

Eq. differenziali ordinarie: modello matematico

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

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

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

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

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

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

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

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

1

Eq. differenziali ordinarie: problema di Cauchy

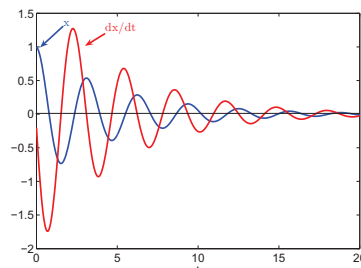
- Un **problema reale** richiede l'assegnazione delle **condizioni iniziali**.

Problema di Cauchy:

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

Esempio:

$$\left\{ \begin{array}{l} x(0) = 1 \\ \frac{dx}{dt}(0) = 0 \end{array} \right. \quad \begin{array}{l} m = 250g \\ k = 85 \text{ N/m} \\ b = 70 \text{ g/s} \end{array}$$



2

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**

$$\frac{d^2\theta}{dt^2} - \frac{g}{L} \sin \theta = 0 \quad t > t_0$$

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

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

Calcolare $\theta(t)$ nel caso in cui $L = 0.6$ m, $\theta_0 = \pi/6$ rad, $v_0 = 0$ m/s (si assuma $g = -9.81$ m/s²)

3

Equazioni differenziali di ordine n

$$\begin{cases} y^{(n)}(t) = g(t, y(t), y'(t), y''(t), \dots, y^{(n-1)}(t)) & t > t_0 \\ y^{(k)}(t_0) = y_k & k = 0, 1, \dots, n-1 \end{cases} \quad \text{Condizioni iniziali}$$

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

$$y^{(n)}(t) = g(t, \underbrace{y(t)}_{y_1(t)}, \underbrace{y'(t)}_{y_2(t)}, \underbrace{y''(t)}_{y_3(t)}, \dots, \underbrace{y^{(n-1)}(t)}_{y_n(t)})$$

$$y_1(t) = y(t)$$

$$\begin{cases} y_1'(t) = y_2(t) \\ y_2'(t) = y_3(t) \\ \dots \\ y_{n-1}'(t) = y_n(t) \\ y_n'(t) = g(t, y_1(t), y_2(t), y_3(t), \dots, y_n(t)) \end{cases} \quad \begin{cases} y_{10} = y_0 \\ y_{20} = y_1 \\ \dots \\ y_{n0} = y_{n-1} \end{cases} \quad \text{Condizioni iniziali}$$

4

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:

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

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

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

5

Eq. differenziali ordinarie del primo ordine /1

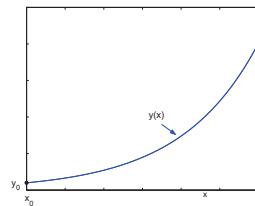
$$\text{Problema di Cauchy: } \begin{cases} y'(t) = f(t, y(t)) & t > t_0 \\ y(t_0) = y_0 \end{cases} \quad \text{condizione iniziale}$$

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

$$f(t, y(t)) = Ky(t) \quad K > 0 \quad \rightarrow \quad \begin{cases} y'(t) = Ky(t) \\ y(t_0) = y_0 \end{cases}$$

Soluzione esatta:

$$y(t) = y_0 e^{K(t-t_0)}$$



6

Eq. differenziali ordinarie del primo ordine /2

Nei **problemi applicativi**, l'espressione di $f(t, y(t))$ è **complicata** e quindi non si può calcolare esplicitamente la soluzione del **problema di Cauchy** oppure la sua espressione è data tramite funzioni non elementari.

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

Esempio

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

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

chiamata **equazione logistica**.

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

7

Eq. differenziali ordinarie del primo ordine /3

Assumiamo: *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 $d(t) = kP(t)$ proporzionale al numero di individui presenti al tempo t con $k = 1.4 \times 10^{-7}$.

Crescita di una popolazione Equazione logistica

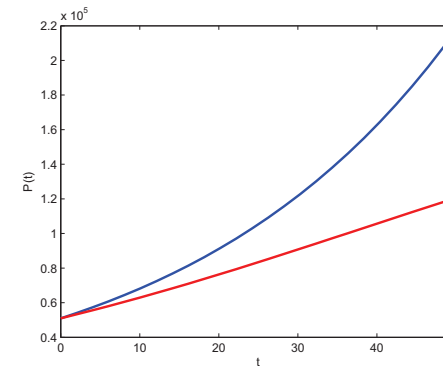
$$\begin{cases} \frac{dP(t)}{dt} = bP(t) \\ P(0) = 50976 \end{cases}$$

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

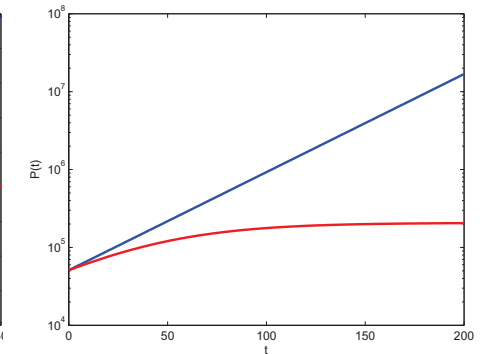
$$\begin{cases} \frac{dP(t)}{dt} = bP(t) - d(t)P(t) \\ P(0) = 50976 \end{cases}$$

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

8



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



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

9

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** $y(t)$.

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

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

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

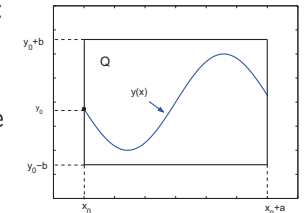
Esempio: $f(t, y) = Ky \rightarrow f_y(t, y) = K \rightarrow$ è **limitata**
 $\rightarrow f$ è **lipschitziana**

10

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

$$Q := \{(t, y) \in \mathbb{R}^2 \mid t \in [t_0, t_0 + a]; y \in [y_0 - b, y_0 + b]\}$$

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

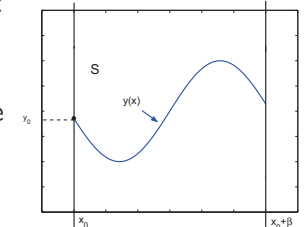


\Rightarrow esiste un'**unica** soluzione $y(t) \in C^1[t_0, t_0 + \beta]$ del **problema di Cauchy** in $I = [t_0, t_0 + \beta]$ con $\beta = \min \left[a, \frac{b}{M} \right]$, $M = \max_Q |f(t, y)|$.

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

$$S := \{(t, y) \in \mathbb{R}^2 \mid t \in I = [t_0, t_0 + \beta]; y \in \mathbb{R}\}$$

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



\Rightarrow esiste un'**unica** soluzione $y(t) \in C^1[t_0, t_0 + \beta]$ del **problema di Cauchy** in $I = [t_0, t_0 + \beta]$.

11

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**

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \quad \text{condizione iniziale}$$

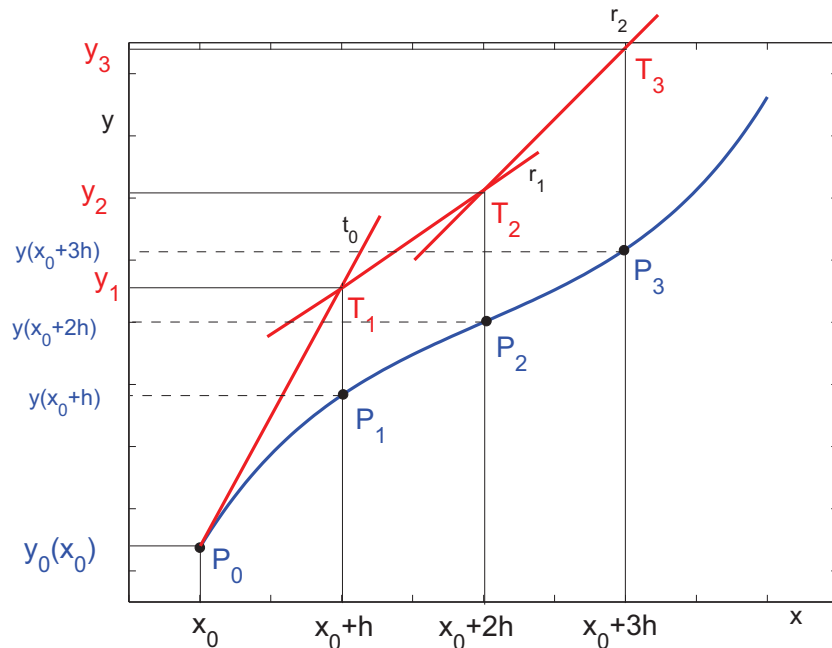
è **ben posto** se, dette $y(t; y_0)$ e $y(t; y_0 + \delta)$ le soluzioni con **condizioni iniziali** $y(t_0) = y_0$ e $y(t_0) = y_0 + \delta$ rispettivamente, si ha

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

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

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

12



Metodo di Eulero

Problema di Cauchy: $\begin{cases} y'(t) = f(t, y(t)) & t \in [t_0, t_0 + \beta] \\ y(t_0) = y_0 & \text{condizione iniziale} \end{cases}$

Discretizzazione di I : $t_i = t_0 + ih \quad i = 0, \dots, n \quad h = \frac{\beta}{n}$

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

Sviluppo in **serie di Taylor**:

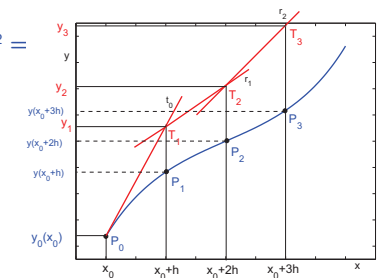
$$\begin{aligned} y(t_1) &= y(t_0 + h) = y(t_0) + y'(t_0)h + \frac{1}{2}y''(\tau_1)h^2 = \\ &= y(t_0) + f(t_0, y(t_0))h + \frac{1}{2}y''(\tau_1)h^2 \\ &\quad \tau_1 \in [t_0, t_1] \end{aligned}$$

Soluzione **approssimata**:

$$y_1 = y_0 + hf(t_0, y_0)$$

Errore globale di troncamento:

$$e_1(t_1) = y(t_1) - y_1 = \overline{P_1 T_1}$$



13

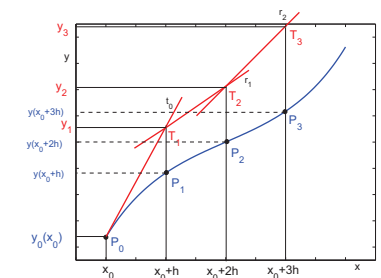
Algoritmo del metodo di Eulero

Algoritmo:

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

Errore globale di troncamento:

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



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

- l'**errore locale di troncamento**, dovuto al fatto che la soluzione esatta $y(t)$ viene **approssimata localmente** con una **retta**:

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

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

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

15

Convergenza del metodo di Eulero

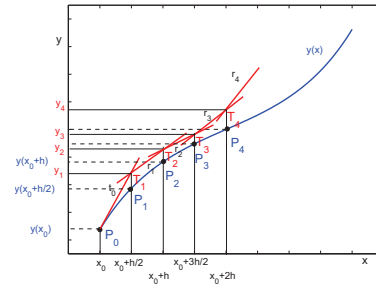
Definizione. Un **metodo numerico** per la soluzione approssimata di un'equazione differenziale è **convergente** se

$$\lim_{h \rightarrow 0} \max_{1 \leq i \leq n} |e_i| = 0 \Leftrightarrow \lim_{i \rightarrow \infty} y_i = y(\bar{t}) \quad \text{con } \bar{t} = t_0 + ih \text{ fissato}$$

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

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

⇒ Il metodo di Eulero è **convergente**



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

• **Consistente:**

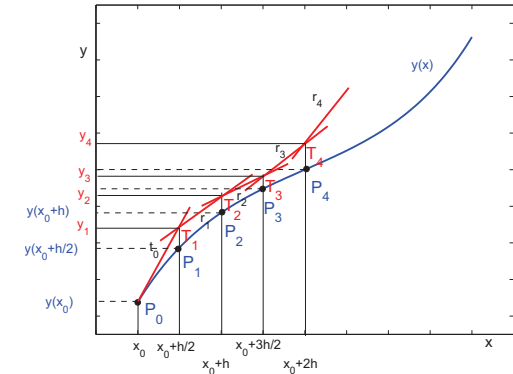
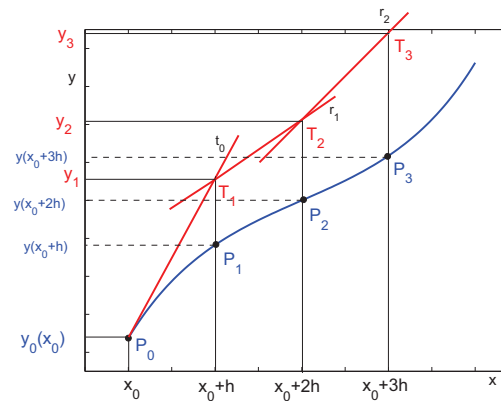
$$\lim_{h \rightarrow 0} \frac{R(t, y; h; f)}{h} = 0$$

+

• **Stabile:**

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

16



17

Metodi di ordine superiore al primo

Sviluppo in serie di Taylor di **ordine** m ($y \in C^{m+1}[t_0, t_n]$)

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

Schema di Taylor di ordine m

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

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

Errore locale di troncamento: $R(t_i, y(t_i); h; f) = \frac{h^{m+1}}{(m+1)!} y^{(m+1)}(\eta_i)$

Le derivate della f possono essere calcolate ricorsivamente da

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

18

Metodi di Runge-Kutta

Metodi di **ordine superiore** al primo si possono ottenere muovendosi lungo una **retta** che ha come coefficiente angolare una **combinazione lineare** di valori assunti da $f(t, y)$ in punti opportuni dell'intervallo $[t_i, t_i + h]$.

Relazione esatta

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

$$\text{con } \begin{cases} \Phi(t, y; h; f) = \sum_{l=1}^r a_l k_l(t, y) \\ k_1(t, y) = f(t, y) \\ k_l(t, y) = f\left(t + \lambda_l h, y + h \sum_{j=1}^{l-1} b_{lj} k_j(t, y)\right) \quad l = 2, \dots, r \end{cases}$$

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

19

Metodi di Runge-Kutta del secondo ordine

Relazione esatta

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

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

Sviluppo in serie di Taylor

$$\begin{aligned} y(t+h) &= y(t) + y'(t)h + \frac{1}{2}y''(t)h^2 + O(h^3) = \\ &= y(t) + f(t, y)h + \frac{1}{2}(f_t(t, y) + f_y(t, y)f(t, y))h^2 + O(h^3) \end{aligned}$$

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

20

$$\begin{aligned} R(t, y; h; f) &= y(t+h) - y(t) - h\Phi(t, y; h; f) = \\ &= hf(t, y) + \frac{h^2}{2}(f_t(t, y) + f_y(t, y)f(t, y)) + O(h^3) - \\ &\quad - h(a_1 f(t, y) + \\ &\quad \quad a_2(f(t, y) + f_t(t, y)\lambda h + f_y(t, y)\lambda h f(t, y) + O(h^2))) \end{aligned}$$

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

↓

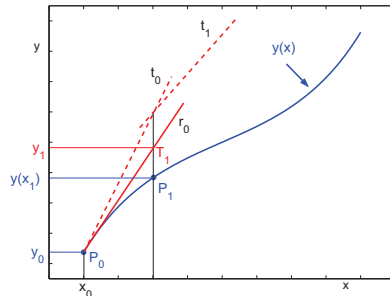
$$\begin{cases} 1 - a_1 - a_2 = 0 \\ \frac{1}{2} - a_2 \lambda = 0 \end{cases} \Rightarrow \text{infinita soluzioni}$$

21

Metodo di Heun

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

$$\begin{cases} y_{i+1} = y_i + \frac{h}{2}(f(t_i, y_i) + f(t_i + h, y_i + hf(t_i, y_i))) & i = 0, 1, \dots, n \\ y_0 = y(t_0) \end{cases}$$



Errore locale di truncamento:

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

22

Metodo di Runge-Kutta classico

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6}(k_1(t_i, y_i) + 2k_2(t_i, y_i) + 2k_3(t_i, y_i) + k_4(t_i, y_i)) & i = 0, 1, \dots, n \\ y_0 = y(t_0) \end{cases}$$

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

Errore locale di truncamento:

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

23

Esempio

Problema di Cauchy:

$$\begin{cases} y'(t) = y(t) - t & t \in [0, 2] \\ y(0) = 2 \end{cases} \rightarrow \text{Soluzione esatta: } y(t) = e^t + t + 1$$

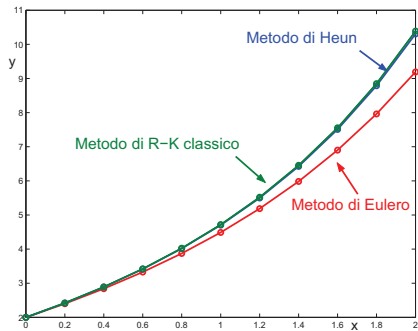


Grafico della soluzione

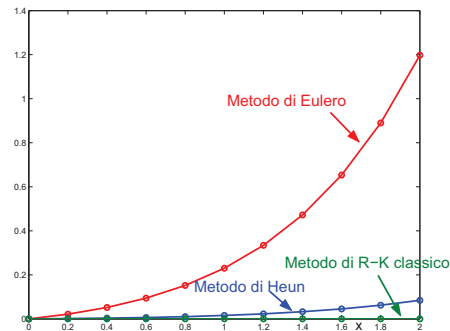


Grafico dell'errore

24

Script MATLAB

```
% Soluzione di un'equazione differenziale del primo ordine
% con il metodo di Eulero
%
% Input
%
clear
t0 = input('t0: ');
y0 = input('y0: ');
h = input('passo: ');
n = input('numero passi: ');
fun = input('funzione: ');
f = inline(fun,'t','y');
%
% Algoritmo
%
ti(1) = t0;
yi(1) = y0;
for i = 2:n
    yi(i) = yi(i-1) + h*f(ti(i-1),yi(i-1));
    ti(i) = ti(i-1)+h;
end
% Grafico
%
figure(1)
plot(ti,yi)
```

25

Programma FORTRAN

```
program eulero
*
* Soluzioni di equazioni differenziali del primo ordine
* con il metodo di Eulero esplicito
*
* Funzioni:
* - f(x,y): termine noto dell'equazione differenziale
*
* Input:
* - x0, y0: condizione iniziale
* - h: passo di discretizzazione
* - n: numero di passi
*
* Output:
* - xi: nodo i-esimo
* - yi: approssimazione al nodo xi
*
    implicit none
    real    t0, y0, h, ti, yi
    real    f
    integer n, i
*
*
    f(t,y) = (y-t)
```

26

```
    write (*,*) 'Inserisci t0, y0, h, n'
    read (*,*) t0, y0, h, n
*
    ti = t0
    yi = y0
    write (10,*) 't0=',t0,' y0=',y0,' h=',h,' n=',n
    do i = 1, n
        yi = yi + h*f(ti,yi)
        ti = ti + h
        write (*,*) i, ti, yi, exp(ti)+ti+1.
        write (10,*) i, ti, yi, exp(ti)+ti+1.
    enddo
*
    stop
end
```

27

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:

$$\begin{cases} y_{i+1} = y_i + h \underbrace{\Phi(t_i, y_i; h; f)}_{\text{Funzione incremento}} & i = 0, 1, \dots, n \\ y_0 = y(t_0) \end{cases}$$

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

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

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

28

Convergenza

Errore globale di troncamento:

$$e_i = y(t_i) - y_i$$

Convergenza:

$$\lim_{h \rightarrow 0} \max_{0 \leq i \leq n} |e_i| = 0 \Leftrightarrow \lim_{i \rightarrow \infty} y_i = y(\bar{t}) \quad \bar{t} = t_0 + ih$$

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** $\rightarrow 0$ per $h \rightarrow 0$

Stabilità: l'**accumularsi** degli errori locali di troncamento si mantiene **limitato** per $h \rightarrow 0$ e t fissato

29

Errore locale di troncamento

L'**errore locale di troncamento** rappresenta l'**errore** dovuto al fatto di aver approssimato **localmente** la soluzione esatta con una retta.

Errore locale (unitario) di troncamento:

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

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

$$\tau(t, y(t); h; f) = O(h^p)$$

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

30

Consistenza dei metodi one-step espliciti

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

$$\begin{aligned} \lim_{h \rightarrow 0} \tau(t, y; h; f) &= \lim_{h \rightarrow 0} \left[\frac{y(t+h) - y(t)}{h} - \Phi(t, y(t); h; f) \right] \\ &= y'(t) - \lim_{h \rightarrow 0} \Phi(t, y(t); h; f) = 0 \end{aligned}$$

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

$$\lim_{h \rightarrow 0} \Phi(t, y(t); h; f) = y'(t) = f(t, y(t))$$

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

31

Stabilità dei metodi one-step espliciti

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

$$e_{i+1} = y(t_{i+1}) - y_{i+1} = \left(y(t_i) + h\Phi(t_i, y(t_i); h; f) + h\tau(t_i, y(t_i); h; f) \right) - \left(y_i + h\Phi(t_i, y_i; h; f) \right)$$

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

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

Esempio: per il **metodo di Eulero** $|\tau| \leq \frac{M}{2}h$, dove M è il **massimo** di $|y''(t)|$ nell'intervallo di integrazione $\Rightarrow C_1 = M/2$

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

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

32

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

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

$$\Rightarrow \begin{cases} |e_{i+1}| \leq |e_i|(1 + hL) + C_p h^{p+1} & i = 0, 1, \dots \\ e_0 = y(t_0) - y_0 = 0 \end{cases}$$

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

$$\begin{cases} \zeta_{i+1} = \zeta_i(1 + hL) + C_p h^{p+1} & i = 0, 1, \dots \\ \zeta_0 = 0 \end{cases} \Rightarrow \zeta_i = C_p h^p \frac{(1 + hL)^i - 1}{L}$$

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

$$|e_i| \leq C_p h^p \frac{e^{L(t_i - t_0)} - 1}{L}$$

L : costante di Lipschitz di $f(t, y)$

C_p : costante dipendente dal metodo one-step.

33

Convergenza dei metodi one-step espliciti

Teorema. Sia $\Phi(t, y(t); h; f) \in C^0(D)$, $D = S \times [t_0, t_0 + \beta]$, $0 < h \leq \beta$, e **lipshitziana** 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

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

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

34

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

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

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

$$|e_i| \leq \frac{e^{L(t_i - t_0)} - 1}{L} \left(C_p h^p + \frac{\eta}{h} \right)$$

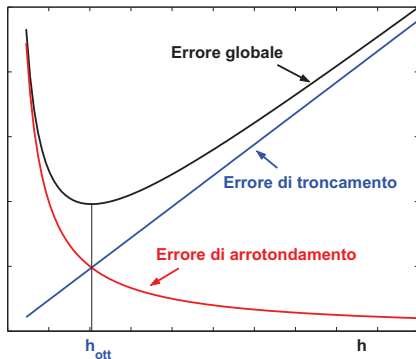
In particolare, per il **metodo di Eulero** si ha

$$|e_i| \leq \frac{e^{L(t_i - t_0)} - 1}{L} \left(\frac{M}{2} h + \frac{\eta}{h} \right) \quad M = \max_{t \in [t_0, t_0 + \beta]} |y''(t)|$$

35

Errore globale:

$$|e_i| \leq \frac{e^{L(t_i-t_0)} - 1}{L} \left(\frac{M}{2} h + \frac{\eta}{h} \right) \quad M = \max_{t \in [t_0, t_0+\beta]} |y''(t)|$$



In corrispondenza del **valore otti-**

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

36

Esempio

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

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

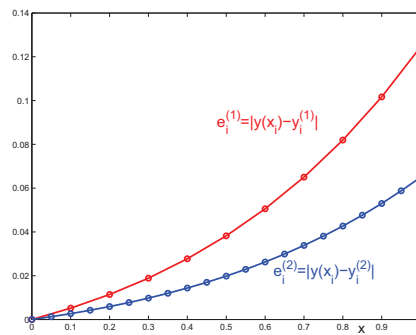
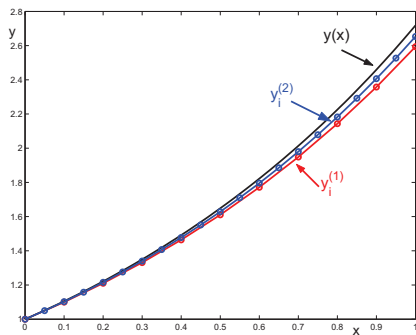
Metodo di Eulero:

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

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

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

37



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

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

38

Esempio

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

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

Metodo di Eulero:

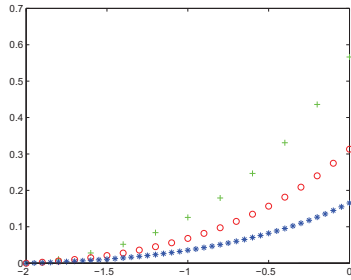
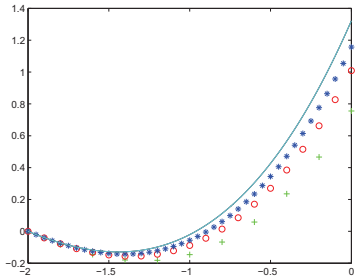
$$\begin{cases} y_{i+1} = y_i + hf(x_i, y_i) = y_i + h(y_i + \cos(x_i)) & i = 1, 2, \dots, n \\ y_0 = 0 \end{cases}$$

1) $x_i^{(1)} = ih_1 \quad i = 0, 1, \dots, 10 \quad h_1 = 0.2$

2) $x_i^{(2)} = ih_2 \quad i = 0, 1, \dots, 20 \quad h_2 = 0.1$

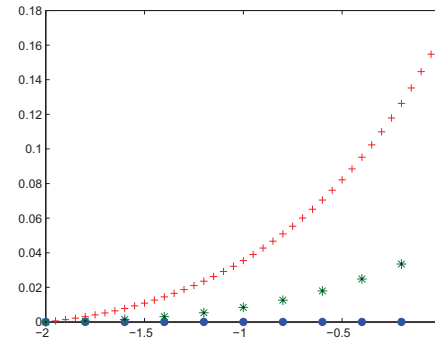
3) $x_i^{(3)} = ih_3 \quad i = 0, 1, \dots, 40 \quad h_3 = 0.05$

39



Nota. $M = \max_{x \in [-2,0]} |y''(x)| \simeq 1.32$

$$\text{Se } \eta = 0.5 \cdot 10^{-14} \Rightarrow h_{\text{ott}} = \sqrt{\frac{2\eta}{M}} \simeq 0.87 \cdot 10^{-7}$$



Metodo di Eulero $h = 0.05$:
 $\max_i |e_i| \approx 0.16$

Metodo di Heun $h = 0.2$:
 $\max_i |e_i| \approx 0.44 \cdot 10^{-1}$

Metodo di Runge-Kutta $h = 0.2$:
 $\max_i |e_i| \approx 0.71 \cdot 10^{-4}$

Metodi Predictor-Corrector

I **metodi predictor-corrector** si basano sull'utilizzo di un **metodo implicito** costruito integrando **localmente** l'equazione differenziale $y'(t) = f(t, y(t))$:

$$\Rightarrow y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} y'(t) dt = \underbrace{\int_{t_i}^{t_{i+1}} f(t, y(t)) dt}_{\text{formula di quadratura}}$$

Metodo (implicito) di Crank-Nicolson:

si ottiene utilizzando la **formula dei trapezi**

$$\int_{t_i}^{t_{i+1}} f(t, y(t)) dt \approx \frac{h}{2} (f(t_{i+1}, y(t_{i+1})) + f(t_i, y(t_i)))$$

$$\Rightarrow y_{i+1} = y_i + \frac{h}{2} (f(t_{i+1}, y_{i+1}) + f(t_i, y_i))$$

Metodo di Eulero Modificato ($p = 2$)

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

$$\begin{cases} \text{Predictor: } & y_{i+1} = y_i + h f(t_i, y_i) & 0 \leq i \\ \text{Corrector: } & y_{i+1}^{(r)} = y_i + \frac{h}{2} (f(t_{i+1}, y_{i+1}^{(r-1)}) + f(t_i, y_i)) & 1 \leq r \leq N \\ & y_{i+1} = y_{i+1}^{(N)} \end{cases}$$

Convergenza:

Il metodo **converge** se $h < \frac{2}{L}$ (L è la costante di Lipschitz di f).

Nota. Se L è **grande (problemi stiff)** la limitazione su h può essere troppo **restrittiva** \Rightarrow metodi di tipo Newton

Sistemi di equazioni differenziali

Sistema di equazioni differenziali del primo ordine

$$\begin{cases} y_1'(t) = f_1(t, y_1(t), \dots, y_n(t)) \\ y_2'(t) = f_2(t, y_1(t), \dots, y_n(t)) \\ \dots \\ y_n'(t) = f_n(t, y_1(t), \dots, y_n(t)) \end{cases} \quad \begin{cases} y_1(t_0) = y_{10} \\ y_2(t_0) = y_{20} \\ \dots \\ y_n(t_0) = y_{n0} \end{cases} \quad \text{Condizioni iniziali}$$

In forma **vettoriale**:

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

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

$$\begin{cases} y'(t) = f(t, y(t), z(t)) \\ z'(t) = g(t, y(t), z(t)) \end{cases} \quad \begin{cases} y(t_0) = y_0 \\ z(t_0) = z_0 \end{cases} \quad \text{Condizioni iniziali}$$

Nota. Per i **sistemi di equazioni differenziali** del primo ordine valgono teoremi di **esistenza** e **unicità** della soluzione analoghi a quelli validi nel caso di un'unica equazione differenziale.

44

Soluzione numerica di sistemi di equazioni differenziali

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

Metodo di Eulero:

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

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

$$\begin{cases} y'(t) = f(t, y(t), z(t)) \\ z'(t) = g(t, y(t), z(t)) \\ y(t_0) = y_0 \\ z(t_0) = z_0 \end{cases} \rightarrow \begin{cases} y_{i+1} = y_i + hf(t_i, y_i, z_i) \\ z_{i+1} = z_i + hg(t_i, y_i, z_i) \\ y_0 = y(t_0) \\ z_0 = z(t_0) \end{cases} \quad i = 0, 1, \dots, n$$

45

Metodo di Heun:

$$\begin{cases} Y_{i+1} = Y_i + \frac{h}{2}[F(t_i, Y_i) + F(t_i + h, Y_i + hF(t_i, Y_i))] \\ Y_0 = Y(t_0) \end{cases} \quad i = 0, 1, \dots, n$$

$$\text{dove} \quad \begin{cases} Y_0 = [y_1(t_0), y_2(t_0), \dots, y_n(t_0)]^T \\ Y_i = [y_{1i}, y_{2i}, \dots, y_{ni}]^T \\ F(t_i, Y_i) = [f_1(t_i, Y_i), \dots, f_n(t_i, Y_i)]^T \end{cases}$$

$$\text{Caso particolare: } n = 2 \quad \begin{cases} y'(t) = f(t, y(t), z(t)) \\ z'(t) = g(t, y(t), z(t)) \\ y(t_0) = y_0 \\ z(t_0) = z_0 \end{cases}$$

$$\begin{cases} y_{i+1} = y_i + \frac{h}{2}[f(t_i, y_i, z_i) + f(t_i + h, y_i + hf(t_i, y_i, z_i), z_i + hg(t_i, y_i, z_i))] \\ z_{i+1} = z_i + \frac{h}{2}[g(t_i, y_i, z_i) + g(t_i + h, y_i + hf(t_i, y_i, z_i), z_i + hg(t_i, y_i, z_i))] \\ y_0 = y(t_0) \\ z_0 = z(t_0) \end{cases} \quad i = 0, 1, \dots, n$$

46

Metodo di Runge-Kutta classico (IV ordine):

$$\begin{cases} Y_{i+1} = Y_i + \frac{h}{6}[K_1 + 2K_2 + 2K_3 + K_4] \\ Y_0 = Y(t_0) \end{cases} \quad (i = 0, 1, \dots, n) \quad \text{dove} \quad \begin{cases} K_1 = F(t_i, Y_i) \\ K_2 = F\left(t_i + \frac{h}{2}, Y_i + \frac{h}{2}K_1\right) \\ K_3 = F\left(t_i + \frac{h}{2}, Y_i + \frac{h}{2}K_2\right) \\ K_4 = F(t_i + h, Y_i + hK_3) \end{cases}$$

$$\text{Caso particolare: } n = 2 \quad \begin{cases} y'(t) = f(t, y(t), z(t)) \\ z'(t) = g(t, y(t), z(t)) \\ y(t_0) = y_0 \\ z(t_0) = z_0 \end{cases}$$

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6}[k_1 + 2k_2 + 2k_3 + k_4] \\ y_0 = y(t_0) \end{cases} \quad \begin{cases} z_{i+1} = z_i + \frac{h}{6}[q_1 + 2q_2 + 2q_3 + q_4] \\ z_0 = z(t_0) \end{cases} \quad (i = 0, 1, \dots, n)$$

$$\begin{cases} k_1 = f(t_i, y_i, z_i) & q_1 = g(t_i, y_i, z_i) \\ k_2 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_1, z_i + \frac{h}{2}q_1\right) & q_2 = g\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_1, z_i + \frac{h}{2}q_1\right) \\ k_3 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_2, z_i + \frac{h}{2}q_2\right) & q_3 = g\left(t_i + \frac{h}{2}, y_i + \frac{h}{2}k_2, z_i + \frac{h}{2}q_2\right) \\ k_4 = f(t_i + h, y_i + hk_3, z_i + hq_3) & q_4 = g(t_i + h, y_i + hk_3, z_i + hq_3) \end{cases}$$

47

Esempio 1: Modello preda-predatore

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

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

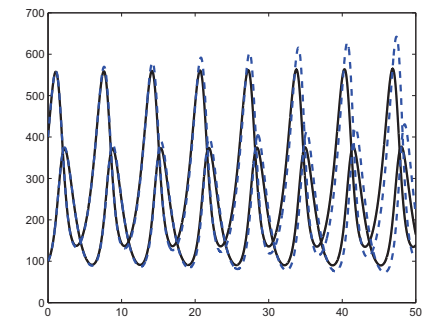
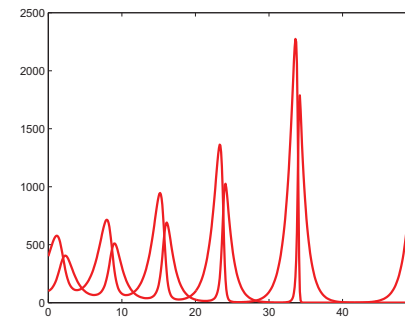
in cui $y_1(t)$ rappresenta il **numero di prede** e $y_2(t)$ rappresenta il **numero di predatori**.

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**.

48

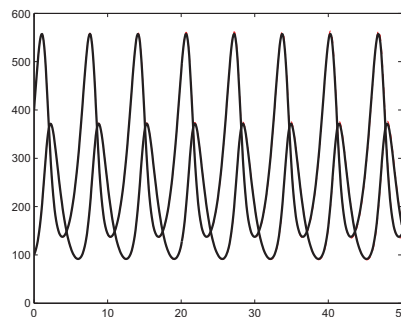
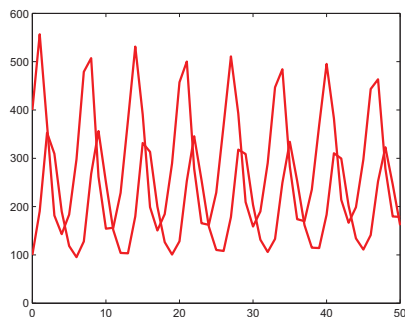
Metodo di Eulero



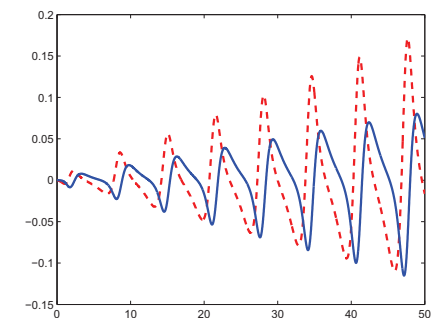
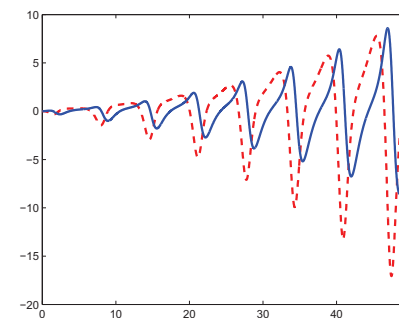
Passi: $h = 0.1$ (**grafico a sinistra**), $h = 0.001$ (grafico a destra, **linea continua**), $h = 0.01$ (grafico a destra, **linea tratteggiata**)

49

Metodo di Runge-Kutta classico



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



Differenza tra la soluzione ottenuta con il metodo di Runge-Kutta classico con passo $h = 0.01$ e quella ottenuta con i metodi di Eulero (sinistra) e di Crank-Nicholson con passi (destra) $h_1 = 0.01$ e $h_2 = 0.001$ ($y_1^{(RK)} - y_1^{(Eu,CN)}$, **linea tratteggiata**; $y_2^{(RK)} - y_2^{(Eu,CN)}$, **linea continua**).

Un modello preda-predatore per lo sviluppo sostenibile

$$\begin{cases} \dot{p} = (\alpha_1 + \alpha_3 f)p + \alpha_2 g p & t > t_0 \\ \dot{g} = \beta_1 g + \beta_2 p g \\ p(t_0) = p_0 & g(t_0) = g_0 \end{cases} \quad \begin{array}{l} p(t) : \text{popolazione (predatore)} \\ g(t) : \text{PIL (preda)} \end{array}$$

- dove
- α_1 e β_1 rappresentano il **tasso di crescita** della popolazione e del PIL, rispettivamente
 - α_2 e β_2 sono i termini di **interazione tra popolazione e PIL** e rallentano la crescita
 - $\alpha_3 f$ **modula la crescita** di p .
Spesso si assume $f = k_1 g/p$

50

Esempio - 2

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

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

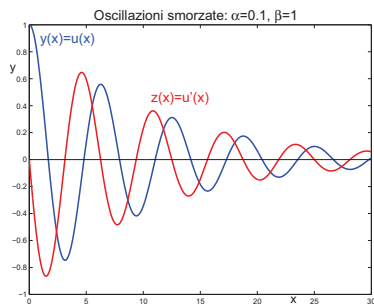
↓

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

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

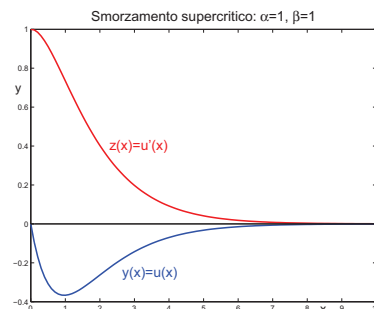
Per risolvere il sistema differenziale ottenuto, si possono applicare i metodi già visti per la soluzione dei sistemi (ad esempio, il metodo di Runge-Kutta classico).

51



Dati del problema

```
x0: 0
y0: 1
z0: 0
passo: 0.2
numero passi: 150
funzione f(x,y,z): 'z'
funzione g(x,y,z): '-2*0.1*z-y'
```



Dati del problema

```
x0: 0
y0: 1
z0: 0
passo: 0.2
numero passi: 150
funzione f(x,y,z): 'z'
funzione g(x,y,z): '-2*z-y'
```

52

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');
```

53

```

%
% 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')

```

54

Programma FORTRAN

```

program rungekutta_n2
*
* Soluzioni di un sistema di due equazioni differenziali del primo ordine
* con il metodo di Runge-Kutta del IV ordine
*
* Input:
* - x0, y0, z0: condizioni iniziali
* - h: passo di integrazione
* - n: numero di passi
* Output:
* - xi: nodi
* - yi: approssimazione
* External functions:
* - f(x,y,z)
* - g(x,y,z)
*
    implicit none
    real x0, y0, z0
    real h, xi(0:2000), yi(0:2000), zi(0:2000)
    real k1, k2, k3, k4
    real t1, t2, t3, t4
    real f, g, x, y, z
    integer n, i

```

55

```

*
* Lettura dati di input
*
    write (*,*) 'Inserisci x0, y0, z0, h, n'
    read (*,*) x0, y0, z0, h, n
*
* Metodo di Runge-Kutta del IV ordine
*
    xi(0) = x0
    yi(0) = y0
    zi(0) = z0
    write (*,*) 0, xi(0), yi(0), zi(0)

    do i = 1, 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)

```

56

```

        yi(i) = y + h*(k1+2.*k2+2.*k3+k4)/6.
        zi(i) = z + h*(t1+2.*t2+2.*t3+t4)/6.
        xi(i) = x0 + i*h
        write (*,*) i, xi(i), yi(i), zi(i)
    enddo
*
* File di output per GNUPLOT
*
    open(10,file='grafrk4.dat')
    write (10,120) (xi(i),yi(i),zi(i),i=0,n)
    close(10)
120 format (3(2x,e12.6))
*
* Fine
*
    stop
end

```

57

External Functions (Fortran)

```
real function f(x,y,z)
implicit none
real x,y,z
f = z
return
end
```

```
real function g(x,y,z)
implicit none
real x,y,z
g = -2*0.1*z-y
return
end
```

Esercizio.

Utilizzare il programma per approssimare la soluzione del problema del **pendolo non lineare**.

Confrontare graficamente la soluzione approssimata ottenuta con quella del problema lineare.

Riferimenti bibliografici

L. Gori, *Calcolo Numerico*: Cap. 9 §§ 9.1-9.6, 9.8 (cenni), 9.14 (cenni)

L. Gori, M.L. Lo Cascio, *Esercizi di Calcolo Numerico*: 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 7.76, 7.80, 7.85