)| \approx O(h)$$

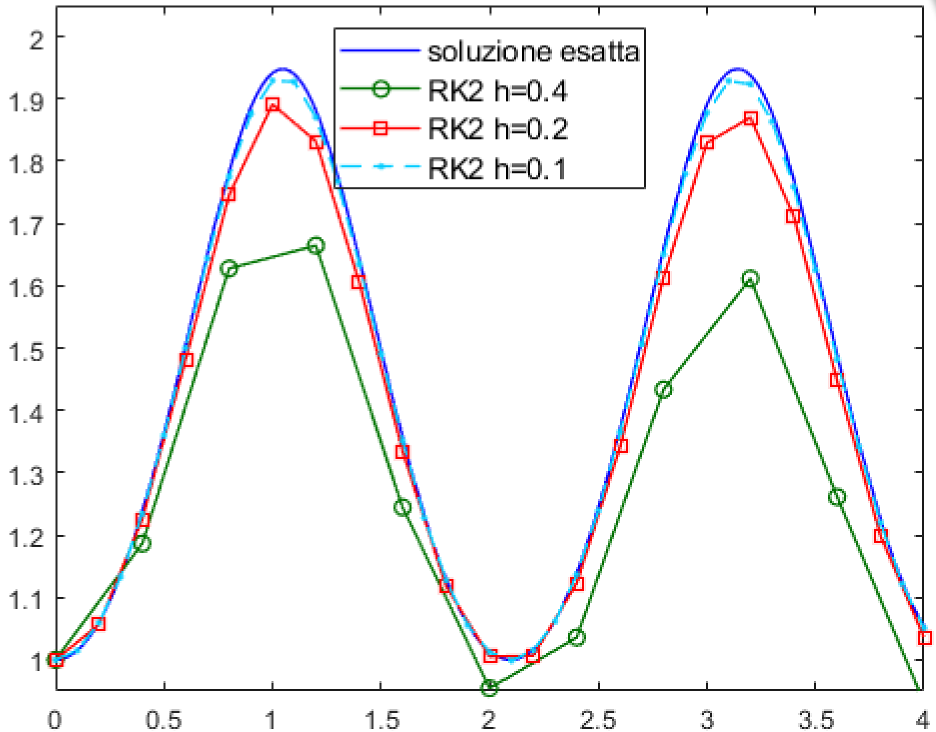
il metodo di Eulero è un metodo del primo ordine

Altri metodi

I **metodi Runge-Kutta** (**RK**) sono metodi one-step di ordine > 1

Esempio: metodo RK2

(metodo RK a 2 stadi o metodo di Heun)



$$K_1 = hf(x_k, y_k)$$

$$K_2 = hf(x_{k+1}, y_k + K_1)$$

$$y_{k+1} = y_k + \frac{1}{2}(K_1 + K_2)$$

L'errore si riduce come h^2

$$E \approx O(h^2)$$

il metodo RK2 è un metodo del **secondo** ordine

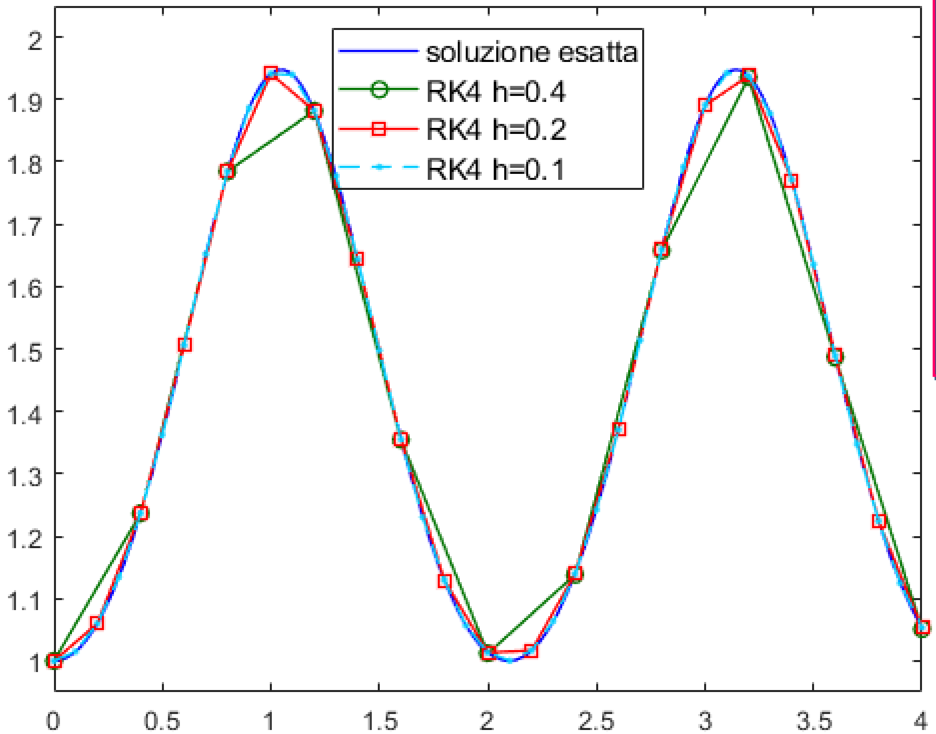
Il metodo RK esplicito più usato è **RK4** che risulta di ordine 4

Altri metodi

I **metodi Runge-Kutta** (**RK**) sono metodi one-step di ordine > 1

Esempio: metodo RK4

(metodo RK a 4 stadi)



$$K_1 = hf(x_k, y_k)$$

$$K_2 = hf\left(x_k + \frac{h}{2}, y_k + \frac{K_1}{2}\right)$$

$$K_3 = hf\left(x_k + \frac{h}{2}, y_k + \frac{K_2}{2}\right)$$

$$K_4 = hf(x_k + h, y_k + K_3)$$

$$y_{k+1} = y_k + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

L'errore si riduce come h^4

$$E \approx O(h^4)$$

il metodo RK4 è un metodo del **quarto** ordine

Errore globale

Se $y(x_i)$ è il valore esatto e y_i è la sua approssimazione numerica, l'errore globale

$$E_i = |y_i - y(x_i)|$$

dipende:

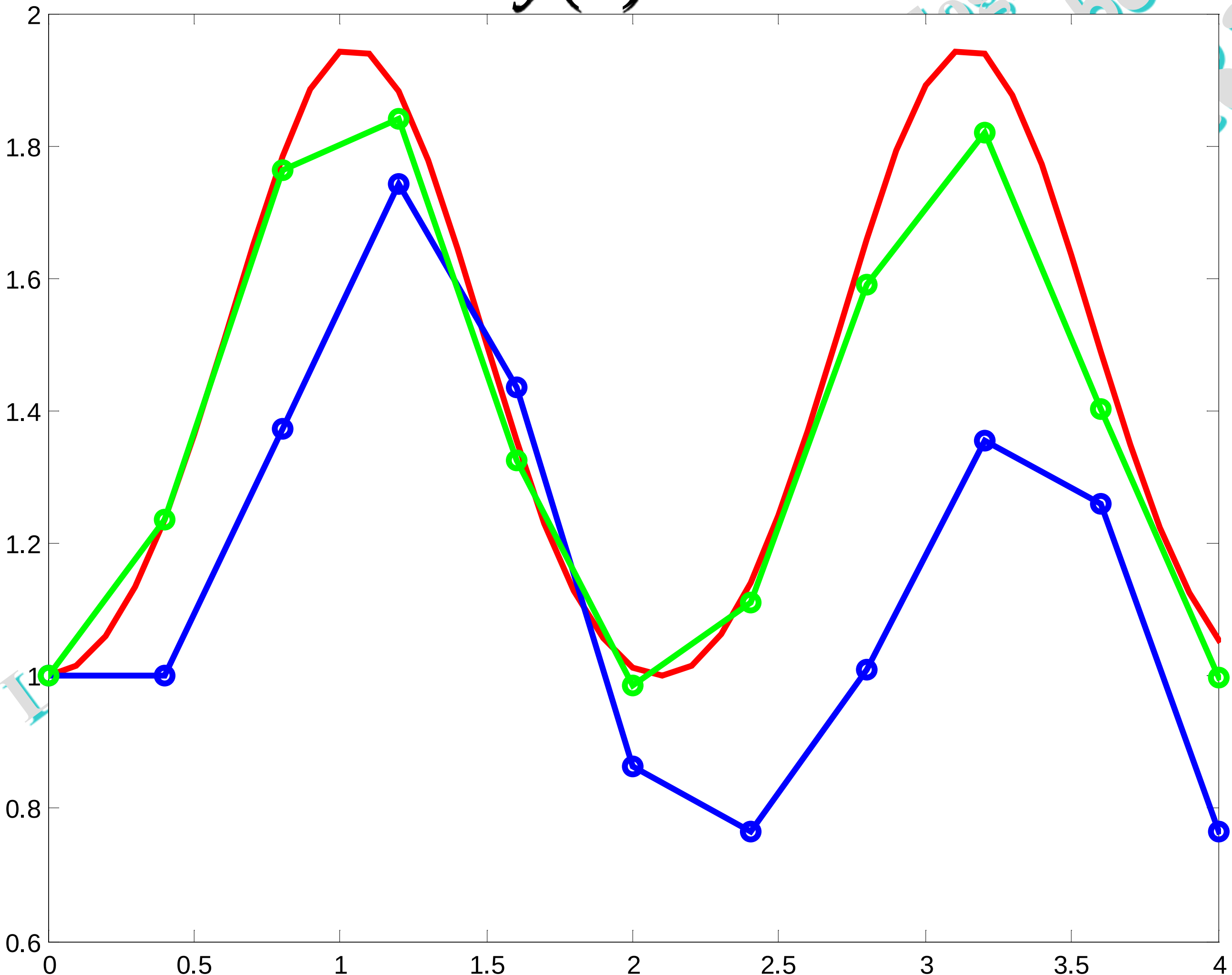
- dal metodo numerico;
- dal passo $h = x_{i+1} - x_i$;
- dall'ampiezza dell'intervallo $[a,b]$;
- dalla famiglia di soluzioni.

Errori ...

L'errore $|y_i - y(x_i)|$ dipende:

- dal metodo numerico;

Esempio: $$y' = y\sin 3x \quad x \in (0,4)$$ $$y(0) = 1$$



sol. vera

passo fisso h=0.4

Eulero

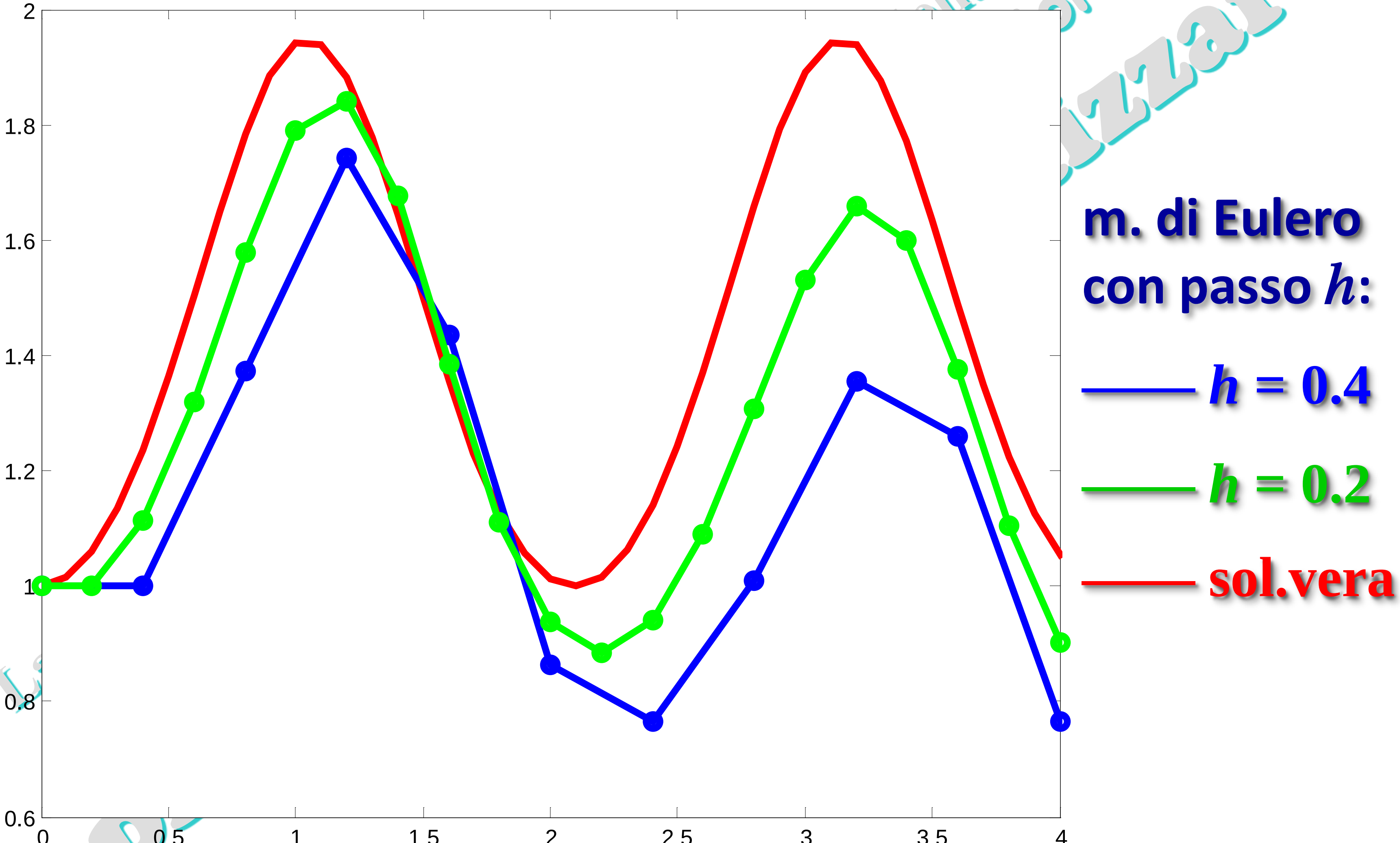
RK4

Runge-Kutta 4°ord.

Errori ...

L'errore $|y_i - y(x_i)|$ dipende:

- dal passo $h = x_{i+1} - x_i$;

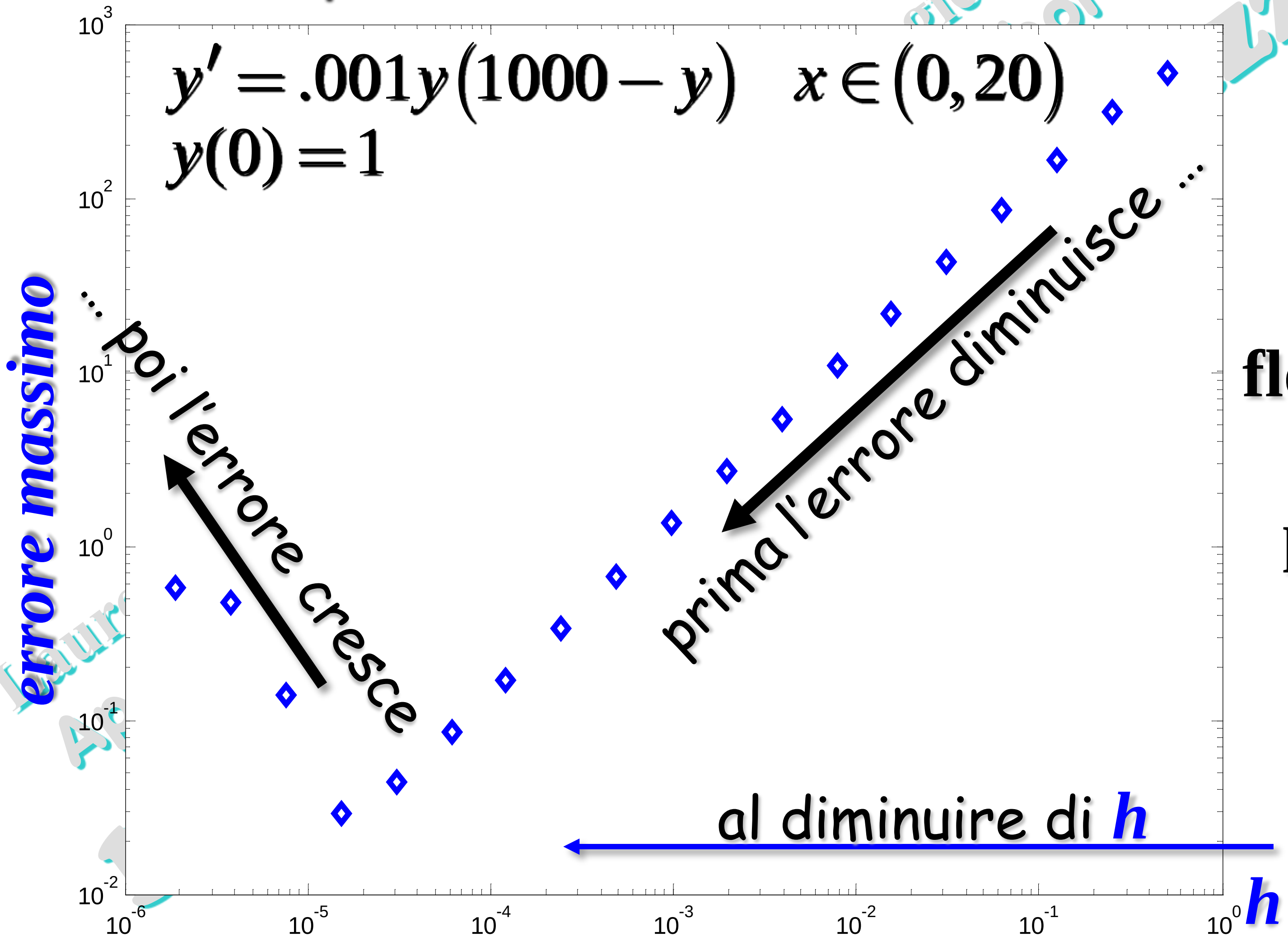


col m. di Eulero dimezzando il passo si dimezza l'errore

Nel **metodo di Eulero**, dimezzando il passo si dimezza l'errore

Cosa succede per *h* sempre più piccolo?

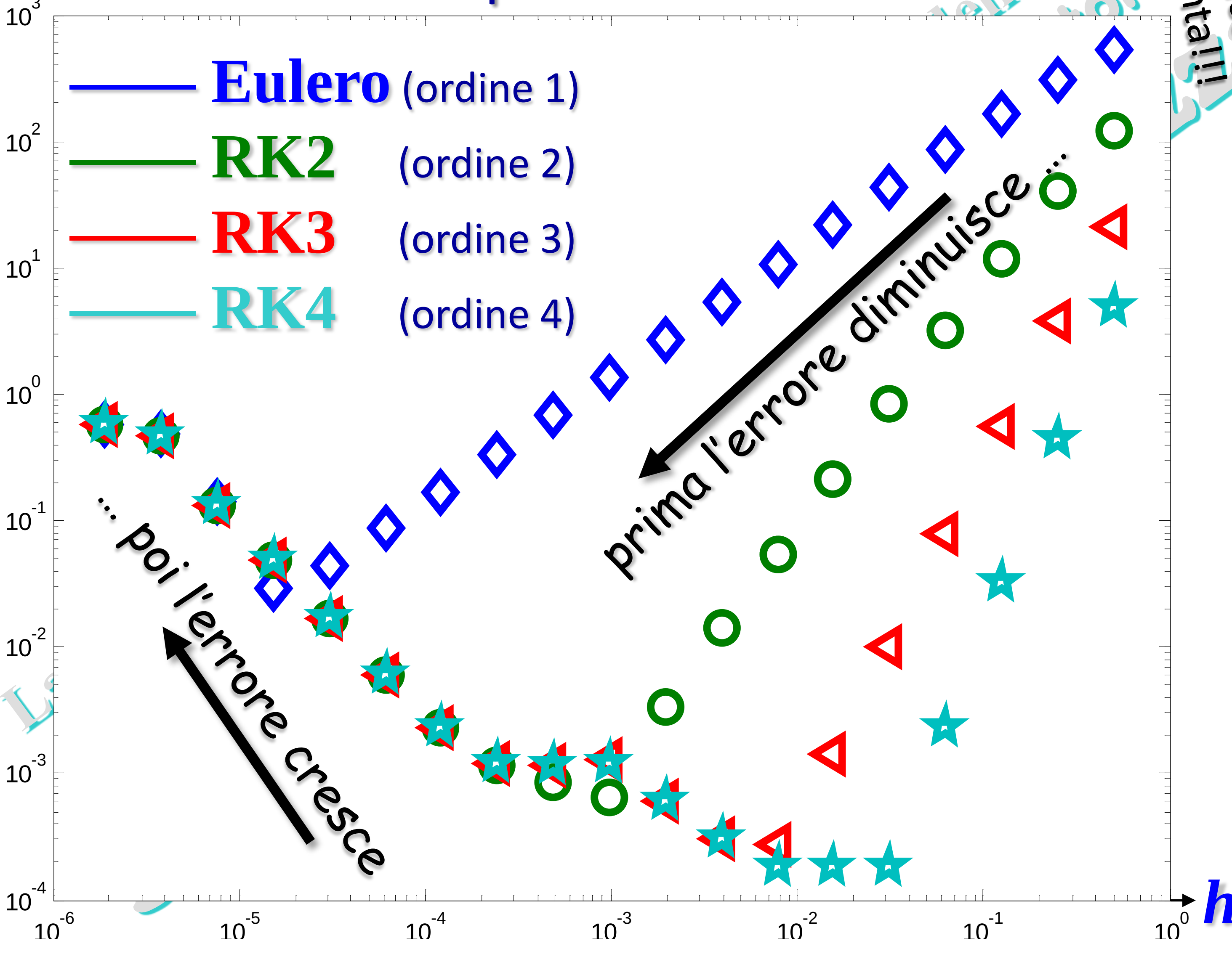
Esempio:



floating-p. in singola precisione

Il fenomeno non dipende dal metodo

errore massimo per $h=2^{-k}$:



L'errore globale è la somma dell'errore di troncamento (o di discretizzazione) e dell'errore di roundoff: per h sempre più piccolo il primo diminuisce, mentre il secondo aumenta!!!

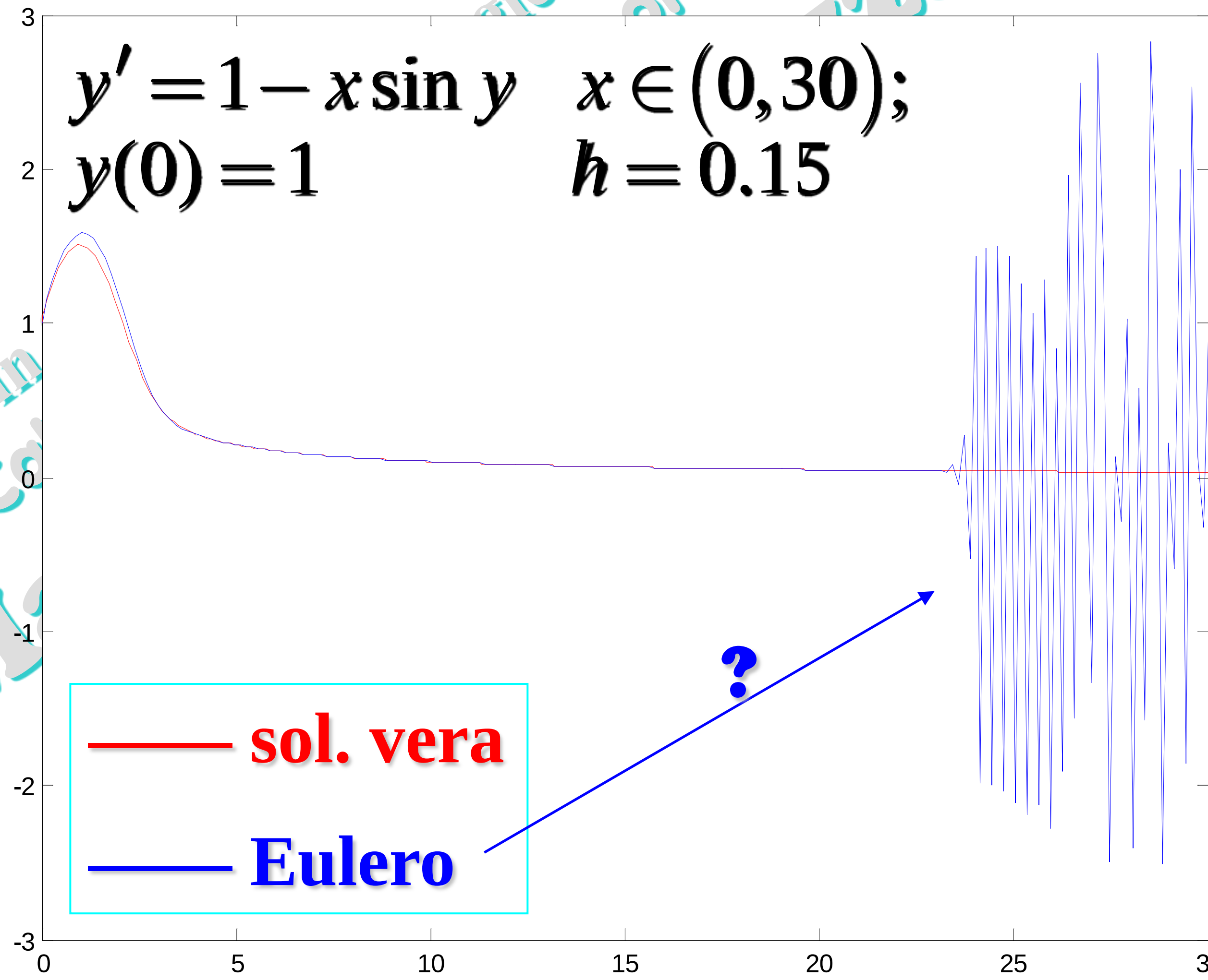
Errori ...

Eulero3

L'errore $|y_i - y(x_i)|$ dipende:

- dall'ampiezza dell'intervallo $[a,b]$;

Esempio:



Eulero6

Errori ... L'errore $|y_i - y(x_i)|$ dipende:

- dalla famiglia di soluzioni;

Esempio: $$y'' + 0.015y' - 0.5y + 0.5y^3 = 0$$

particolare Eq. di Duffing (o oscillatore di Duffing)

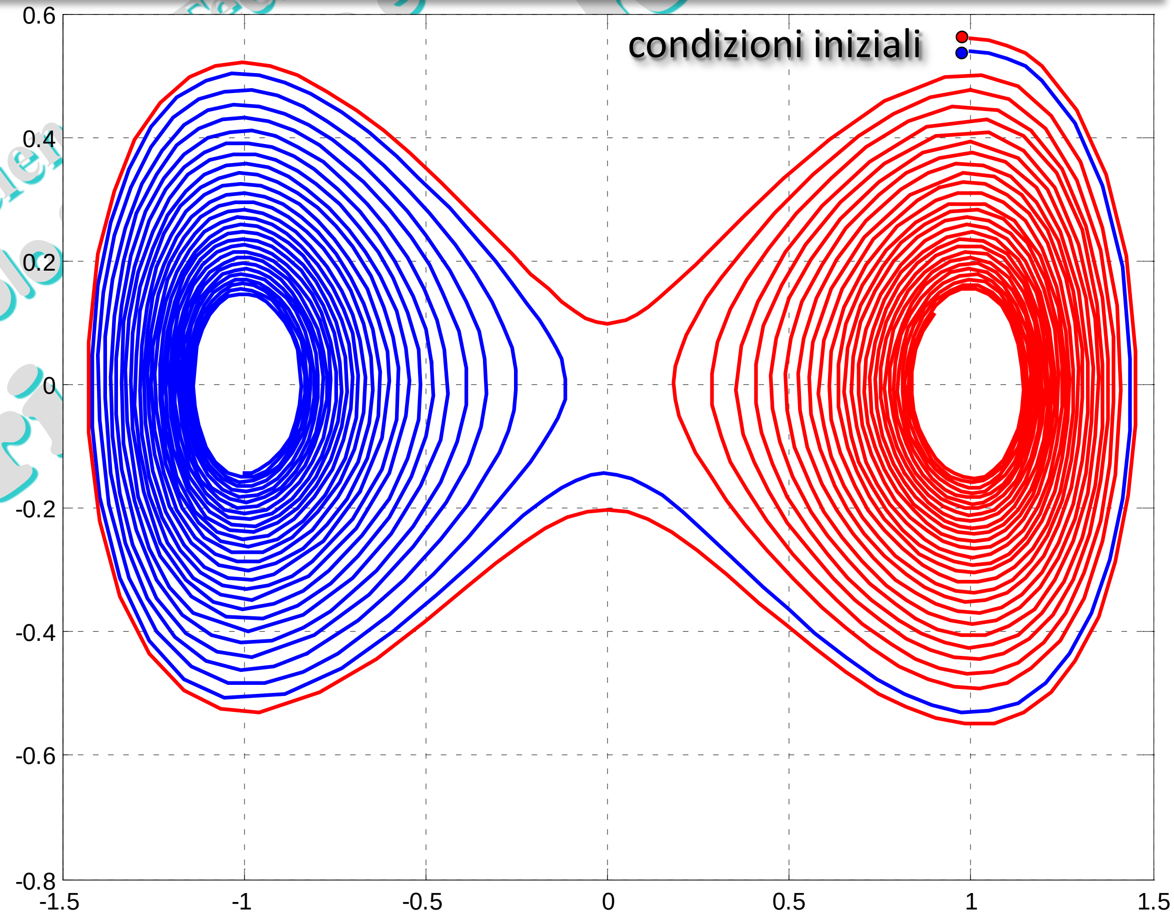
condizioni iniziali:

vicine

— $(1, 0.5\underline{4})$

— $(1, 0.5\underline{6})$

2 soluzioni completamente diverse!



Per quanto visto, **lo studio dell'errore nella soluzione numerica di un problema di Cauchy (IVP)** può essere ricondotto allo studio della **dipendenza** delle soluzioni della ODE **dalla condizione iniziale**.

Se risulta … $\left| y_0 - \widetilde{y}_0 \right| < \varepsilon$

si vuole che …

$$\left| y(x_i) - \widetilde{y}(x_i) \right| < \mathcal{O}(\varepsilon), \quad \forall i = 1, 2, \ldots, n$$

Quando succede ?

Esempio:

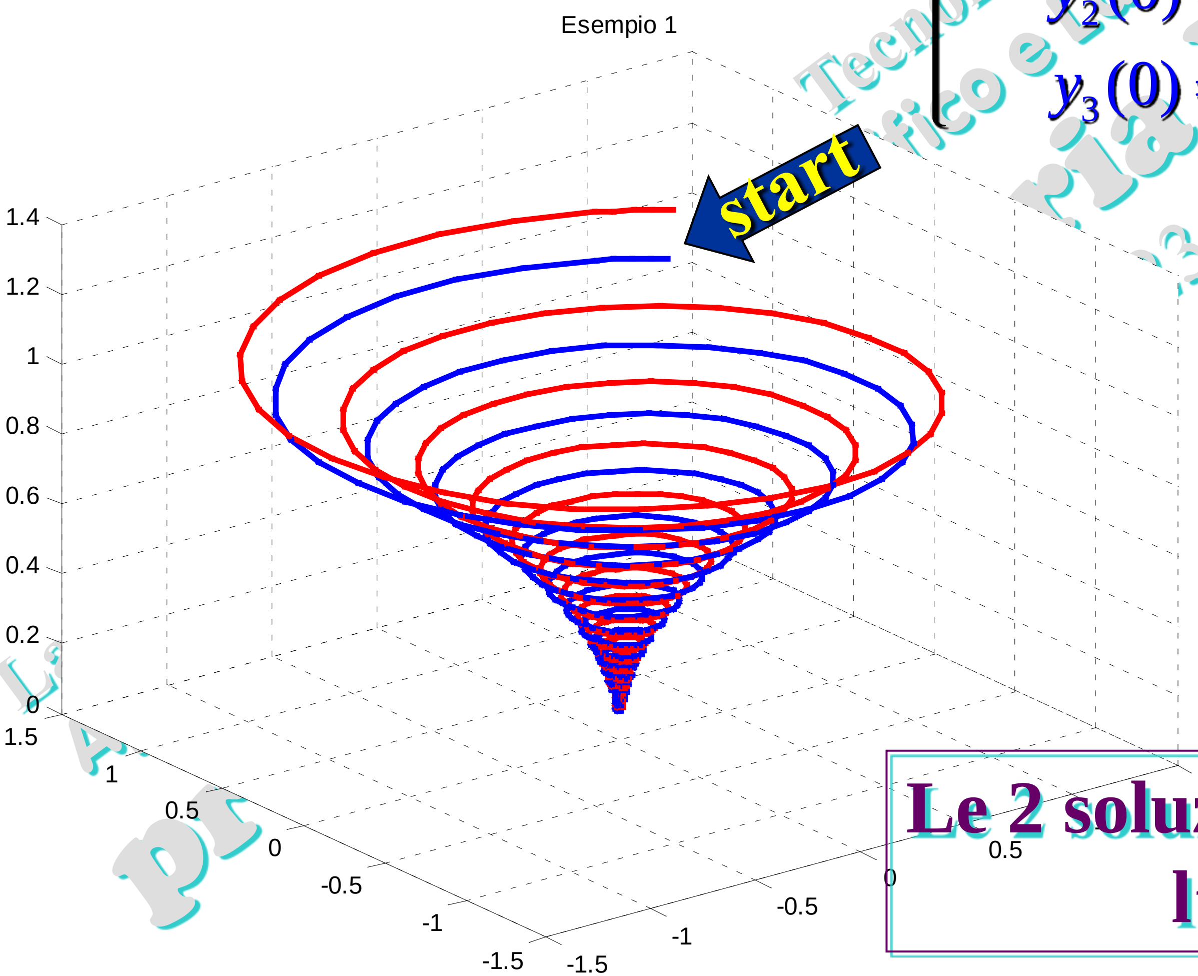
ode_cond

$$
\begin{cases}
y_1' = -y_1 - 20y_2 \\
y_2' = 20y_1 - y_2 \qquad x \in (0,5) \\
y_3' = -y_3/2 \\
y_1(0) = 1 \\
y_2(0) = 1 \\
y_3(0) = 1
\end{cases}
$$

condizioni iniziali perturbate

$$y_1(0) = 1 + \varepsilon$$
$$y_2(0) = 1 + \varepsilon$$
$$y_3(0) = 1 + \varepsilon$$

perturbazione
ε= 0.01



Le 2 soluzioni convergono: l'errore diminuisce

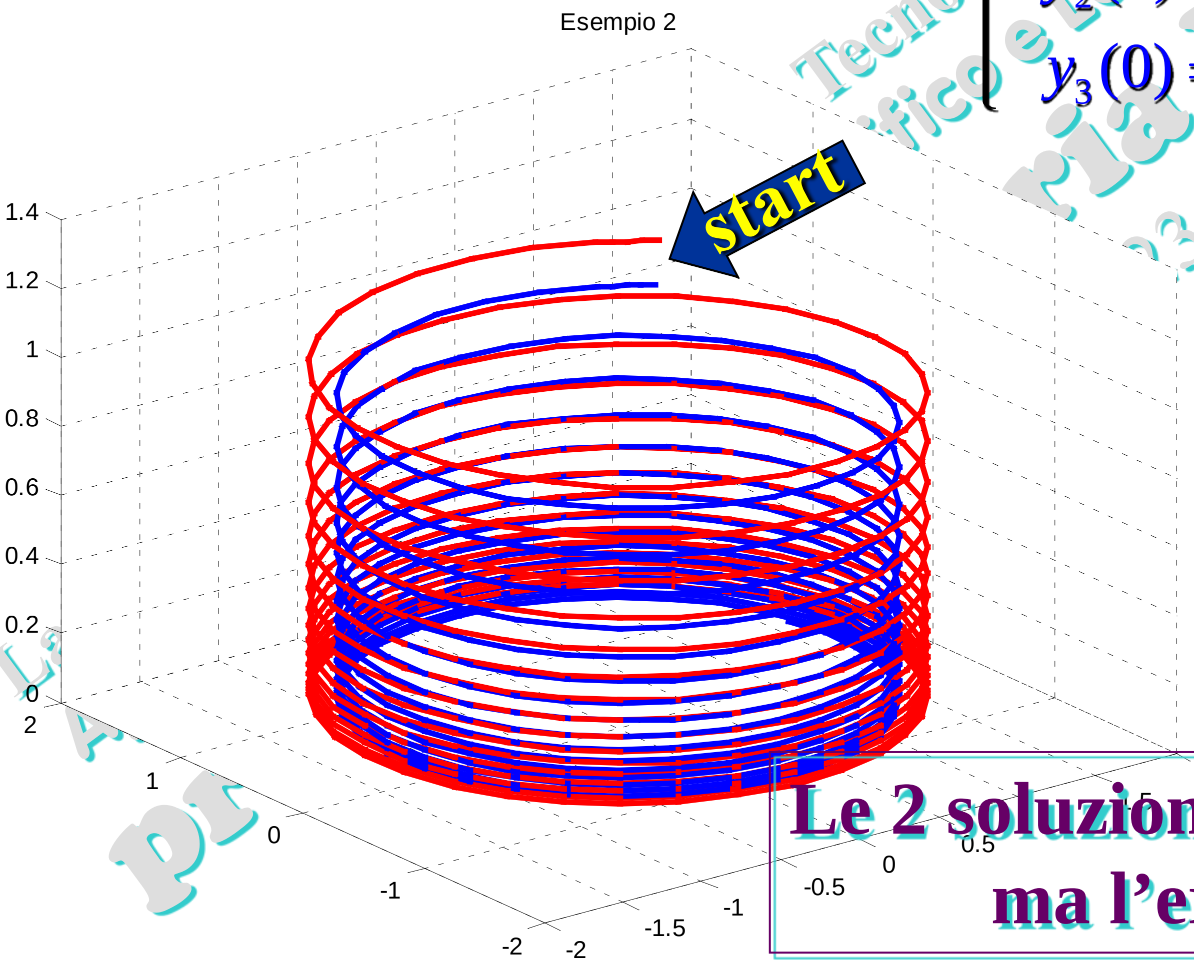
Esempio:

ode_cond

$$\begin{cases} y_1' = -20 y_2 \\ y_2' = 20 y_1 & x \in (0,5) \\ y_3' = -y_3/2 \\ y_1(0) = 1 & y_1(0) = 1+\varepsilon \\ y_2(0) = 1 & y_2(0) = 1+\varepsilon \\ y_3(0) = 1 & y_3(0) = 1+\varepsilon \end{cases}$$

condizioni iniziali perturbate

perturbazione
ε= 0.01



Le 2 soluzioni non convergono, ma l'errore non aumenta

Quando, perturbando il dato iniziale di un ***Problema di Cauchy***, si ha amplificazione della perturbazione nella soluzione corrispondente, si dice che il **problema differenziale è mal-condizionato** (si parla anche di stabilità dell'equazione differenziale nel senso di sensibilità della sua soluzione rispetto alla condizione iniziale)

problema

$$y' = \lambda y \qquad x \in (0,b)$$
$$y(0) = 1$$

$$\left[\text{sol. } y(x) = e^{\lambda x}\right]$$

problema perturbato

$$y' = \lambda y$$
$$y(0) = 1 + \varepsilon$$

$$\left[\text{sol. } y_\varepsilon(x) = (1+\varepsilon)e^{\lambda x}\right]$$

$$\max_{x\in[0,b]} \left|y_\varepsilon(x) - y(x)\right| = \begin{cases} |\varepsilon| \\ |\varepsilon| e^{\lambda b} \end{cases}$$

Problema

$\lambda \le 0$ **bencondizionato**

$\lambda > 0$ **malcondizionato**

malcond

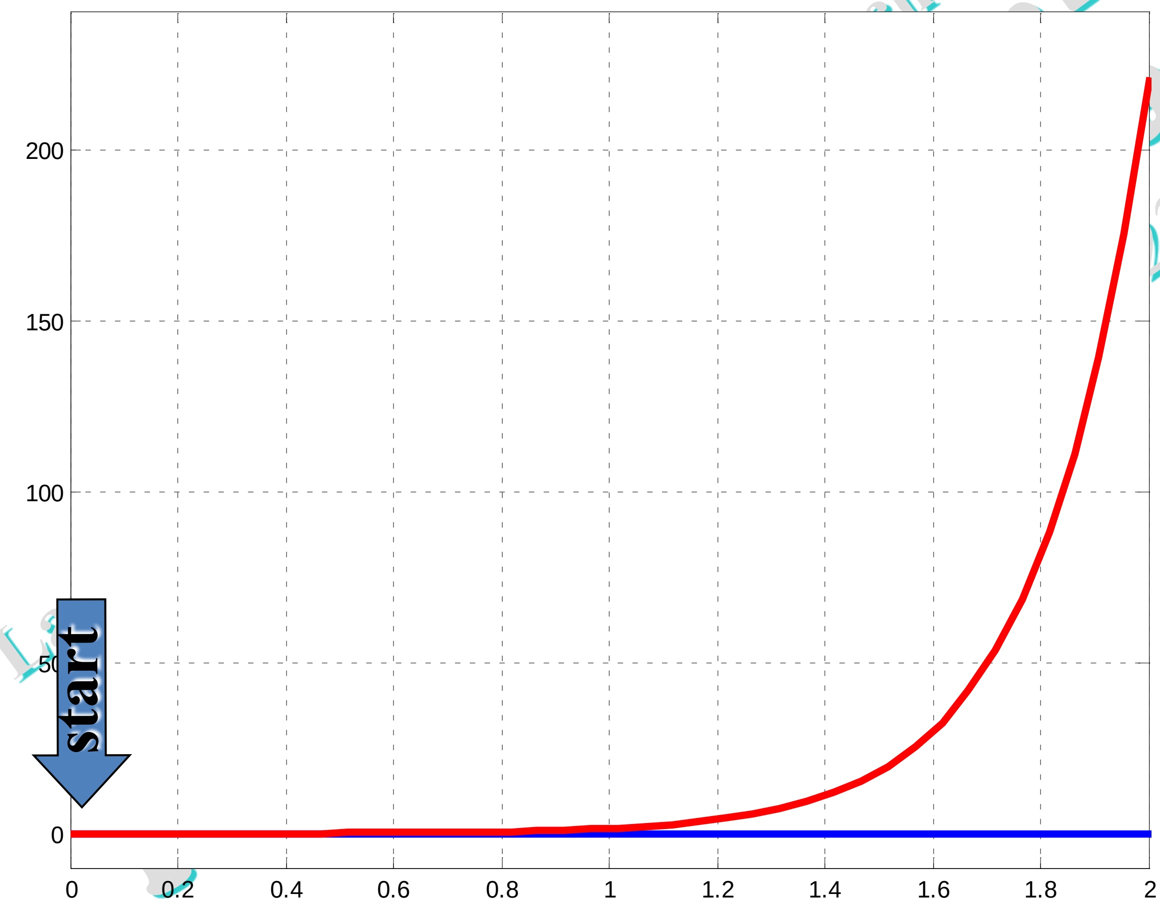
Esempio:

$$y' = 5y \qquad x \in (0,2)$$

$$y(0) = 0 \qquad y(0) = \varepsilon$$

$$\left[\text{sol. } y(x) \equiv 0\right] \qquad \left[\text{sol. } y(x) = \varepsilon e^{5t}\right]$$

perturbazione
ε= 0.01



Le 2 soluzioni divergono

L'esempio precedente è un caso particolare di …

$$y' = -y, \qquad x \in (0, b)$$
$$y(0) = y_0$$

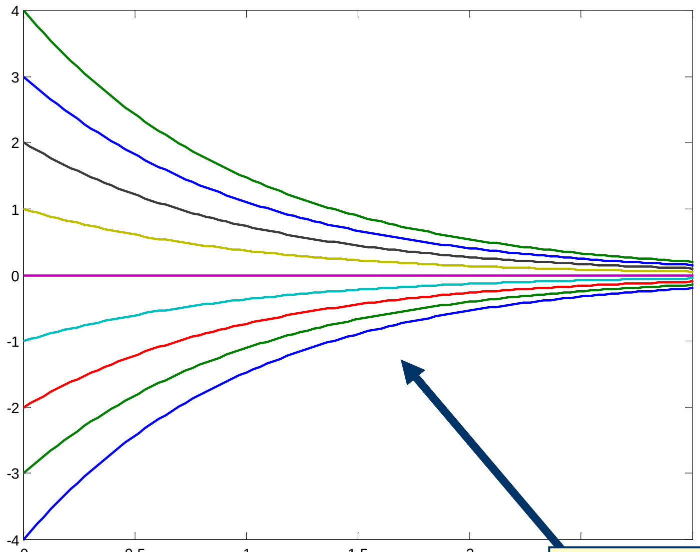
$$y' = +y, \qquad x \in (0, b)$$
$$y(0) = y_0$$

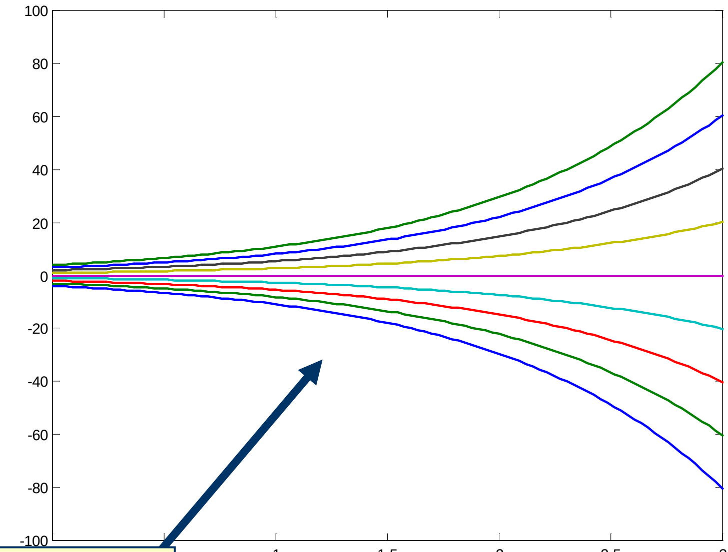
***Problema* benconditionato**

***Problema* malcondizionato**

soluzione analitica $y(x)=y_0\, e^{-x}$

soluzione analitica $y(x)=y_0\, e^{+x}$





Alcune soluzioni

Problema bencondizionato: cosa comporta?

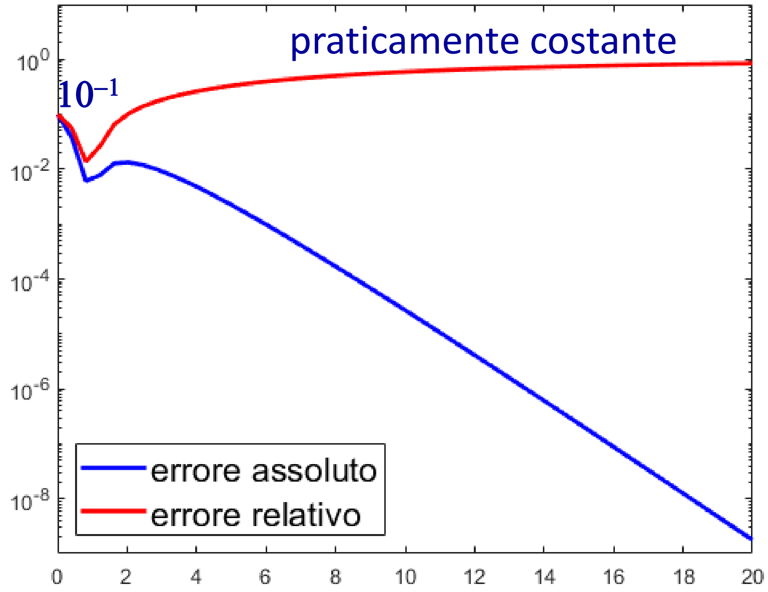
$$y' = -y, \qquad x \in (0, 20)$$
$$y(0) = y_0$$

Symbolic Math Toolbox

```
syms y(x);  Dy=diff(y,x); % y'
Y=dsolve(Dy == -y, y(0) == 1)
Y = exp(-x)
Yp=dsolve(Dy == -y*1.1, y(0) == 1*1.1)
Yp = (11*exp(-(11*x)/10))/10
xx=linspace(0,20,50)'; yy=subs(Y,x,xx); yyp=subs(Yp,x,xx);
semilogy(xx,abs(yy-yyp),'b',xx,abs(yy-yyp)./yy,'r')
legend('errore assoluto','errore relativo')
```

perturbazione = 10%

Err. assoluto **Err. relativo**



L'errore tra le **2 soluzioni analitiche** non si amplifica rispetto alla perturbazione introdotta

Problema malcondizionato: cosa comporta?

$$y' = +y, \qquad x \in (0, 20)$$
$$y(0) = y_0$$

Symbolic Math Toolbox

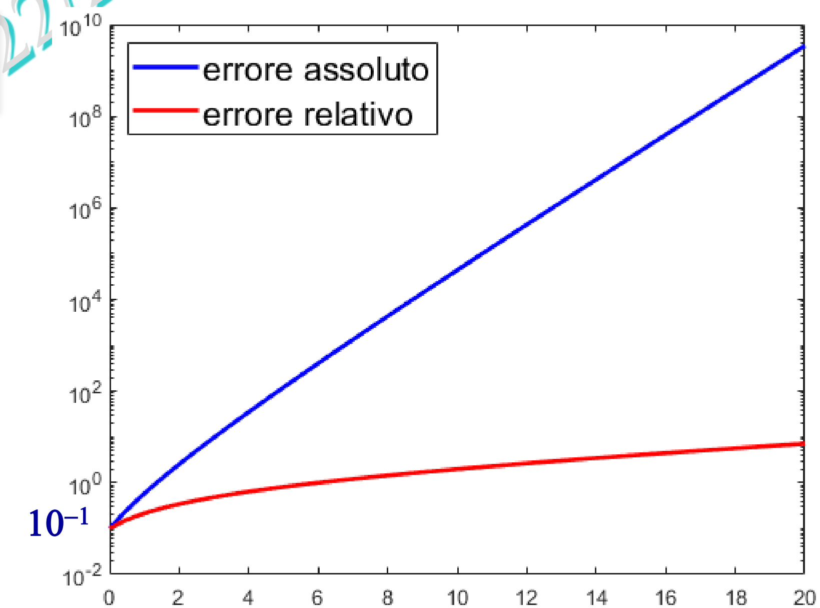
```
syms y(x);  Dy=diff(y,x); % y'
Y=dsolve(Dy == +y, y(0) == 1)
Y = exp(x)
Yp=dsolve(Dy == +y*1.1, y(0) == 1*1.1)
Yp = (11*exp((11*x)/10))/10
xx=linspace(0,20,50)'; yy=subs(Y,x,xx); yyp=subs(Yp,x,xx);
semilogy(xx,abs(yy-yyp),'b',xx,abs(yy-yyp)./yy,'r')
legend('errore assoluto','errore relativo')
```

perturbazione = 10%

Err. assoluto

Err. relativo

L'errore tra le **2 soluzioni analitiche** si amplifica rispetto alla perturbazione introdotta



Poiché la risoluzione numerica di un problema introduce sempre delle perturbazioni, allora <u>non si possono risolvere accuratamente</u> i problemi malcondizionati

Il problema generale di Cauchy

$$\begin{cases} y' = f(x,y) & x \in (a,b) \\ y(a) = y_0 \end{cases}$$

per definizione, si dice

benconidizionato $\Longleftrightarrow$ $\dfrac{\partial f}{\partial y}(x,y) \le 0$

malcondizionato $\Longleftrightarrow$ $\dfrac{\partial f}{\partial y}(x,y) > 0$

Nel caso di un sistema differenziale il ruolo di $\dfrac{\partial f}{\partial y}(x,y)$ è assunto dalla parte reale degli autovalori della matrice Jacobiana J_f della f

Esempi di condizionamento: 1 eq. diff.

$$y' = 5y$$

$$\frac{\partial f}{\partial y}(x,y) = 5 \quad \Longleftrightarrow \quad \textbf{malcondizionato}$$

$$y' = -5y$$

$$\frac{\partial f}{\partial y}(x,y) = -5 \quad \Longleftrightarrow \quad \textbf{bencondizionato}$$

$$y' = -10y(x-1)$$

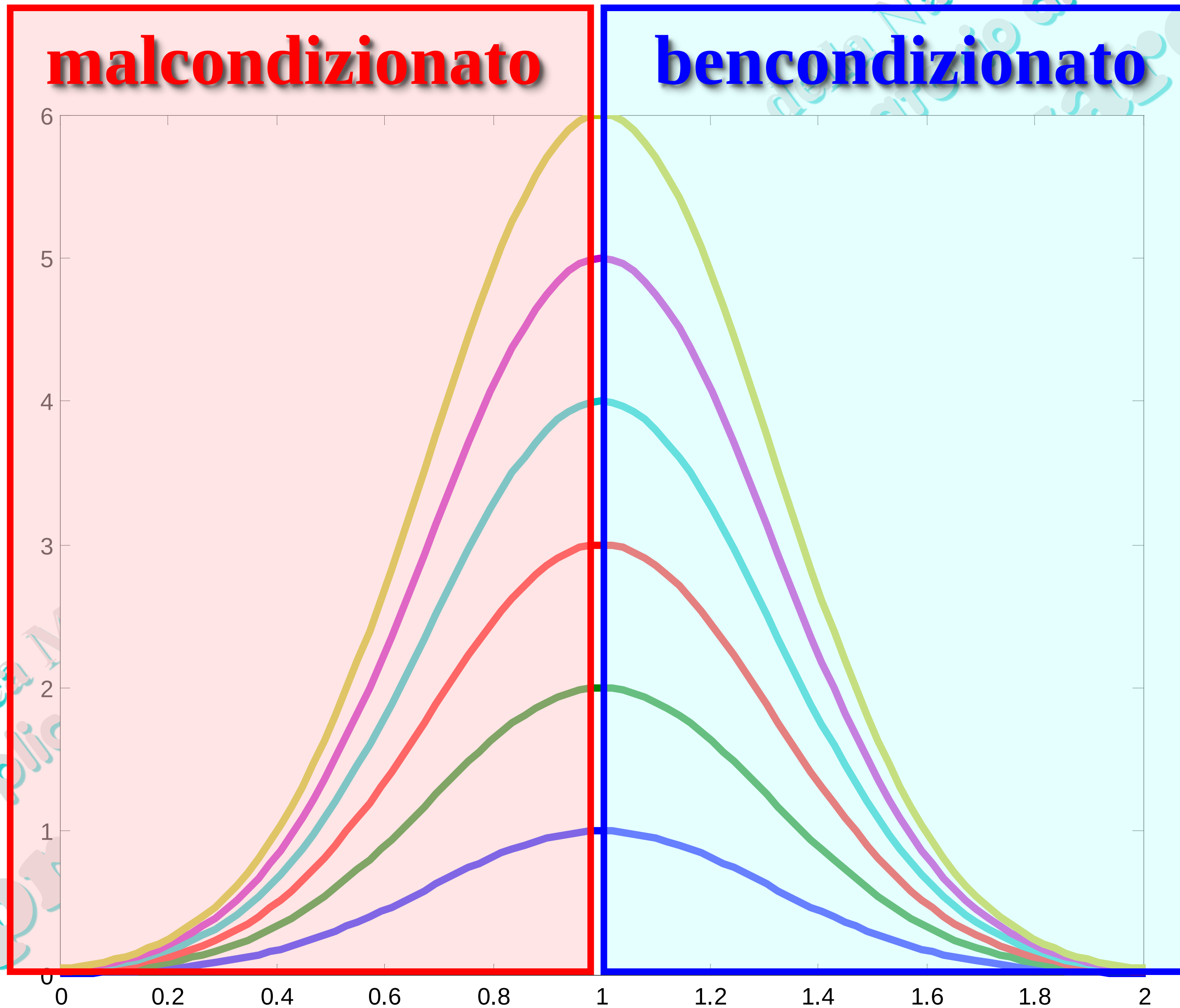
$$\frac{\partial f}{\partial y}(x,y) = -10(x-1) \quad \Longleftrightarrow$$

$x \geq 1$ **bencondizionato**

$x < 1$ **malcondizionato**

Alcune curve della famiglia di soluzioni di $y' = -10y(x-1)$

soluzione analitica $y = C_1 e^{-5x(x-2)}$



Esempi di condizionamento: sistema diff.

$$\begin{cases} y_1' = -y_1 - 20y_2 \\ y_2' = 20y_1 - y_2 \qquad x \in (0,5) \\ y_3' = -y_3/2 \\ y_1(0) = 1 \\ y_2(0) = 1 \\ y_3(0) = 1 \end{cases}$$

$$\underline{y}' = \underline{f}(x, y_1, y_2, y_3)$$

matrice jacobiana di f

$$J_f(x, y_1, y_2, y_3) = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \frac{\partial f_1}{\partial y_3} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \frac{\partial f_2}{\partial y_3} \\ \frac{\partial f_3}{\partial y_1} & \frac{\partial f_3}{\partial y_2} & \frac{\partial f_3}{\partial y_2} \end{pmatrix} = \begin{pmatrix} -1 & -20 & 0 \\ 20 & -1 & 0 \\ 0 & 0 & -.5 \end{pmatrix}$$

autovalori: $\boxed{-0.5, \quad -1} \pm i\,20$

parte reale < 0 **bencondizionato**

Esempi di condizionamento: sistema diff.

$$\begin{cases} y_1' = y_1 - 20y_2 \\ y_2' = 20y_1 \qquad\qquad x \in (0,5) \\ y_3' = -y_3/2 \\ y_1(0) = 1 \\ y_2(0) = 1 \\ y_3(0) = 1 \end{cases}$$

$$\underline{y}' = \underline{f}(x, y_1, y_2, y_3)$$

matrice jacobiana di f

$$J_f(x, y_1, y_2, y_3) = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \frac{\partial f_1}{\partial y_3} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \frac{\partial f_2}{\partial y_3} \\ \frac{\partial f_3}{\partial y_1} & \frac{\partial f_3}{\partial y_2} & \frac{\partial f_3}{\partial y_2} \end{pmatrix} = \begin{pmatrix} 1 & -20 & 0 \\ 20 & 0 & 0 \\ 0 & 0 & -.5 \end{pmatrix}$$

autovalori: -0.5, $\boxed{0.5} \pm i\,19.99$

parte reale > 0 **malcondizionato**

A che serve sapere se il problema sia **bencondizionato** oppure **malcondizionato**?

Un Problema di Cauchy bencondizionato non assicura un'accurata risoluzione numerica!

* **dipende dalla stabilità del metodo numerico**

ma

Un Problema di Cauchy malcondizionato comporta una soluzione numerica inaccurata!

Esempio

$$y' = -10y(x-1)$$

soluzione analitica $y = C_1 e^{-5x(x-2)}$

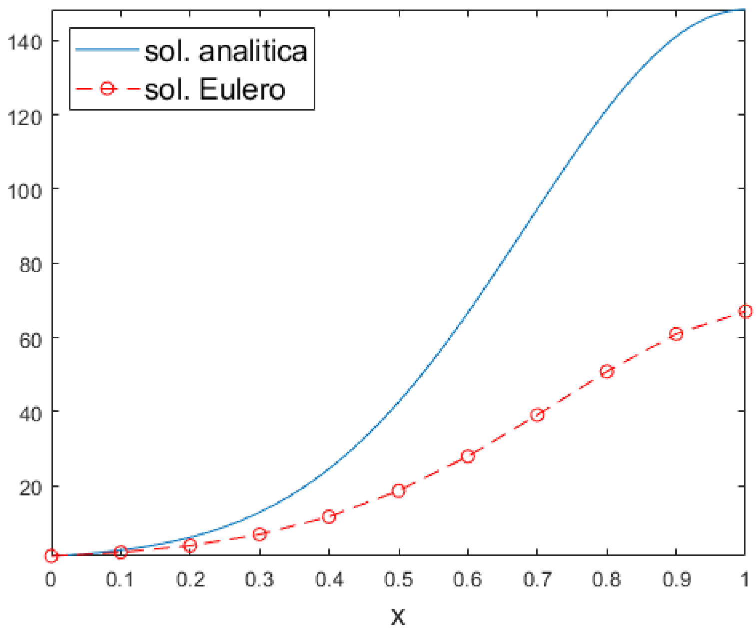
$$\begin{cases} y' = -10y(x-1), & x \in]0,1[\\ y(0) = 1 \end{cases}$$

soluzione analitica $y = e^{-5x(x-2)}$

$$\begin{cases} y' = -10y(x-1), & x \in]1,2[\\ y(1) = 6 \end{cases}$$

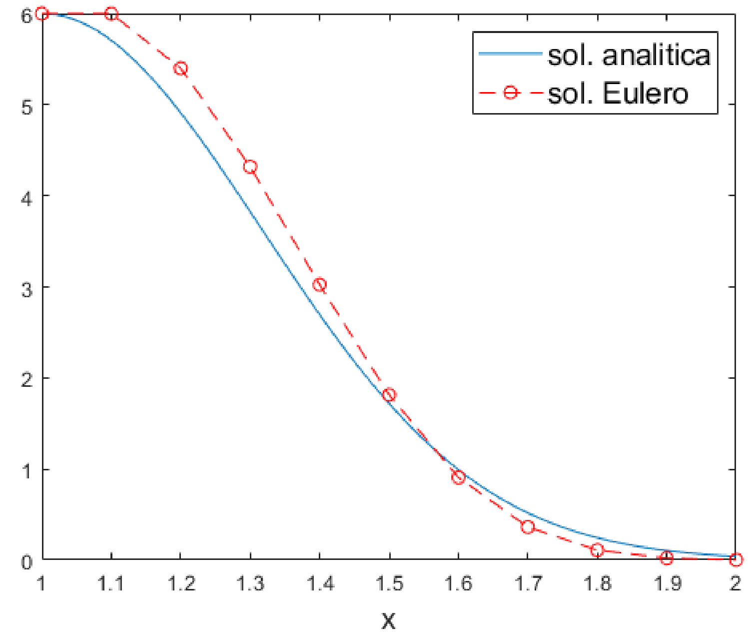
soluzione analitica $y = 6e^{-5(x-1)^2}$

regione di mal condizionamento



L'accuratezza della soluzione numerica degrada

regione di buon condizionamento



L'accuratezza della soluzione numerica non degrada

Analisi dell'errore per il metodo di Eulero

Errore locale di troncamento $\quad e_{k+1}^{\mathrm{T}} = y(x_{k+1}) - y_{k+1}, \quad \boxed{y(x_k) = y_k}$

È l'errore introdotto in un solo passo di applicazione del metodo numerico; si ottiene supponendo che al passo $k+1$ siano esatti i valori calcolati nei passi precedenti.

Errore globale di troncamento $\quad E_{k+1}^{\mathrm{T}} = y(x_{k+1}) - y_{k+1}, \quad \boxed{y(x_k) \neq y_k}$

È l'errore commesso nell'approssimare ciascun valore esatto $y(x_k)$ mediante y_k. A tale errore contribuiscono sia gli errori locali nei vari punti precedenti, sia la loro propagazione.

Errore locale di troncamento per il *metodo di Eulero*

$$e_{k+1}^{\mathrm{T}} = y(x_{k+1}) - y_{k+1} = y(x_{k+1}) - \left[y_k + hf(x_k, y_k)\right] =$$

$$= y(x_{k+1}) - \left[y(x_k) + hf(x_k, y(x_k))\right]$$

$\Leftarrow \quad \boxed{y(x_k) = y_k}$

Dallo sviluppo di Taylor di $y(x_{k+1})$:

$$y(x_{k+1}) = y(x_k) + hf(x_k, y(x_k)) + \frac{h^2}{2} y''(\xi)$$

con $x_k < \xi < x_{k+1}$

$$\Rightarrow \quad \boxed{e_{k+1}^{\mathrm{T}} = \frac{h^2}{2} y''(\xi) \simeq O(h^2)}$$

Errore locale di troncamento del *metodo di Eulero*

Analisi dell'errore per il metodo di Eulero

Taylor $\quad y(x_{k+1}) = y(x_k) + hf(x_k, y(x_k)) + \frac{h^2}{2} y''(\xi)$

Eulero $\quad y_{k+1} = y_k + hf(x_k, y_k)$

Per l'Errore globale di troncamento del metodo di Eulero

$$\boxed{E_{k+1}^{\mathrm{T}} = y(x_{k+1}) - y_{k+1}} = \boxed{y(x_k) - y_k} + h\left[f(x_k, y(x_k)) - f(x_k, y_k)\right] + \frac{h^2}{2} y''(\xi)$$

cioè

$$\boxed{E_{k+1}^{\mathrm{T}} = E_k^{\mathrm{T}} + h\left[f(x_k, y(x_k)) - f(x_k, y_k)\right] + e_{k+1}^{\mathrm{T}}}$$

Errore globale di troncamento del *m. di Eulero*

Per il *Teorema del valor medio di Lagrange*, si ha:

$$f(x_k, y(x_k)) - f(x_k, y_k) = \boxed{\frac{\partial f}{\partial y}}(x_k, \eta_k)\boxed{y(x_k) - y_k} \cong L E_k^{\mathrm{T}}$$

$$\boxed{L < 0}$$

problema bencondizionato

$$\boxed{E_{k+1}^{\mathrm{T}} = (1 + hL) E_k^{\mathrm{T}} + e_{k+1}^{\mathrm{T}}}$$

Relazione tra errore globale ed errore locale nel *metodo di Eulero*

Il fattore $(1+hL)$ rappresenta l'amplificazione dell'errore globale al passo precedente.

$(1+hL)E_k^{\mathrm{T}}$ rappresenta l'accumulo degli errori precedenti (*errore di propagazione*).

Perché Eulero è un metodo del 1° ordine

$$e_{k+1}^{\mathrm{T}} = \frac{h^2}{2} y''(\xi) \simeq O\left(h^2\right)$$

$$E_{k+1}^{\mathrm{T}} = (1+hL) E_k^{\mathrm{T}} + e_{k+1}^{\mathrm{T}}$$

errore di propagazione — errore locale di troncamento

$$E_1^{\mathrm{T}} = e_1^{\mathrm{T}}$$
$$E_2^{\mathrm{T}} = e_2^{\mathrm{T}} + E_1^{\mathrm{T}}(1+hL) = e_2^{\mathrm{T}} + e_1^{\mathrm{T}}(1+hL)$$
$$E_3^{\mathrm{T}} = e_3^{\mathrm{T}} + E_2^{\mathrm{T}}(1+hL) = e_3^{\mathrm{T}} + e_2^{\mathrm{T}}(1+hL) + e_1^{\mathrm{T}}(1+hL)^2$$
$$\vdots$$
$$E_{n+1}^{\mathrm{T}} = e_{n+1}^{\mathrm{T}} + e_n^{\mathrm{T}}(1+hL) + e_{n-1}^{\mathrm{T}}(1+hL)^2 + \cdots + e_1^{\mathrm{T}}(1+hL)^n$$

$$E_{n+1}^{\mathrm{T}} = O\left(h^2\right)\Big\{\underbrace{1+(1+hL)+(1+hL)^2+\cdots+(1+hL)^n}_{\text{Ridotta della Serie Geometrica}}\Big\}$$

Ridotta della Serie Geometrica $\quad 1+x+x^2+x^3+\cdots+x^n+\cdots = \frac{1}{1-x}, \quad |x|<1$

Essendo $\boxed{L<0}$ e h piccolo, allora $|1+hL|<1$ e la Serie Geometrica converge a: $\frac{1}{1-(1+hL)}$

$$E_{n+1}^{\mathrm{T}} \approx O\left(h^2\right)\left|\frac{1}{hL}\right| = -\frac{1}{L} O(h) \approx O(h)$$

Eulero è un metodo del 1° ordine

Convergenza e stabilità del metodo di Eulero

Per $h \to 0$, il metodo di Eulero converge come:

$$E_{n+1}^{\mathrm{T}} = \underbrace{e_1^{\mathrm{T}} + e_2^{\mathrm{T}} + \cdots + e_n^{\mathrm{T}}}_{n} \simeq n\,O\left(h^2\right) = \frac{1}{h}O\left(h^2\right) = O\left(h\right) \qquad h = \frac{b-a}{n}$$

$$\boxed{E_{k+1}^{\mathrm{T}} = \left(1 + hL\right) E_k^{\mathrm{T}} + e_{k+1}^{\mathrm{T}}} \Longrightarrow \boxed{\left|E_{k+1}^{\mathrm{T}}\right| \le \left|1 + hL\right| \left|E_k^{\mathrm{T}}\right| + \left|e_{k+1}^{\mathrm{T}}\right|}$$

Il fattore $|1+hL|$ rappresenta l'amplificazione dell'errore globale al passo precedente:

- Se il problema è mal condizionato ($L > 0$), allora è sempre $|1+hL|>1$ e l'errore si amplifica ad ogni passo.
- Se il problema è ben condizionato ($L \le 0$), allora la disuguaglianza $|1+hL|\le 1$ è verificata per particolari valori di h (il m. di Eulero è *condizionatamente stabile*):

$$|1+hL|\le 1 \iff -1 \le 1-hL \le 1 \quad \text{sempre verificata}$$

$$\iff -2 \le -hL \le 0$$

$$\iff \boxed{0 \le h \le -2/L} \longrightarrow \textit{m. di Eulero} \text{ stabile, altrimenti instabile}$$

limitazione superiore su h

Esempio: stabilità del metodo di Eulero e di BE

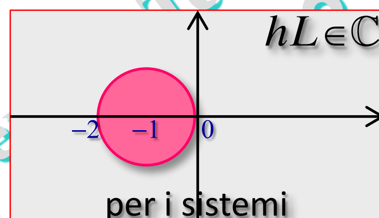
problema: $\begin{cases} y' = -30y \\ y(0) = 1 \end{cases}$ $\quad x \in]0, 1[$

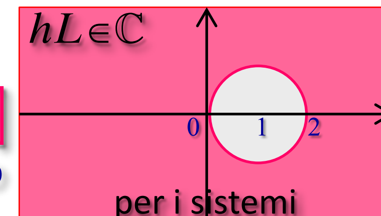
$$\frac{\partial f}{\partial y} = \boxed{-30 = L} \le 0$$

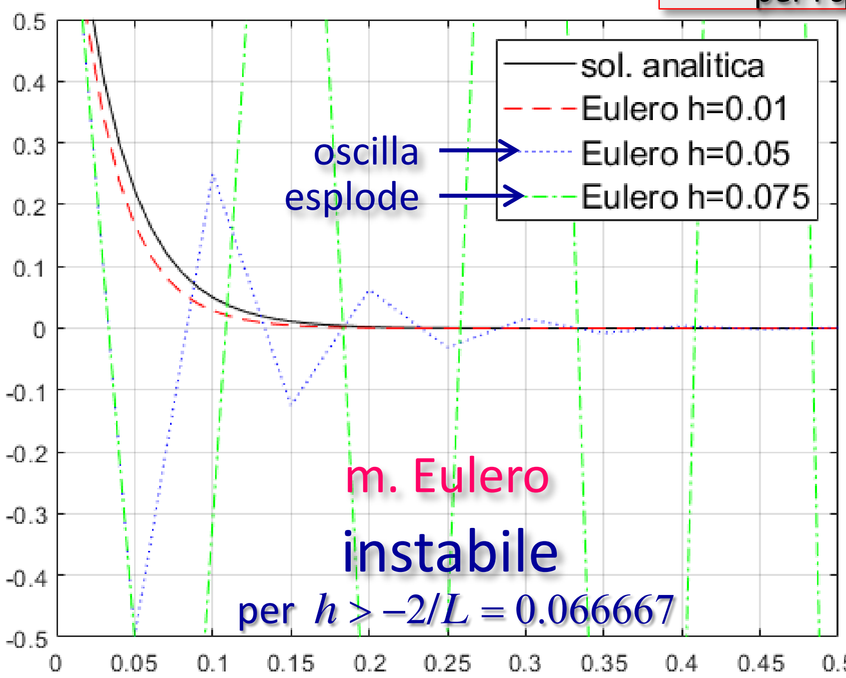
problema ben condizionato

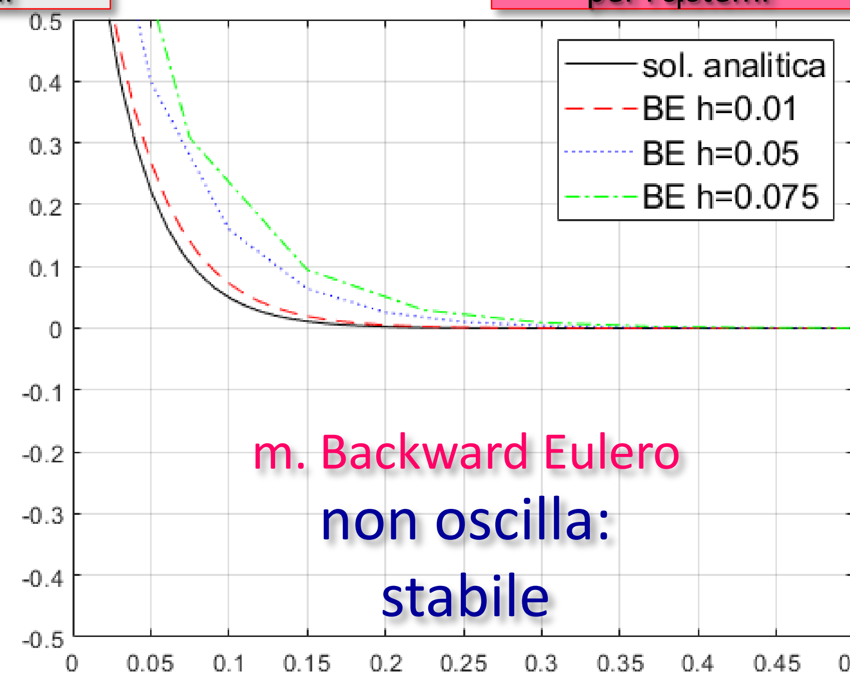
soluzione: $y(x) = e^{-30x}$

	m. Eulero	m. Backward Eulero
fattore di amplificazione	$\lvert 1 + hL \rvert \le 1$	$\dfrac{1}{1-hL} < 1$
regione di assoluta stabilità hL	$hL \in [-2, 0]$	$hL \in]-\infty, 0]$ sempre verificato



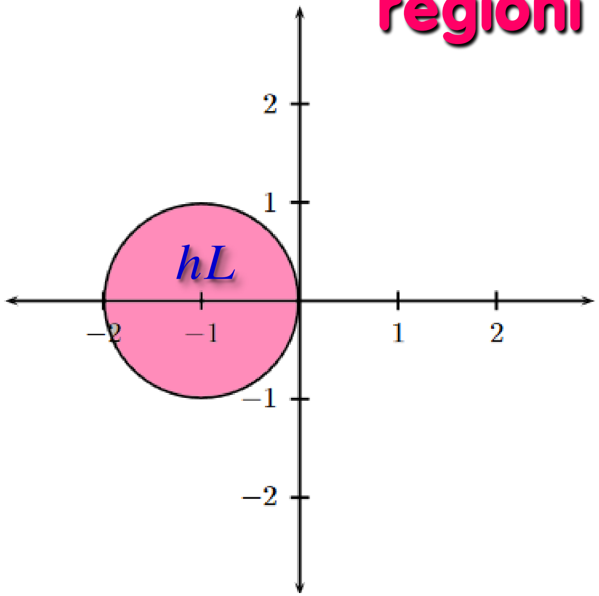




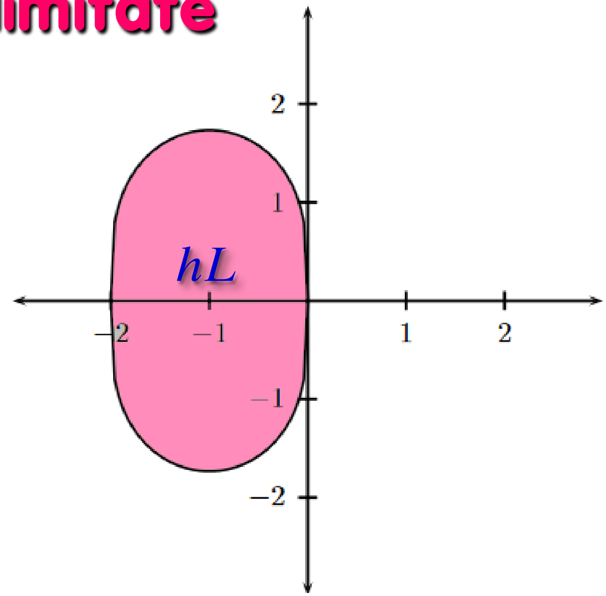


Regioni di stabilità di alcuni metodi espliciti

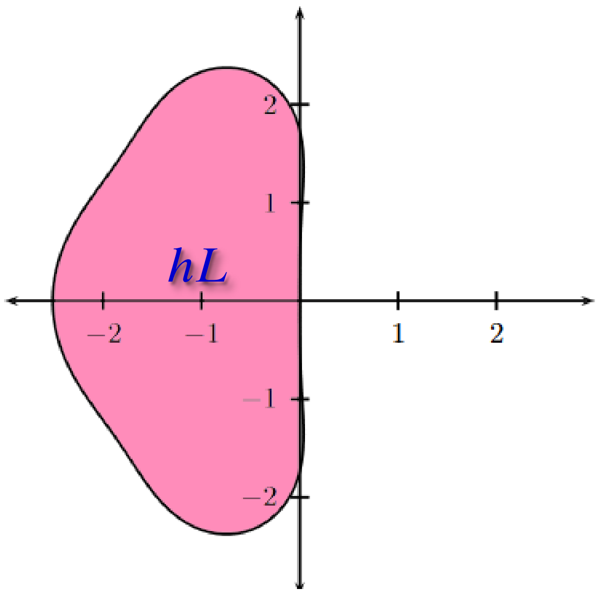
regioni limitate



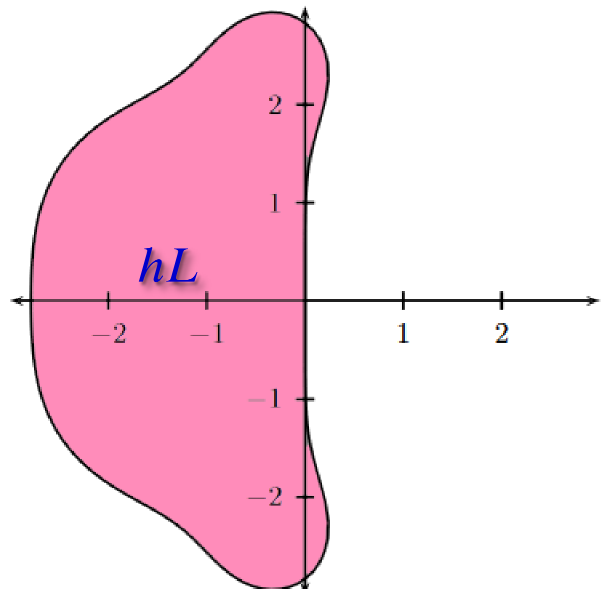
metodo di Eulero (RK 1° ordine)



metodo RK 2° ordine



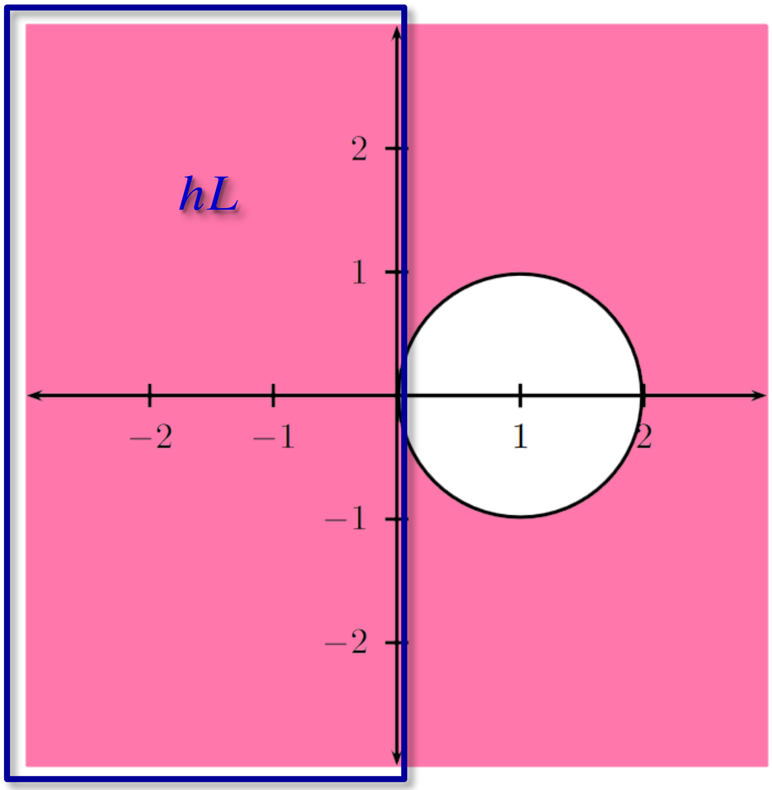
metodo RK 3° ordine



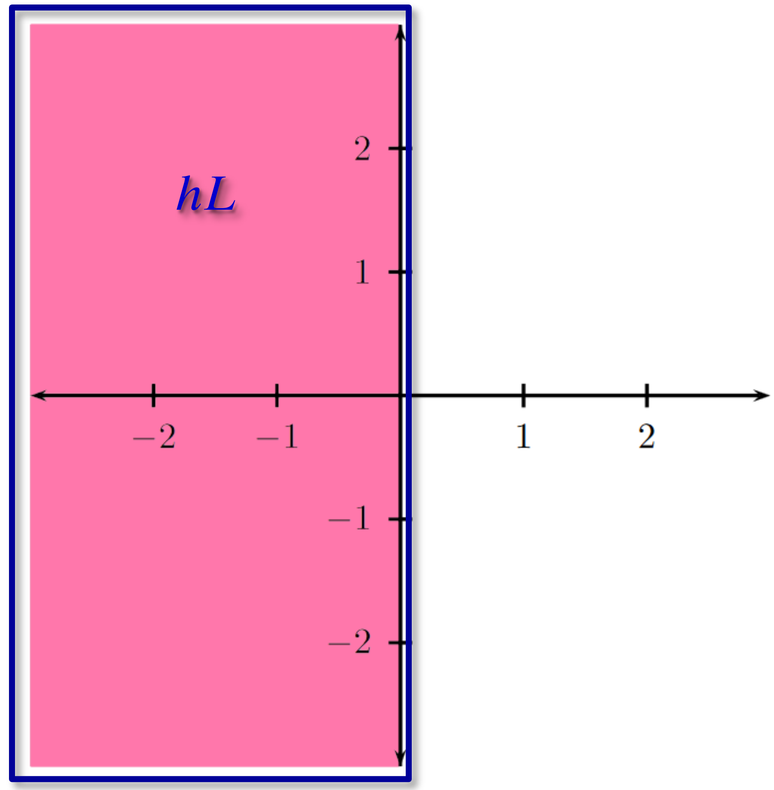
metodo RK 4° ordine

Regioni di stabilità di alcuni metodi impliciti

regioni illimitate



metodo Backward Eulero



metodo Trapezoidale

Equazioni Differenziali Ordinarie (ODE)

MATLAB offre molte funzioni (ODE solver) per risolvere problemi di Equazioni Differenziali Ordinarie (Ordinary Differential Eqs - ODE):

Non-stiff Solvers	
ode45	Solve nonstiff differential equations — medium order method.
ode23	Solve nonstiff differential equations — low order method.
ode78	Solve nonstiff differential equations — high order method.
ode89	Solve nonstiff differential equations — high order method.
ode113	Solve nonstiff differential equations — variable order method.
Stiff Solvers	
ode15s	Solve stiff differential equations and DAEs (differential-algebraic eqs) — variable order method
ode23s	Solve stiff differential equations — low order method
ode23t	Solve moderately stiff ODEs and DAEs — trapezoidal rule
ode23tb	Solve stiff differential equations — trapezoidal rule + backward differentiation formula
Fully Implicit Solvers	
ode15i	Solve fully implicit differential equations — variable order method
decic	Compute consistent initial conditions for ode15i

in *MATLAB*…

```
[x,y]=ode23 (fun, [a b], y0);
[x,y]=ode45 (fun, [a b], y0); . . .
```

ode23 e **ode45** sono basati su una coppia di *metodi espliciti Runge-Kutta Fehlberg* di ordini 2, 3 e 4, 5 rispettivamente. Per gli ODE Solver i due numeri rappresentano l'ordine dei metodi accoppiati (per efficienza, hanno in comune quasi tutte le valutazioni di funzione) di cui quello di ordine maggiore è usato per stimare l'errore in quello di ordine minore.

Esempio $\begin{cases} y' = \cos(x) \\ y(0) = 0 \end{cases}$

anonymous function

```
fxy=@(x,y) cos(x);
[x,y]=ode23(fxy,[0 4*pi],0);
```

oppure function file

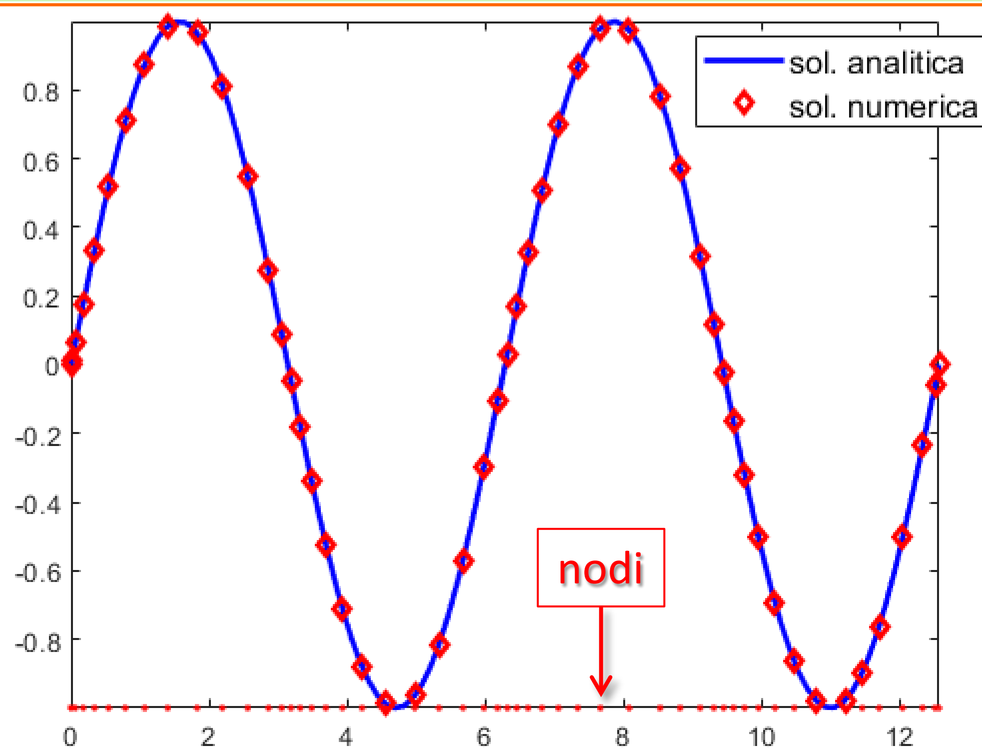
```
% function file fun(x,y)
function yprime=fun(x,y)
    yprime=cos(x);
end
```

```
[x,y]=ode23(@fun,[0 4*pi],0);
```

oppure local function

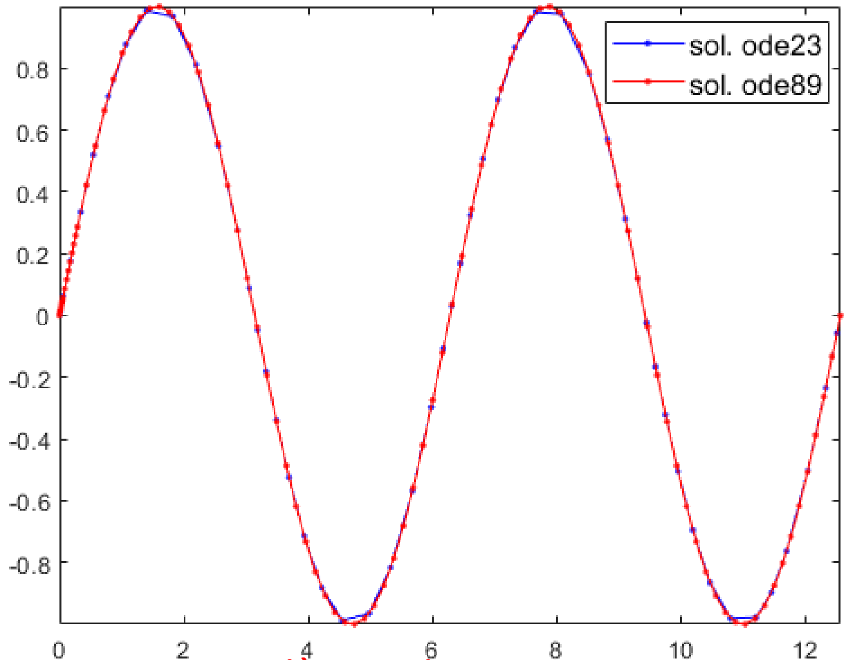
```
[x,y]=ode23(@fxy_loc,[0 4*pi],0);
xx= ... legend(…
%% local function %%
function yprime = fxy_loc(x,y)
    yprime = cos(x);
end
```

```
xx=linspace(0,4*pi,100); yy=sin(xx); % soluzione analitica
plot(xx,yy,'b',x,y,'dr','LineWidth',2)
axis tight; hold on; AX=axis; plot(x,AX(3)*ones(size(x)),'.r')
legend('sol. analitica','sol. numerica','FontSize',12)
```



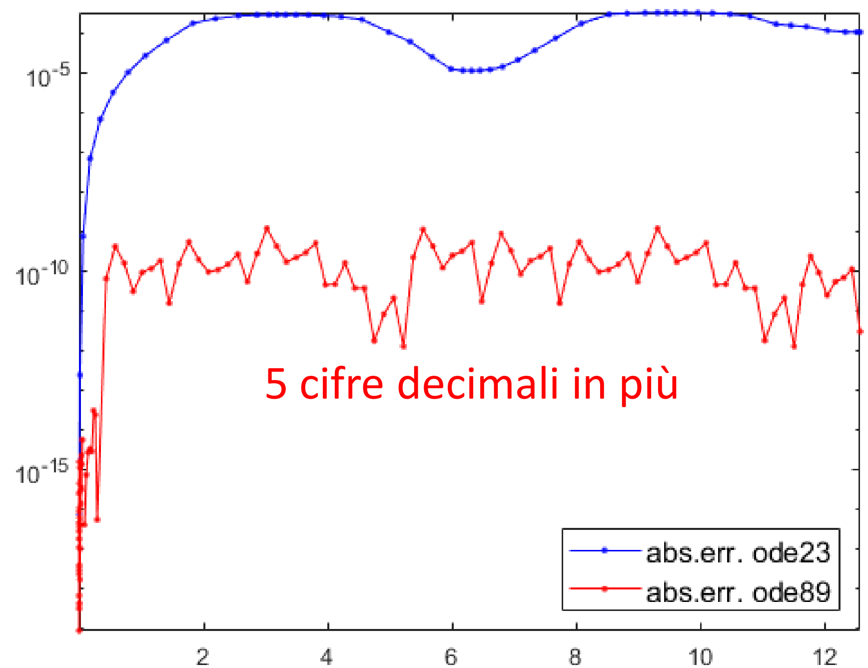
Confronto delle soluzioni numeriche di ode23 e ode89

low order method: ode23
VS
high order method: ode89



più punti

```
fxy=@(x,y) cos(x);
[x,y]=ode23(fxy,[0 4*pi],0);
disp([numel(x) max(abs(y-sin(x)))])
          55    0.00033626
[X,Y]=ode89(fxy,[0 4*pi],0);
disp([numel(X) max(abs(Y-sin(X)))])
         121    1.2218e-09
```



ode89: più accurato

```
option=odeset('RelTol',1e-10);
[x,y]=ode23(fxy,[0 4*pi],0,option);
disp([numel(x) max(abs(y-sin(x)))])
         360    6.7321e-07
[X,Y]=ode89(fxy,[0 4*pi],0,option);
disp([numel(X) max(abs(Y-sin(X)))])
          81    1.2204e-09
```

Esempio: problema predatore-preda

$y_1 = y_1(t)$ numero prede al tempo t

$y_2 = y_2(t)$ numero predatori al tempo t

x_b, x_d tassi costanti di accrescimento e morte naturale prede ($x_b > x_d$)

$x'(t) = \alpha x - \beta xy$, $\alpha > 0$ (α tasso di crescita), $\beta > 0$: dove αx=crescita esponenziale popolazione prede in assenza di predatori e βxy=velocità di predazione sulle prede.

$y'(t) = \delta xy - \gamma y$, $\gamma > 0$, $\delta > 0$: δxy=crescita della popolazione di predatori e γy=tasso di perdita dei predatori per morte naturale o per emigrazione; esso porta ad un decadimento esponenziale in assenza di prede.

Equazioni di Lotka-Volterra

$$\begin{cases} y_1' = \alpha y_1 - \beta y_1 y_2 \\ y_2' = -(\gamma y_2 - \delta y_1 y_2) \\ y_1(t_0) = y_1^0 \\ y_2(t_0) = y_2^0 \end{cases}$$

$$\alpha = 1$$

$$\begin{cases} y_1' = \alpha y_1 \left(1 - \dfrac{\beta}{\alpha} y_2\right) = \alpha y_1 \left(1 - \dfrac{y_2}{\mu_2}\right) \\ y_2' = -(\gamma y_2 - \delta y_1 y_2) = -\gamma y_2 \left(1 - \dfrac{y_1}{\mu_1}\right) \\ y_1(0) = \eta_1 \\ y_2(0) = \eta_2 \end{cases}$$

$$\gamma = 1$$

$$\begin{cases} y_1' = y_1 \left(1 - \dfrac{y_2}{\mu_2}\right) \\ y_2' = -y_2 \left(1 - \dfrac{y_1}{\mu_1}\right) \\ y_1(0) = \eta_1 \\ y_2(0) = \eta_2 \end{cases}$$

Se risulta $y_1 = \mu_1$ e $y_2 = \mu_2$, allora si ha

$$y_1' = 0 \quad \text{e} \quad y_2' = 0$$

cioè entrambe le popolazioni rimangono costanti: il punto (μ_1, μ_2) è di equilibrio.

Esempio: problema predatore-preda (vers. 1)

$$\begin{cases} y_1{}' = y_1\left(1-\dfrac{y_2}{\mu_2}\right) \\ y_2' = -y_2\left(1-\dfrac{y_1}{\mu_1}\right) \\ y_1(0)=\eta_1 \\ y_2(0)=\eta_2 \end{cases} \quad \vec{\eta}$$

```
eta=[400 100];
mu=[300 200]'; period=6.5357; Tmax=3*period;
[t,y]=ode45(@(t,y)fxy(t,y,mu),[0 Tmax],eta);
```

```
% “local function” oppure “function file”
function yprime = fxy(t,y,mu)
    yprime(1,1)= y(1).*(1 - y(2)/mu(2));
    yprime(2,1)=-y(2).*(1 - y(1)/mu(1));
end
```

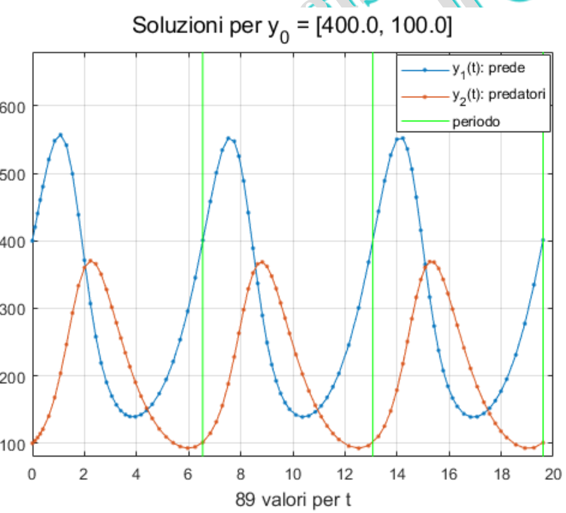
stima il periodo del fenomeno mediante FFT

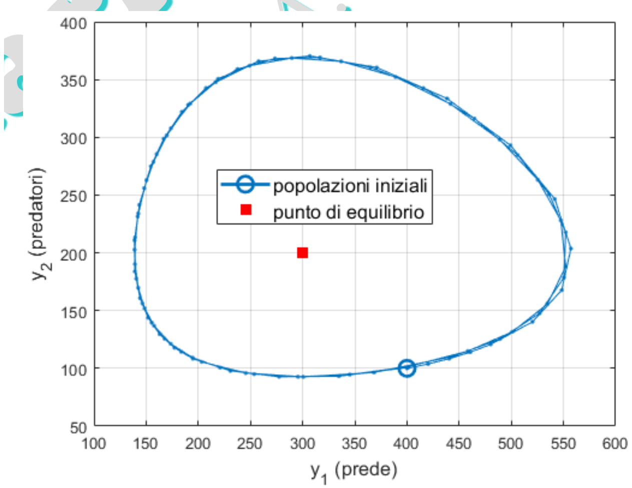
```
N=size(y,1); if rem(N,2)==1, N=N-1; end
Y1=fftshift(fft(y(:,1),N)); Y1=[Y1;Y1(1)];
Y2=fftshift(fft(y(:,2),N)); Y2=[Y2;Y2(1)];
n=(-N/2:N/2)'/Tmax; p=1./n; % periodo stimato
```

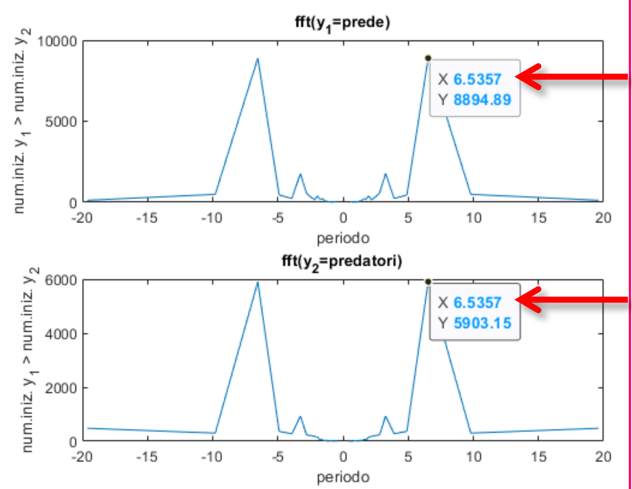
usa la Trasformata di Fourier

condizioni iniziali

$\vec{\eta} = [400 \;\; 100]$







Esempio: problema predatore-preda (vers. 2)

$$\begin{cases} y_1' = y_1\left(1-\dfrac{y_2}{\mu_2}\right) \\ y_2' = -y_2\left(1-\dfrac{y_1}{\mu_1}\right) \\ y_1(0)=\eta_1 \\ y_2(0)=\eta_2 \end{cases} \quad \vec{\eta}$$

$$y' = \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} + \\ - \end{pmatrix} \odot \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \odot \left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1/\mu_2 \\ 1/\mu_1 \end{pmatrix} \odot \begin{pmatrix} y_2 \\ y_1 \end{pmatrix} \right]$$

$\odot$ prodotto di Hadamard (element-wise product)

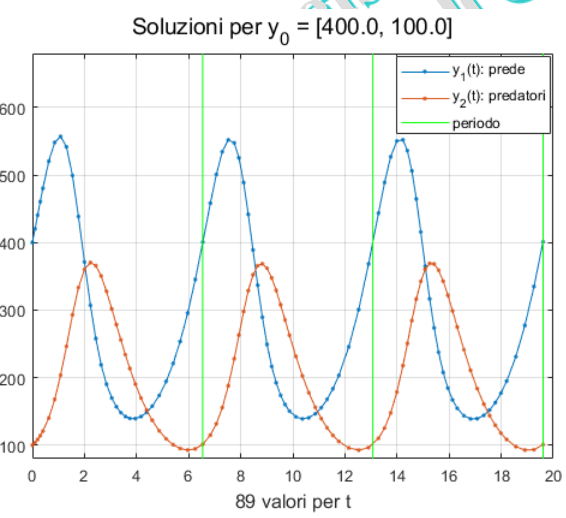
$$y' = \begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} + \\ - \end{pmatrix} \odot y \odot \left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}}_{\texttt{flipud}} \left(\mu^{-1} y\right) \right], \quad \mu = \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix}$$

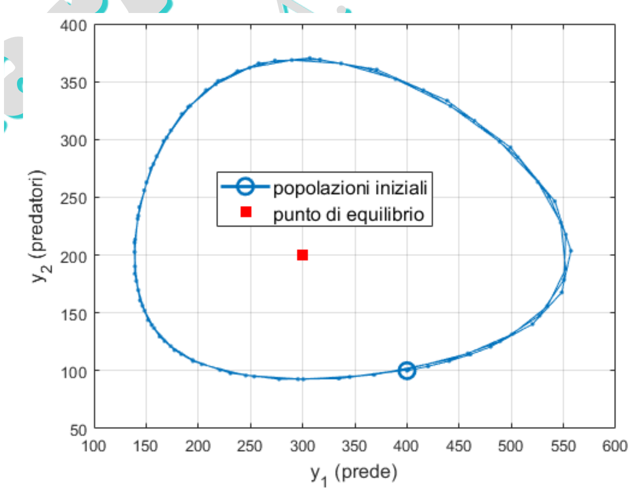
```
signs=[1 -1]'; period=6.5357;
mu=[300 200]'; eta=[...]; Tmax=3*period;
yprime=@(t,y)signs.*y.*(1-flipud(y./mu));
[t,y]=ode45(yprime,[0 Tmax],eta);
```

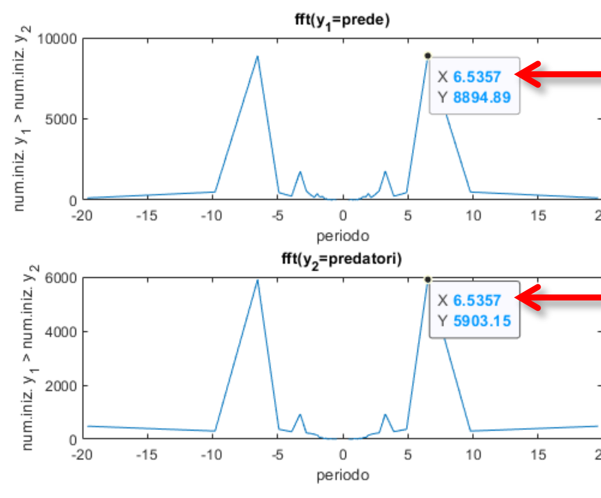
stima il periodo del fenomeno mediante FFT

```
N=size(y,1); if rem(N,2)==1, N=N-1; end
Y1=fftshift(fft(y(:,1),N)); Y1=[Y1;Y1(1)];
Y2=fftshift(fft(y(:,2),N)); Y2=[Y2;Y2(1)];
n=(-N/2:N/2)'/Tmax; p=1./n; % periodo stimato
```

$$\vec{\eta} = [400\ 100]$$

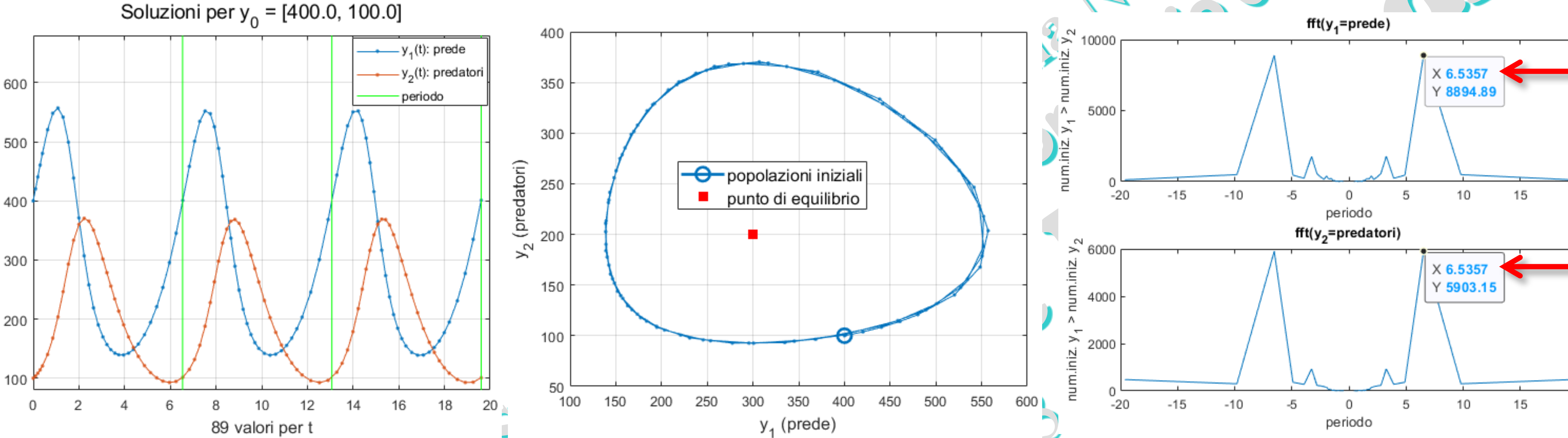




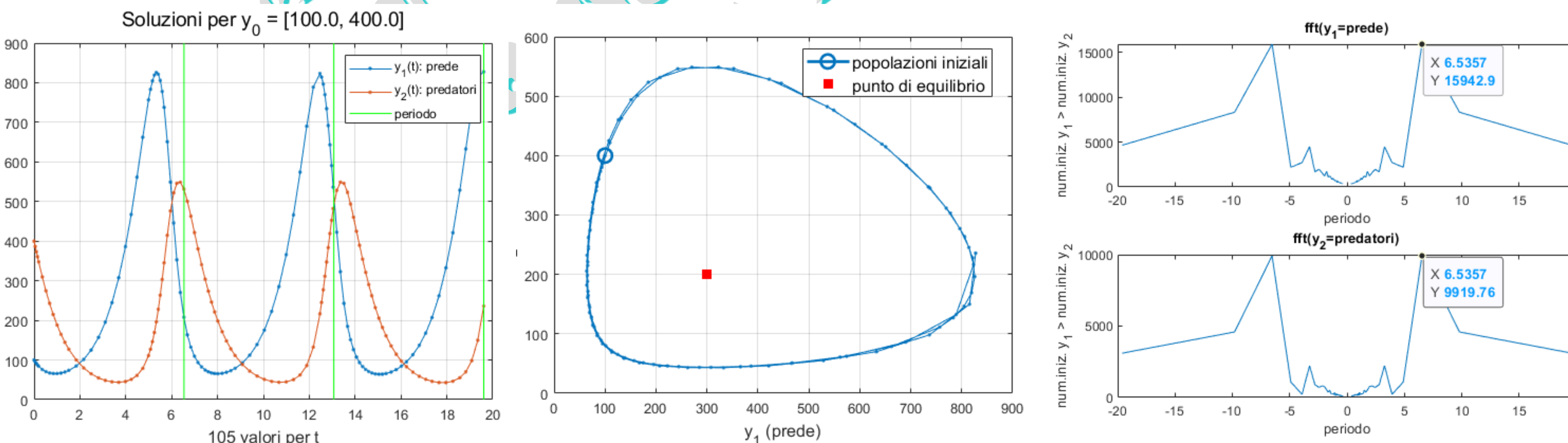


Esempio: problema predatore-preda (vers. 2)

$\vec{\eta} = [400\ 100]$: Numero iniziale prede > numero iniziale predatori



$\vec{\eta} = [100\ 400]$: Numero iniziale prede < numero iniziale predatori



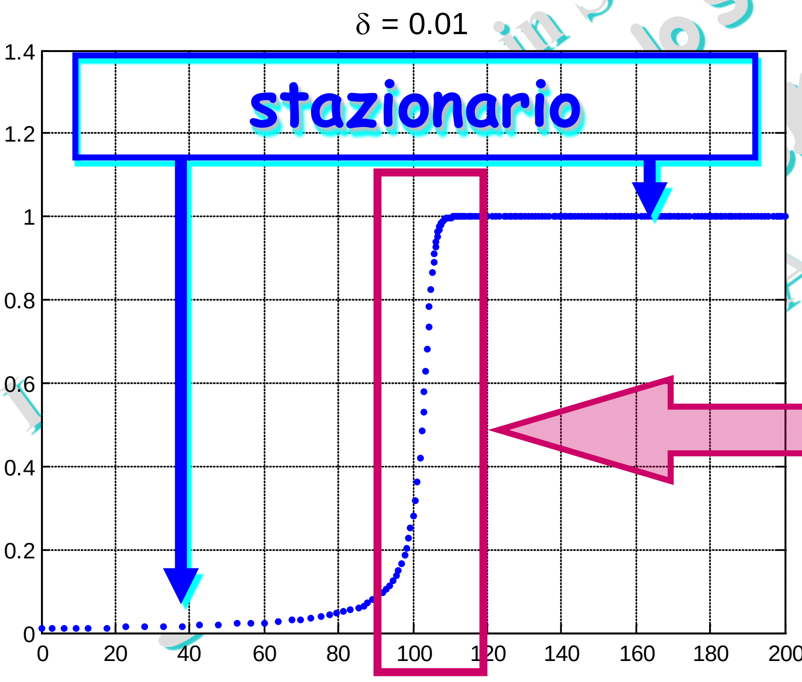
(prof. M. Rizzardi)

ODE stiff

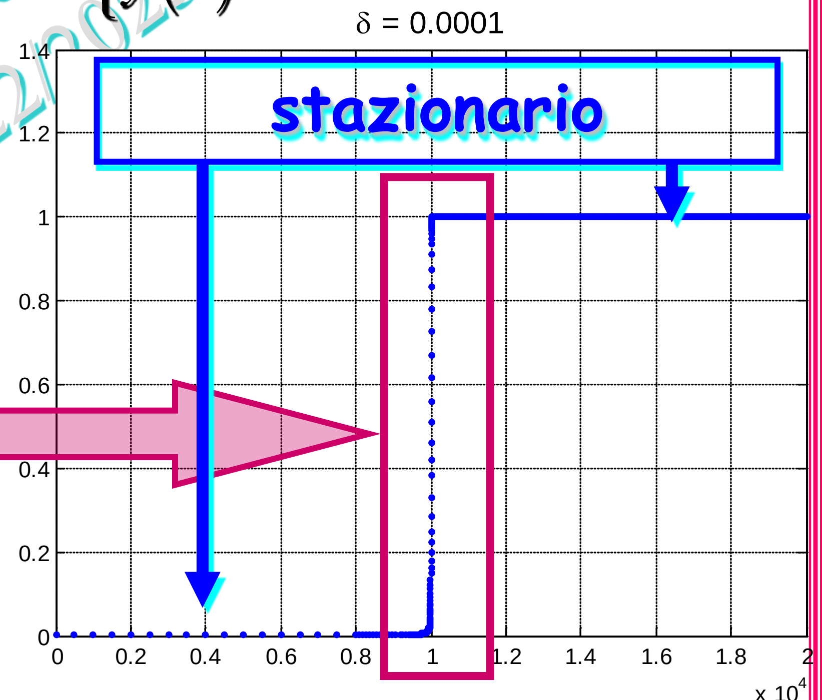
Un ***Problema di ODE*** si dice **stiff** se la sua soluzione $y=y(x)$ contiene un termine (**transiente**) che subisce, nell'intervallo considerato, una rapida variazione. La risoluzione numerica richiede passi molto piccoli per ricostruire soddisfacentemente la soluzione.

Esempio

$$\begin{cases} y' = y^2 - y^3 \\ y(0) = \delta \end{cases} \quad t \in \left[0, \frac{2}{\delta}\right]$$



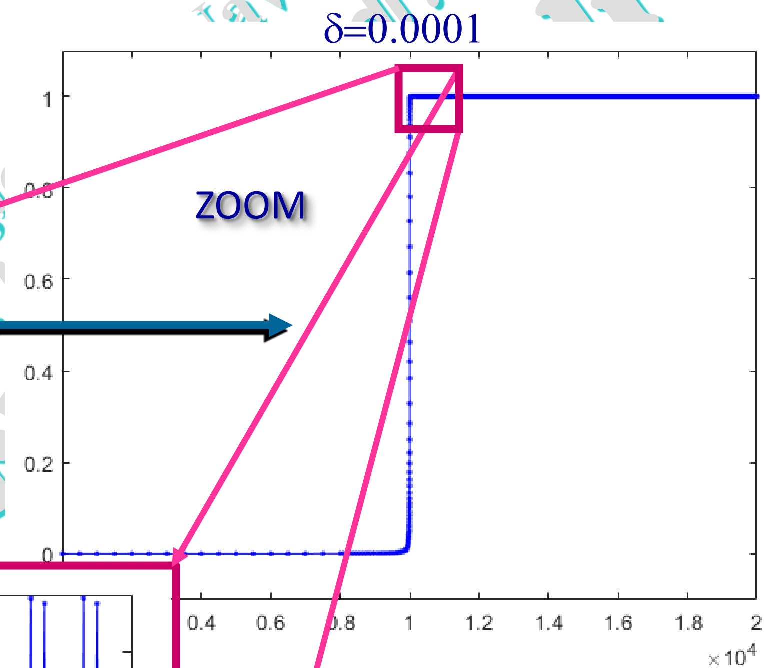
transiente



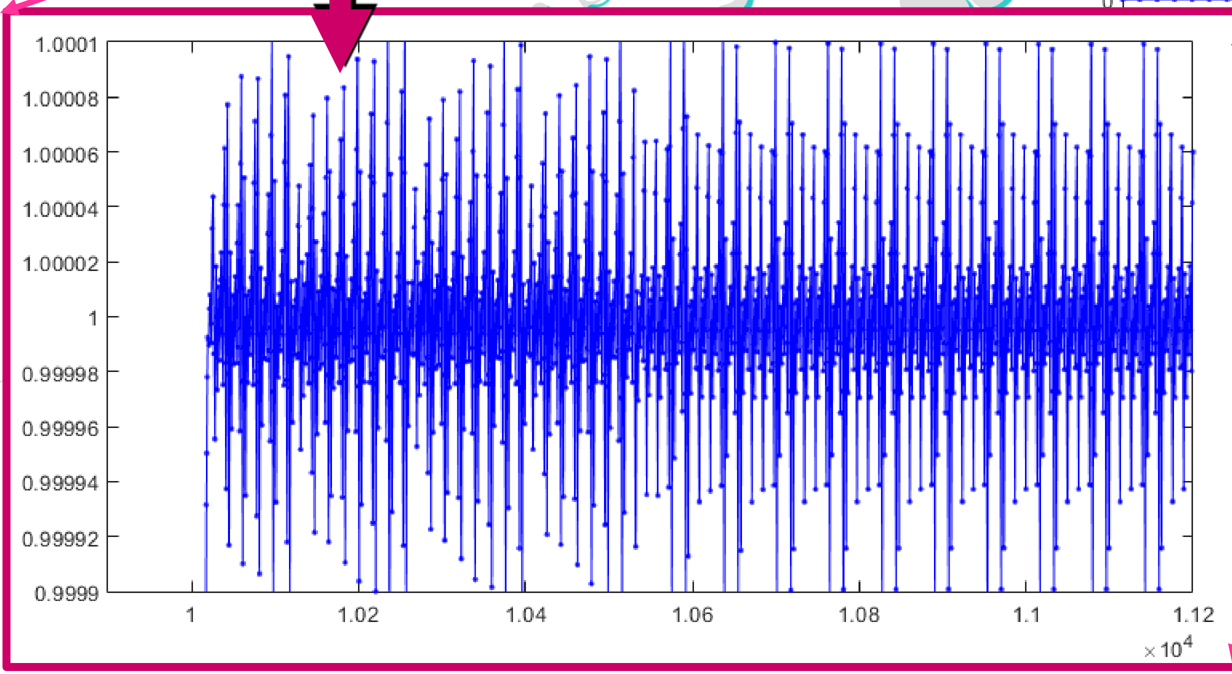
La risoluzione numerica di **problemi stiff** può richiedere passi molto piccoli per ricostruire in modo soddisfacente la soluzione

ODE Solver per problemi non stiff

```
d=0.0001; F=@(t,y) y^2-y^3;
opts=odeset('RelTol',1.e-4);
[x,y]=ode45(F,[0 2/d],d,opts);
plot(x,y,'.b-'); axis([0 2/d -0.1 1.1])
axis([9900 11200 0.9999 1.0001])
```



ZOOM



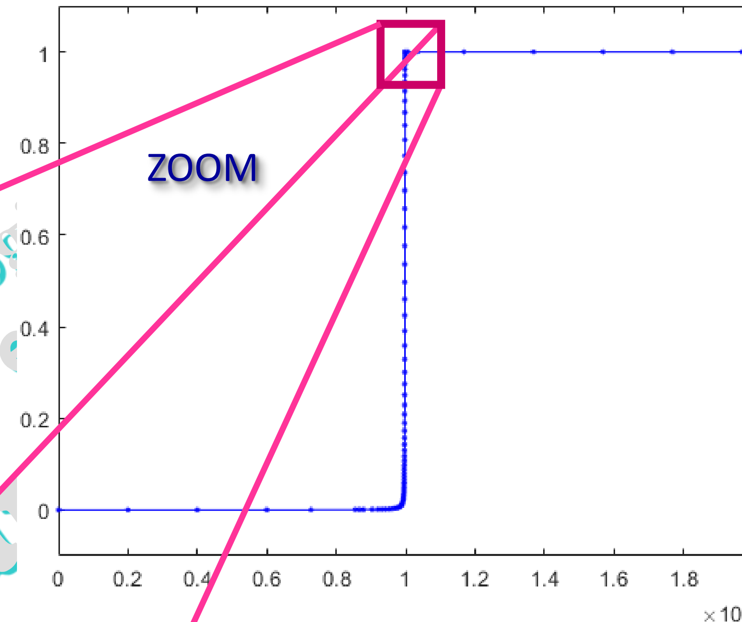
la soluzione di **ODE45** oscilla

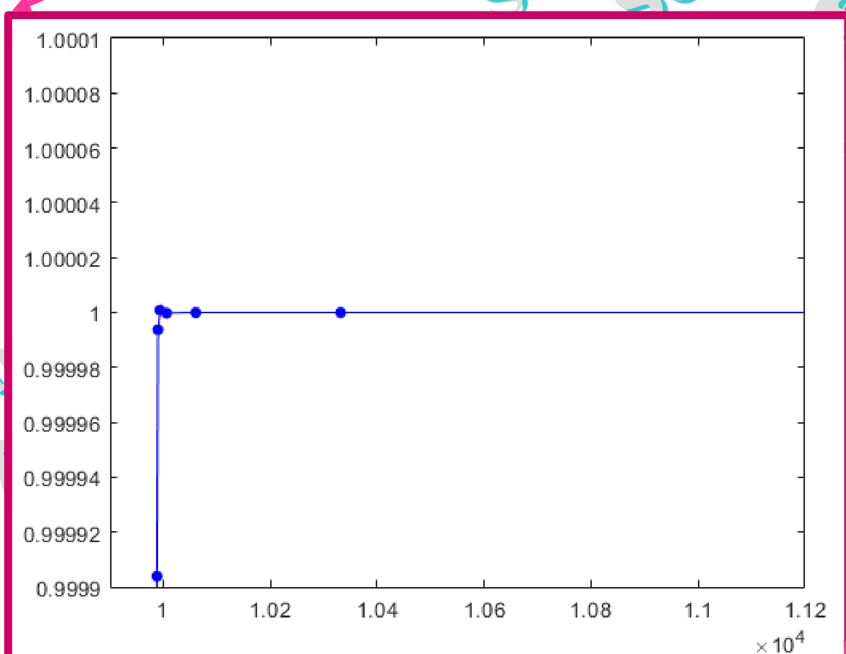
se si usa una funzione *MATLAB* per problemi stiff ...

```
d=0.0001; F=@(t,y) y^2-y^3;
opts=odeset('RelTol',1.e-4);
[x,y]=ode23s(F,[0 2/d],d,opts);
plot(x,y,'.b-'); axis([0 2/d -0.1 1.1])
axis([9900 11200 0.9999 1.0001])
```

s sta per *stiff*

δ=0.0001





la soluzione di **ODE23s** **non oscilla** ed il passo aumenta dove la soluzione è stazionaria

Problemi ai limiti (BVP)

In generale, per determinare la soluzione di una ODE di ordine m sono necessarie m condizioni.

Per $m \geq 2$ tali condizioni possono essere assegnate:

- In un sol punto (Problemi a Valori Iniziali).
- Su punti differenti (Problemi con condizioni al contorno o Problemi ai limiti).

A differenza dei IVP, per i quali esiste il Teor. di Cauchy-Peano, per i BVP la regolarità della $f(x,y)$ non basta per assicurare l'esistenza e/o l'unicità della soluzione.

Esempio: $y''(x) + y(x) = 0$ ⇒ soluzione generale: $y(x) = C_1 \sin x + C_2 \cos x$

condizioni

$$\begin{cases} y(0) = 0 \\ y\left(\pi/2\right) = 1 \end{cases}$$

⇓

unica soluzione particolare: $y(x) = \sin x$

condizioni

$$\begin{cases} y(0) = 0 \\ y(\pi) = 1 \end{cases}$$

⇓

nessuna soluzione

condizioni

$$\begin{cases} y(0) = 0 \\ y(\pi) = 0 \end{cases}$$

⇓

infinite soluzioni $y(x) = C_1 \sin x, \ \forall C_1$

Il problema si dice **ben posto** quando sono assegnate delle condizioni ai limiti che, in presenza di soluzioni, ne garantiscono l'unicità

Particolare BVP: problema dei due punti

In generale, il problema dei due punti si formula nel seguente modo:

- Problemi del primo ordine:

$$\begin{cases} y' = f(x,y) & x \in \,]a,b[\\ \psi\big(y(a),y(b)\big) = 0 \end{cases}$$

- Problemi del secondo ordine:

$$\begin{cases} y'' = f(x,y,y') & x \in \,]a,b[\\ \psi_1\big(y(a),y'(a),y(b),y'(b)\big) = 0 \\ \psi_2\big(y(a),y'(a),y(b),y'(b)\big) = 0 \end{cases}$$

Sono di particolare interesse le seguenti condizioni ai limiti:

$$\begin{cases} y(a) = \alpha \\ y(b) = \beta \end{cases}$$ condizioni di Dirichlet

$$\begin{cases} y'(a) = \gamma \\ y'(b) = \delta \end{cases}$$ condizioni di Neumann

$$\begin{cases} y(a) - y(b) = 0 \\ y'(a) - y'(b) = 0 \end{cases}$$

condizioni di periodicità

$$\begin{cases} \psi_1\big(y(a),y'(a)\big) = 0 \\ \psi_2\big(y(b),y'(b)\big) = 0 \end{cases}$$

condizioni di Sturm-Liouville

Teorema di esistenza e unicità della soluzione per il problema dei due punti

$$\begin{cases} y'' = f\left(x, y, y'\right) & x \in \left]a, b\right[\\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

condizioni di Dirichlet

Si supponga che le derivate parziali $\partial f/\partial y$ e $\partial f/\partial y'$ esistano, siano continue e soddisfino le condizioni

$$\frac{\partial f}{\partial y}\left(x, y, y'\right) > 0$$

$$\left|\frac{\partial f}{\partial y'}\left(x, y, y'\right)\right| \leq M$$

$\forall\, x \in [a, b]$ e $\forall\, y, y' \in \mathbb{R}$.

Allora esiste ed è unica la soluzione del problema.

<u>Caso particolare</u>: $\begin{cases} y'' = \Phi(x)\, y + \psi(x) & x \in \left]a, b\right[\\ y(a) = \alpha, \quad y(b) = \beta \end{cases}$ se: $\dfrac{\partial f}{\partial y}(x, y) = \Phi(x) > 0$

⇒ Esiste ed è unica la soluzione $\forall \alpha, \beta$.

Risoluzione numerica del problema dei due punti mediante differenze finite

Si discretizza $[a,b]$ con $n+1$ punti di griglia x_k

$$x_0 = a < x_2 < \cdots < x_n = b, \quad h = (\mathrm{b}-\mathrm{a})/n, \quad x_k = a + kh, \; k=0,1,\ldots,n$$

e, nei punti di griglia (x_k, y_k) dove $y_k \approx y(x_k)$, si approssimano le derivate dell'eq. differenziale mediante differenze finite trasformando il problema differenziale in un sistema di equazioni algebriche, lineari o non lineari in funzione del tipo di ODE.

Differenze finite:

$$y'(x_k) = \frac{y(x_{k+1}) - y(x_k)}{h} + O(h)$$ differenza prima in avanti

$$y'(x_k) = \frac{y(x_k) - y(x_{k-1})}{h} + O(h)$$ differenza prima all'indietro

entrambe del 2° ordine

$$\boxed{y'(x_k)} = \frac{y(x_{k+1}) - y(x_{k-1})}{2h} + \boxed{O(h^2)}$$ differenza prima centrale

$$\boxed{y''(x_k)} = \frac{y(x_{k+1}) - 2y(x_k) + y(x_{k-1})}{h^2} + \boxed{O(h^2)}$$ differenza seconda centrale

Risoluzione numerica del problema dei due punti mediante differenze finite

$$\begin{cases} y'' = f(x,y,y') & x \in]a,b[\\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

Discretizzazione con differenze finite del 2° ordine:

$$y'(x_k) \approx \frac{y_{k+1} - y_{k-1}}{2h}$$

$$y''(x_k) \approx \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2}$$

ODE: $$y_{k+1} - 2y_k + y_{k-1} - h^2 f\left(x_k, y_k, \frac{y_{k+1} - y_{k-1}}{2h}\right) = 0, \qquad k = 1,2,\ldots n-1$$

È un sistema di $n-1$ equazioni algebriche nelle $n-1$ incognite y_1, y_2, …, y_{n-1}; ogni equazione contiene solo 3 incognite: y_{k-1}, y_k, y_{k+1}.

Il sistema è lineare (tridiagonale) se $f(x,y,y')$ è lineare in y e y'. Esempio:

$$\begin{cases} y'' = -y & x \in \left]0, \frac{\pi}{2}\right[\\ y(0) = 0, \quad y\left(\frac{\pi}{2}\right) = 2 \end{cases} \Rightarrow$$

$$y_{k+1} - 2y_k + y_{k-1} + h^2 y_k = 0, \qquad k = 1,2,\ldots n-1$$

$$y_{k+1} + y_k\left(h^2 - 2\right) + y_{k-1} = 0, \qquad k = 1,2,\ldots n-1$$

equazione lineare nelle y_{k-1}, y_k, y_{k+1}

Il sistema è non lineare se $f(x,y,y')$ è non lineare in y e y'. Esempio:

$$\begin{cases} y'' = -3yy' & x \in]0,2[\\ y(0) = 2, \quad y(2) = 1 \end{cases} \Rightarrow$$

$$y_{k+1} - 2y_k + y_{k-1} + 3h^2 y_k \frac{y_{k+1} - y_{k-1}}{2h} = 0, \qquad k = 1,2,\ldots n-1$$

equazione non lineare nelle y_{k-1}, y_k, y_{k+1}

Esempio: ODE lineare

$$\begin{cases} y'' = -y & x \in \left]0, \frac{\pi}{2}\right[\\ y(0) = 0, \quad y\left(\frac{\pi}{2}\right) = 2 \end{cases}$$

soluzione analitica: $y(x) = 2\sin(x)$

schema alle differenze finite

$$y_{k-1} + y_k\left(h^2 - 2\right) + y_{k+1} = 0, \quad k = 1,2,\ldots n-1$$

$$n = 4 \quad \Rightarrow \quad h = \frac{b-a}{n} = \frac{\pi/2}{4} = \frac{\pi}{8}$$

$y_0 = 0$

$$\begin{array}{ccccccccc} y_0 & + & y_1\left(h^2-2\right) & + & y_2 & & & = & 0 \quad (k=1) \\ & & y_1 & + & y_2\left(h^2-2\right) & + & y_3 & = & 0 \quad (k=2) \\ & & & & y_2 & + & y_3\left(h^2-2\right) \; + \; y_4 & = & 0 \quad (k=3) \end{array}$$

$y_4 = 2$

$$\begin{array}{cccccc} y_1\left(h^2-2\right) & + & y_2 & & = & 0 \\ y_1 & + & y_2\left(h^2-2\right) & + \; y_3 & = & 0 \\ & & y_2 & + \; y_3\left(h^2-2\right) & = & -2 \end{array} \qquad \begin{pmatrix} \left(h^2-2\right) & 1 & 0 \\ 1 & \left(h^2-2\right) & 1 \\ 0 & 1 & \left(h^2-2\right) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -2 \end{pmatrix}$$

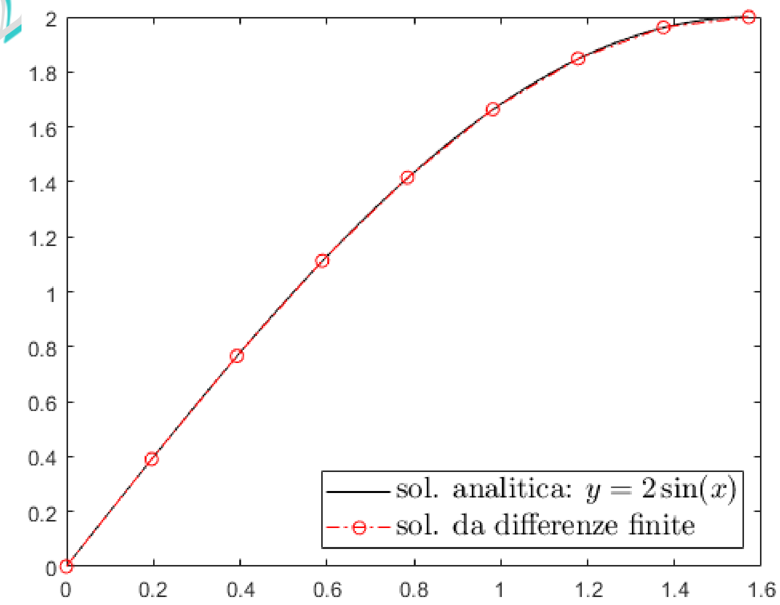
matrice tridiagonale

Esempio: ODE lineare

1) Implementazione MATLAB con matrice piena

$$\begin{pmatrix} (h^2-2) & 1 & & & & & \\ 1 & (h^2-2) & 1 & & & & \\ & 1 & (h^2-2) & 1 & & & \\ & & 1 & (h^2-2) & 1 & & \\ & & & 1 & (h^2-2) & 1 & \\ & & & & 1 & (h^2-2) & 1 \\ & & & & & 1 & (h^2-2) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix}$$

```
a=0; b=pi/2; trueY=@(x) 2*sin(x);
X=linspace(a,b,100)'; Y=trueY(X);
n=8; y0=[0 2]; % condizioni ai limiti
n=8; h=(b-a)/n; xk=linspace(a,b,n+1)';
d=h^2-2; D=d*eye(n-1);
o=ones(n-2,1); E=diag(o,-1) + diag(o,+1);
A=D + E; % A: matrice dei coefficienti, v: termini noti
v=[-y0(1); zeros(n-3,1); -y0(2)];
yk=A\v;
yk=[y0(1);yk;y0(end)];
figure; plot(X,Y,'k',xk,yk,'or:')
figure; spy()
```



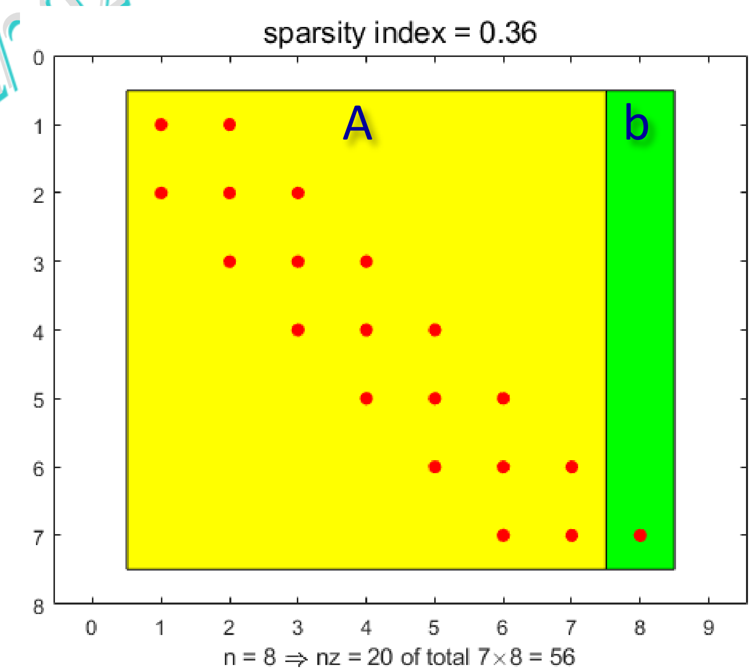
Esempio: ODE lineare

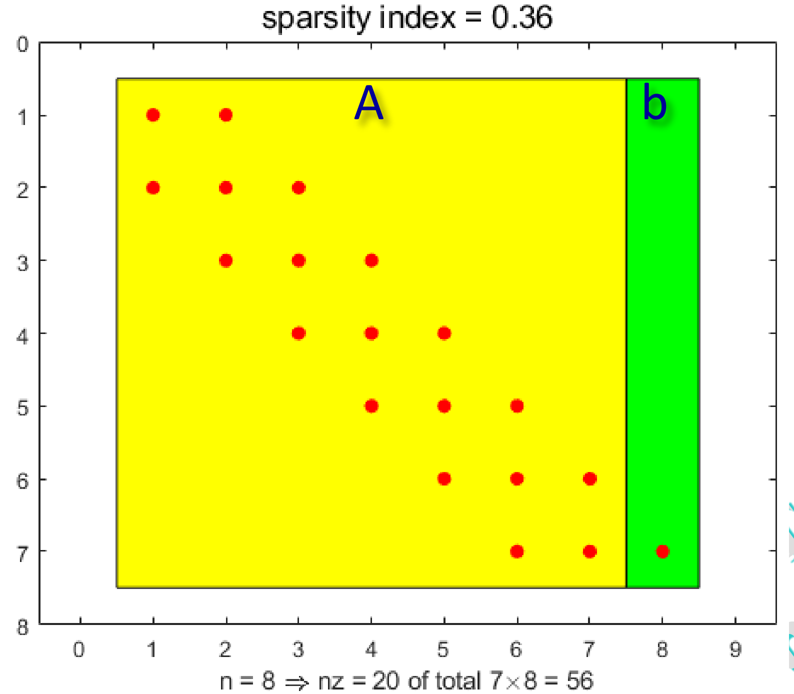
2) Implementazione MATLAB con matrice sparsa

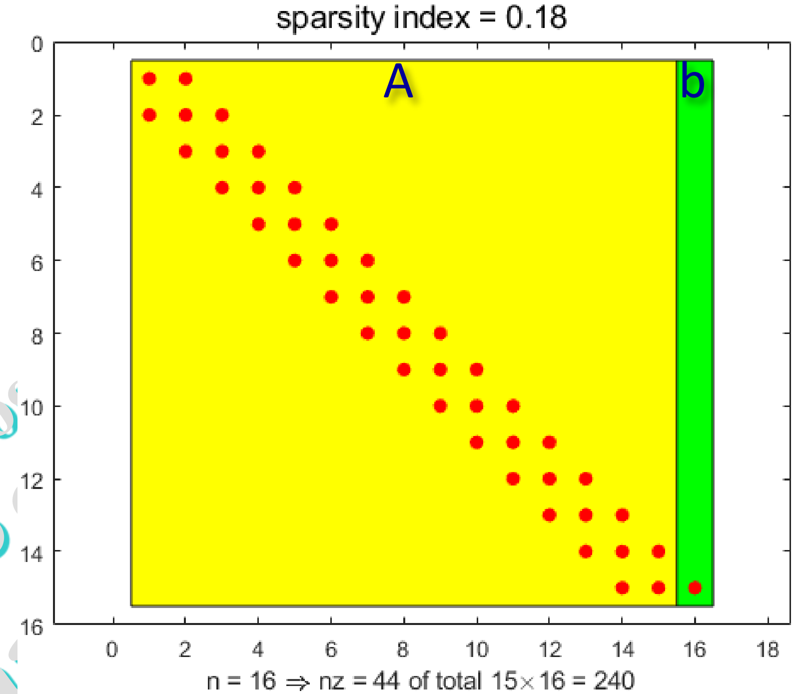
$$
\begin{pmatrix}
(h^2-2) & 1 & & & & & \\
1 & (h^2-2) & 1 & & & & \\
 & 1 & (h^2-2) & 1 & & & \\
 & & 1 & (h^2-2) & 1 & & \\
 & & & 1 & (h^2-2) & 1 & \\
 & & & & 1 & (h^2-2) & 1 \\
 & & & & & 1 & (h^2-2)
\end{pmatrix}
\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix}
=
\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix}
$$

Si può usare l'algoritmo di Thomas per risolvere il sistema tridiagonale

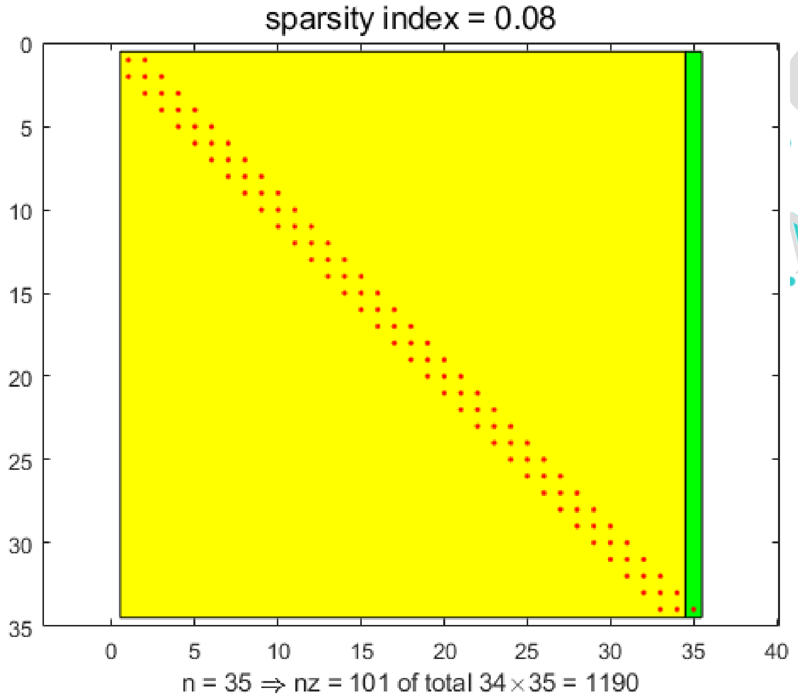
```
a=0; b=pi/2; trueY=@(x) 2*sin(x);
X=linspace(a,b,100)'; Y=trueY(X);
n=8; y0=[0 2]; % condizioni ai limiti
n=8; h=(b-a)/n; xk=linspace(a,b,n+1)';
d=h^2-2; D=d*speye(n-1);
o=ones(n-1,1);
E=spdiags(o,-1,n-1,n-1) + ...
                     spdiags(o,+1,n-1,n-1);
A=D + E;            % A: matrice dei coefficienti
v=sparse(n-1,1); % v: vettore dei termini noti
v(1)=-y0(1); v(n-1)=-y0(2);
yk=A\v;
yk=[y0(1);yk;y0(end)];
plot(X,Y,'k',xk,yk,'or:')
```

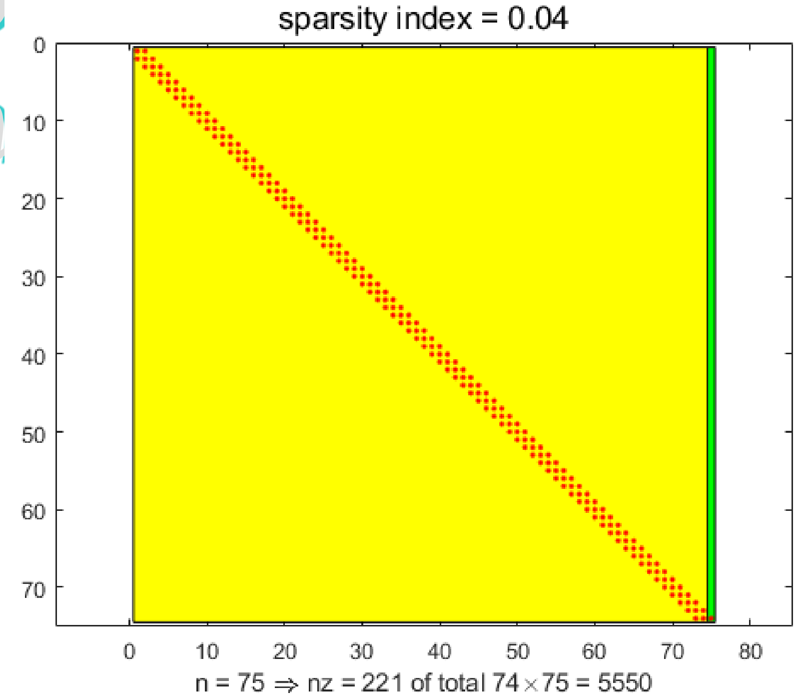






sparsità al crescere della dimensione del problema





Esempio: ODE non lineare

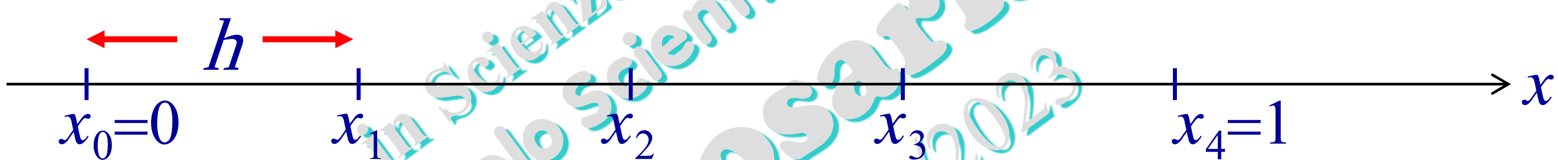
$$\begin{cases} y'' = 12\sqrt{y} & x \in \,]0,1[\\ y(0) = 0, \quad y(1) = 1 & \end{cases}$$

schema alle differenze finite

$$y_{k-1} - 2\left(y_k + 6h^2\sqrt{y_k}\right) + y_{k+1} = 0, \qquad k = 1, 2, \ldots n-1$$

soluzione analitica: $y(x) = x^4$

$$n = 4 \quad \Rightarrow \quad h = \frac{b-a}{n} = \frac{1}{4} = 0.25$$



$$y_0 = 0$$

$$\begin{aligned} y_0 - 2y_1 - 12h^2\sqrt{y_1} + y_2 &= 0 \\ y_1 - 2y_2 - 12h^2\sqrt{y_2} + y_3 &= 0 \\ y_2 - 2y_3 - 12h^2\sqrt{y_3} + y_4 &= 0 \end{aligned}$$

$$y_4 = 1$$

$$\begin{aligned} -2y_1 - 12h^2\sqrt{y_1} + y_2 &= 0 \\ y_1 - 2y_2 - 12h^2\sqrt{y_2} + y_3 &= 0 \\ y_2 - 2y_3 - 12h^2\sqrt{y_3} + 1 &= 0 \end{aligned}$$

Esempio: ODE non lineare

$$\begin{cases} y'' = 12\sqrt{y} & x \in]0,1[\\ y(0) = 0, \quad y(1) = 1 \end{cases}$$

soluzione analitica: $y(x) = x^4$

$$n = 8 \quad \Rightarrow \quad h = \frac{b-a}{n} = \frac{1}{8} = 0.125$$

parametri

```
function F = nonLinODE(y,h,N,y0)
    h2=h^2;
    F=zeros(N,1);
    F(1)=y0(1)-2*y(1)-12*h2*sqrt(y(1))+y(2);
    for j=2:N-1
        F(j)=y(j-1)-2*y(j)-12*h2*sqrt(y(j))+y(j+1);
    end
    F(N)=y(N-1)-2*y(N)-12*h2*sqrt(y(N))+y0(2);
end
```

$y_0 - 2y_1 - 12h^2\sqrt{y_1} + y_2 = 0$

for $j = 2 : n-2$

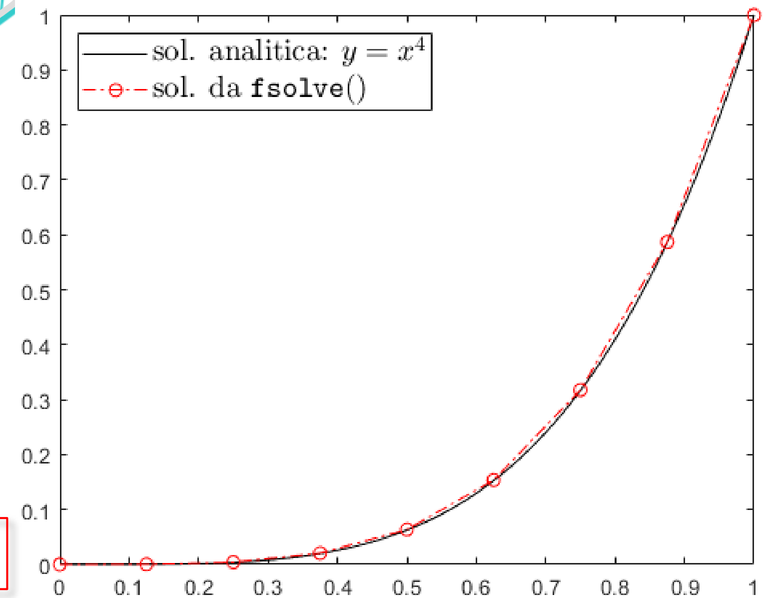
$\qquad y_{j-1} - 2y_j - 12h^2\sqrt{y_j} + y_{j+1} = 0$

end

$y_{n-2} - 2y_{n-1} - 12h^2\sqrt{y_{n-1}} + y_n = 0$

```
a=0; b=1; trueY=@(x) x.^4;
X=linspace(a,b,100)'; Y=trueY(X);
y0=[0 1]; % condizioni ai limiti
n=8; h=(b-a)/n; xk=linspace(a,b,n+1)';
Y0=ones(n-1,1); % valore iniziale per fsolve
myF=@(y) nonLinODE(y,h,n-1,y0);
yk=fsolve(myF,Y0);
yk=[y0(1);yk;y0(end)];
figure; plot(X,Y,'k',xk,yk,'or-.')
```

risolve un sistema di equazioni non lineari del tipo $F(x)=0$



MATLAB Solvers for BVP

bvp4c	Solve boundary value problem — 4th order method.
bvp5c	Solve boundary value problem — 5th order method.
bvpinit	Form initial guess for boundary value problem solver.
deval	Evaluate differential equation solution structure

I MATLAB Solver per BVP **bvp4c** e **bvp5c** risolvono problemi del tipo

$$y' = f(x,y)$$

dove x è la variabile indipendente e y quella dipendente. Nel caso di two-point BVP, le condizioni alla frontiera devono essere espresse nella forma

$$g[y(a),y(b)] = 0$$

```
sol = bvp4c (odefun, bcfun, solinit);
sol = bvp4c (odefun, bcfun, solinit, options);
sol = bvp5c (odefun, bcfun, solinit);
sol = bvp5c (odefun, bcfun, solinit, options);
```

Esempio: MATLAB BVP Solver per ODE lineare

$$\begin{cases} y'' = -y & x \in \left]0, \frac{\pi}{2}\right[\\ y(0) = 0, \quad y\left(\frac{\pi}{2}\right) = 2 \end{cases}$$

soluzione analitica: $y(x) = 2\sin(x)$

$$\begin{cases} \boxed{\begin{array}{l} y_1' = y_2 \\ y_2' = -y_1 \end{array}} & x \in \left]0, \frac{\pi}{2}\right[\\ \psi = \left[y_1(0), \quad y_1\left(\frac{\pi}{2}\right) - 2 \right]^T = \vec{0} \end{cases}$$

sistema del 1° ordine

condizioni ai limiti in forma: $\psi(y(a), y(b)) = 0$

```
function yprime = bvpfun(x,y)
    yprime = zeros(2,1);
    yprime = [ y(2); -y(1)];
end
```

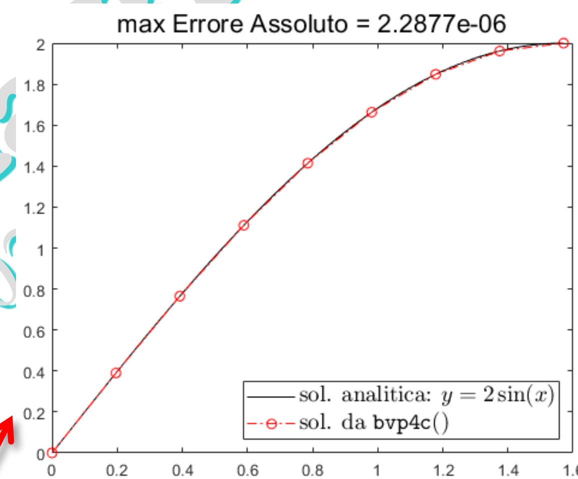
```
function res = bcfun(ya,yb)
    res = [ya(1); yb(1)-2];
end
```

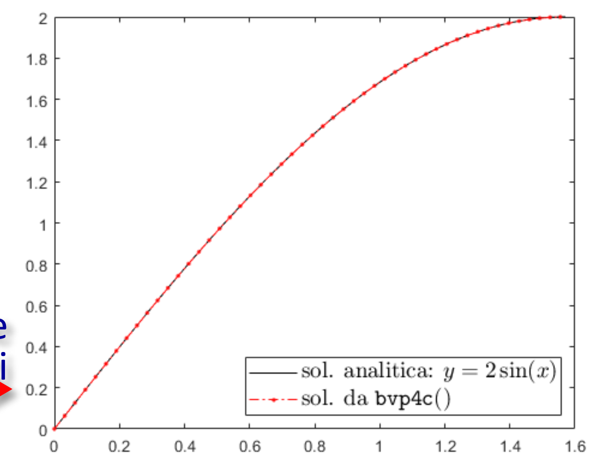
stima iniziale

```
function g = guess(x)
    g=[1;1]; % o [1;0] o [sin(x);cos(x)]
end
```

```
a=0; b=pi/2; trueY=@(x) 2*sin(x);
X=linspace(a,b,100)'; Y=trueY(X);
n=8; h=(b-a)/n; xk=linspace(a,b,n+1)';
solinit=bvpinit(xk, @guess);
sol=bvp4c(@bvpfun,@bcfun,solinit);
figure; plot(X,Y,'k',sol.x,sol.y(1,:),'or-.')
Xk=X(1:2:end); Yk=deval(sol,Xk);
figure; plot(X,Y,'k',Xk,Yk(1,:),'.r-.')
```

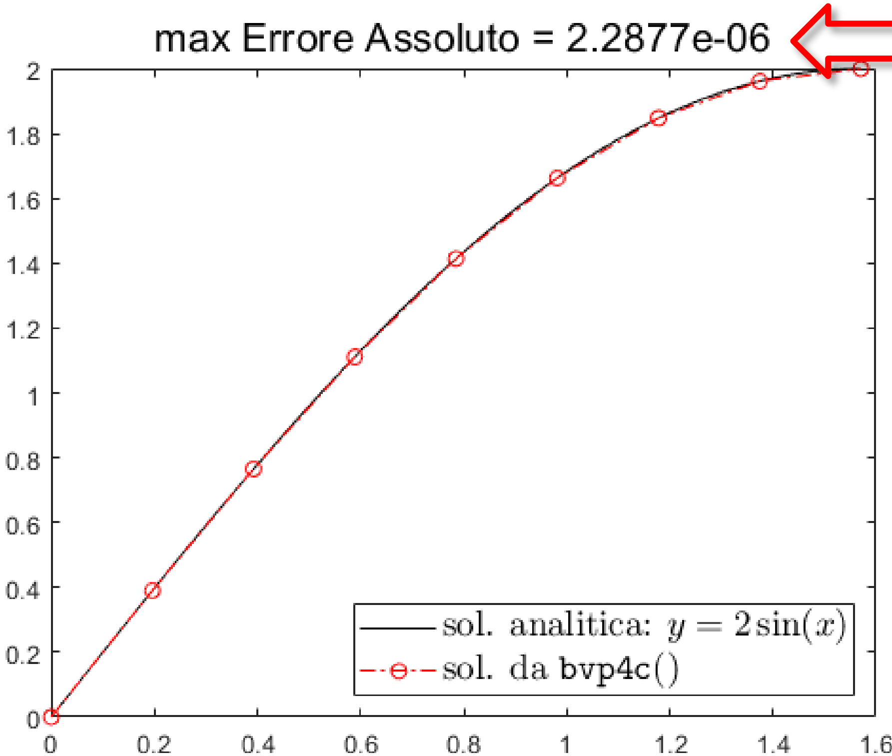
valuta la soluzione in altri punti

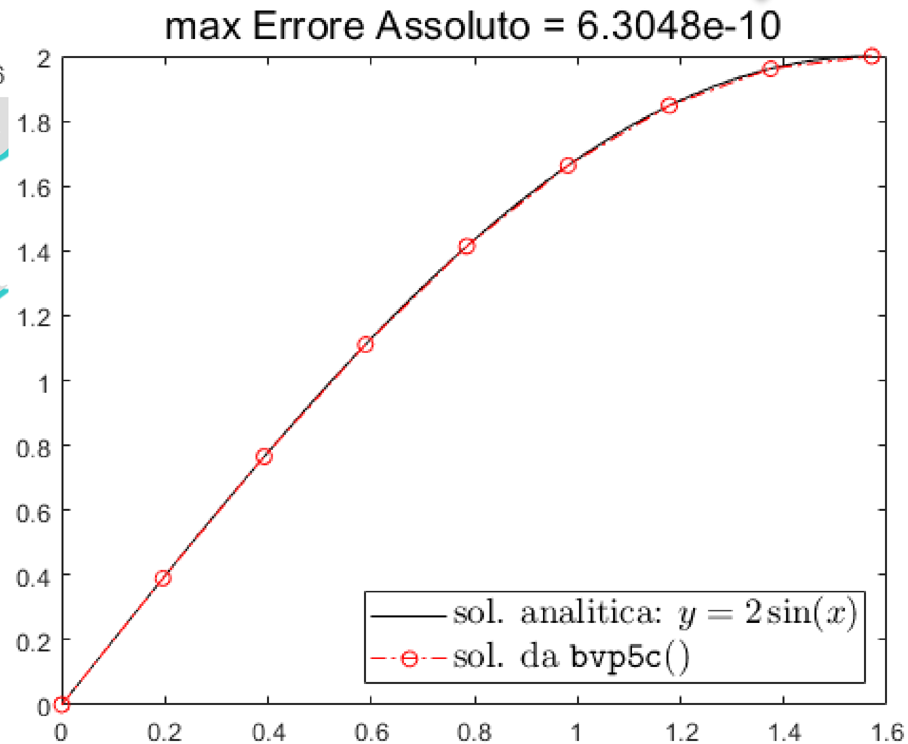




stima iniziale `g=[1;1];`

bvp4c VS bvp5c





Esempio: MATLAB BVP Solver per ODE non lineare

$$\begin{cases} y'' = 12\sqrt{y} & x \in]0,1[\\ y(0)=0, \quad y(1)=1 \end{cases}$$

soluzione analitica: $y(x) = x^4$

$$\begin{cases} \boxed{\begin{aligned} y_1' &= y_2 \\ y_2' &= 12\sqrt{y_1} \end{aligned}} & x \in]0,1[\\ \psi = [y_1(0), \quad y_1(1)-1]^T = \vec{0} \end{cases}$$

sistema del 1° ordine

condizioni ai limiti in forma: $\psi(y(a), y(b)) = 0$

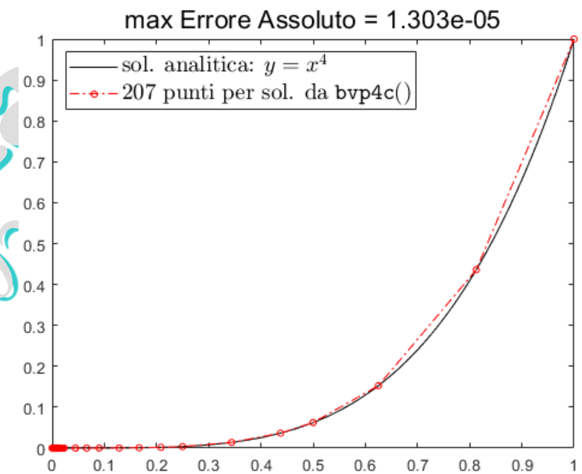
```
function yprime = bvpfun(x,y)
    yprime = zeros(2,1);
    yprime = [y(2);
              12*sqrt(y(1))];
end
```

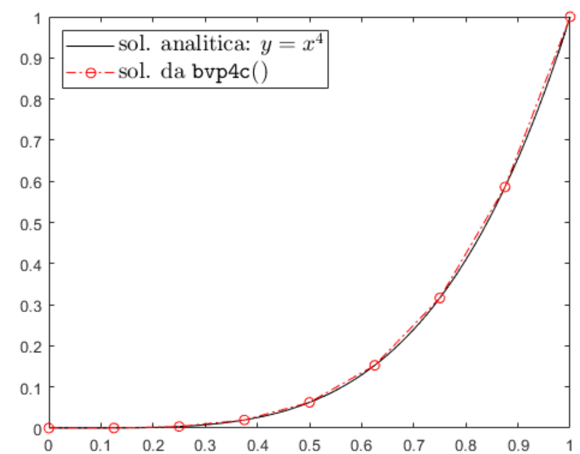
```
function res = bcfun(ya,yb)
    res = [ya(1); yb(1)-1];
end
```

stima iniziale

```
function g = guess(x)
    g=[1;1]; % o [0;0]
end
```

```
a=0; b=1; trueY=@(x) 2*sin(x);
X=linspace(a,b,100)'; Y=trueY(X);
n=8; h=(b-a)/n; xk=linspace(a,b,n+1)';
solinit=bvpinit(xk, @guess);
sol=bvp4c(@bvpfun,@bcfun,solinit);
figure; plot(X,Y,'k',sol.x,real(sol.y(1,:)),'.r-.')
yk=deval(sol,xk);
figure; plot(X,Y,'k',xk,real(yk(1,:)),'or-.')
```





stima iniziale g=[1;1];

bvp4c VS bvp5c