



# Modellistica e Simulazione

(parte 5)

Prof. Salvatore Pirozzi

Email: [salvatore.pirozzi@unicampania.it](mailto:salvatore.pirozzi@unicampania.it)

# MATLAB e SIMULINK

A screenshot of a Windows desktop environment. At the top, a browser window shows a Google search for "mathworks vanvitelli". Below the search results, a link to the University of Campania's MATLAB page is visible. The desktop taskbar at the bottom shows icons for various applications, including MATLAB and Microsoft Office.

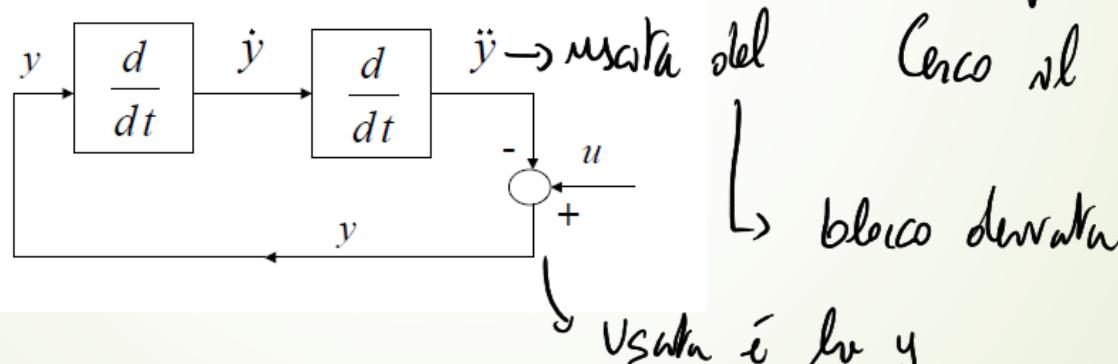
The main content area displays the official website for the University of Campania Luigi Vanvitelli. The header features the university's logo and name. A prominent banner highlights the use of MATLAB and Simulink, stating "Accesso a MATLAB per tutti presso Università degli Studi della Campania Luigi Vanvitelli". Below the banner, a large image shows students working with a robotic arm. A cookie consent message is visible at the bottom of the page. The system tray in the bottom right corner shows standard icons for date, time, and battery level.

# Schema per la simulazione numerica

Supponiamo di voler implementare graficamente uno schema per la simulazione dell'equazione differenziale  $\ddot{y} + y = u$

$$\ddot{y} + y = u$$

Si potrebbe ipotizzare di conoscere  $y$  e ricavare le sue derivate, al fine di costruire lo schema seguente



Come funziona con il simulink?  
Cerco nel blocco osservata

blocco derivata  
Usata i ho  $y$

## Controindicazioni

- i segnali reali sono affetti da rumore che l'operazione di derivata amplifica
- dal punto di vista numerico è un'operazione poco stabile

Trovai una versione  
approssimata della  $y$ ,  
con oscilloscopio vedo.

Derivata è poco stabile perché fatta per approssimazione, perché difficile da implementare.  
faccio rapporto incrementale con 16 passi. Funziona con segnale pulito e senza rumore.

Punto da y derivata e non da y.

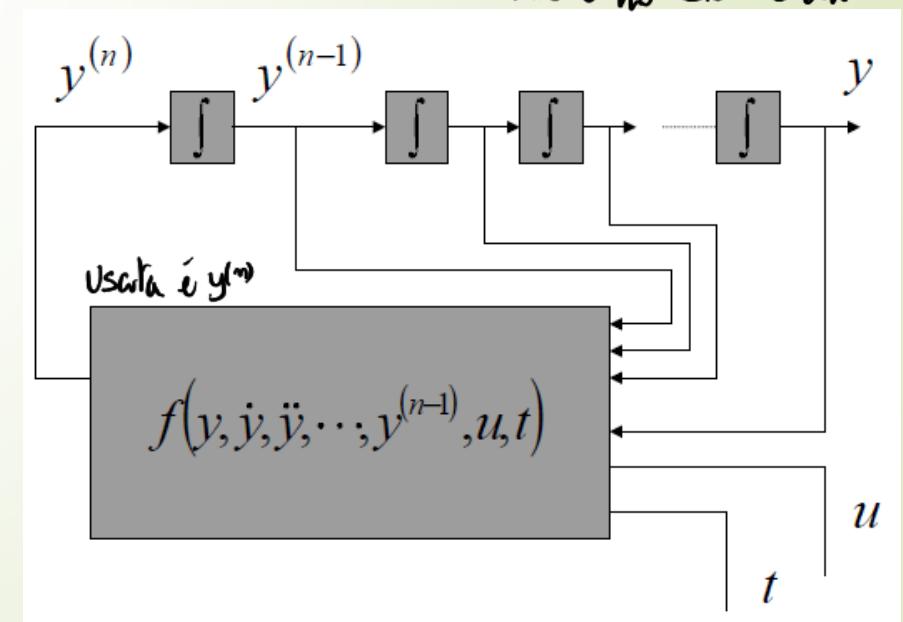
# Schema di Lord Kelvin (1876)

Si basa sull'utilizzo dell'integratore come elemento fondamentale. Isolando nell'equazione differenziale la derivata di grado massimo, possiamo riscrivere l'equazione differenziale

$$\frac{d^n y}{dt^n} = f\left(\frac{d^{n-1}y}{dt^{n-1}}, \dots, \frac{dy}{dt}, y, u, t\right)$$

Ingressi e dipendenze del tempo  
Risolvendo schema e ho soluzioni

- Supponiamo che il primo termine sia noto in qualche modo. A partire da esso integrandolo n volte è possibile ricostruire tutti i termini dalla derivata  $(n-1)$  fino al termine non derivato
- Ricostruiamo poi il primo termine combinando mediante una funzione « $f$ » i termini ricostruiti utilizzando gli integratori



# Schema di Lord Kelvin

Per l'equazione differenziale considerata all'inizio, essendo di secondo ordine possiamo utilizzare 2 integratori in cascata.

In generale per un'equazione differenziale di ordine «n» utilizziamo «n» integratori in cascata

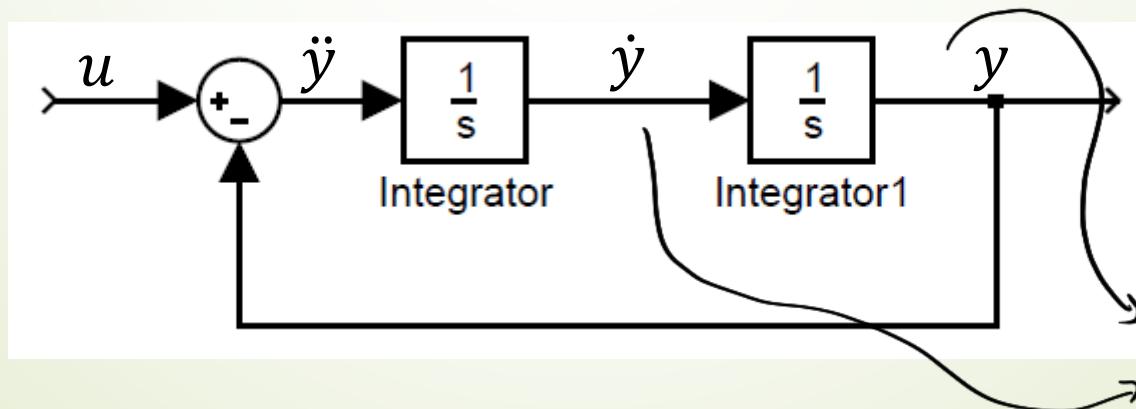
$$\ddot{y} + y = u$$

E le condiz. iniziali?

Nell'integrale. La metto nel blocco.

Ne serve una su  $y$  e su  $\dot{y}$ .

→ Cond. iniz. su  $y$ .  
→ Cond. iniz. su  $\dot{y}$ .



# Schema di Lord Kelvin

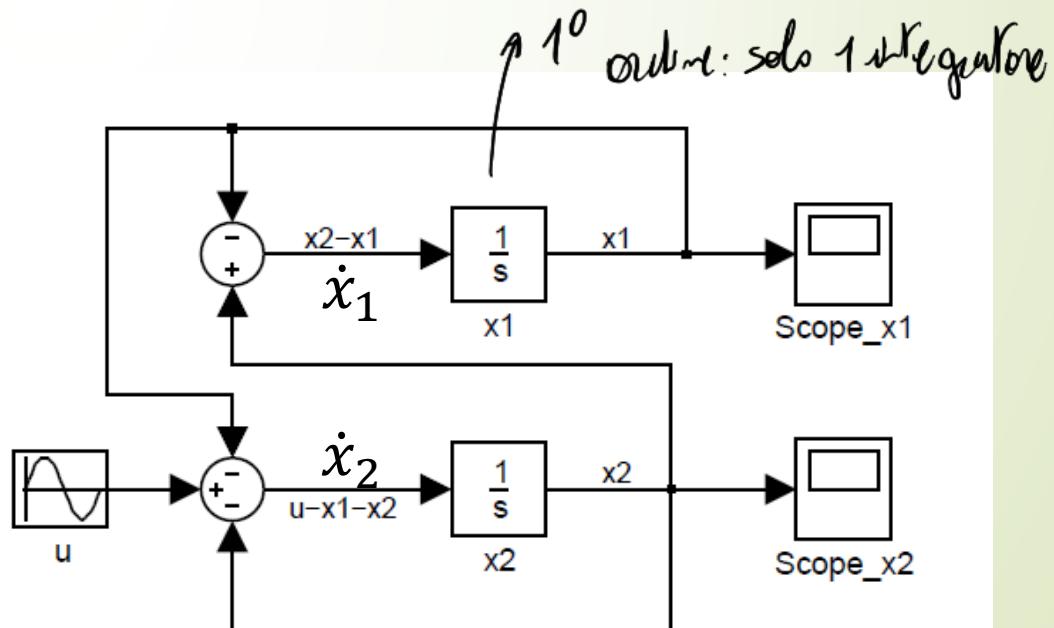
Per un sistema di «n» equazioni differenziali del primo ordine abbiamo sempre «n» integratori le cui uscite saranno le variabili non derivate che opportunamente combinate consentiranno di ricostruire le variabili derivate.

Esempio con due equazioni, un ingresso sinusoidale e due oscilloscopi per le variabili di stato

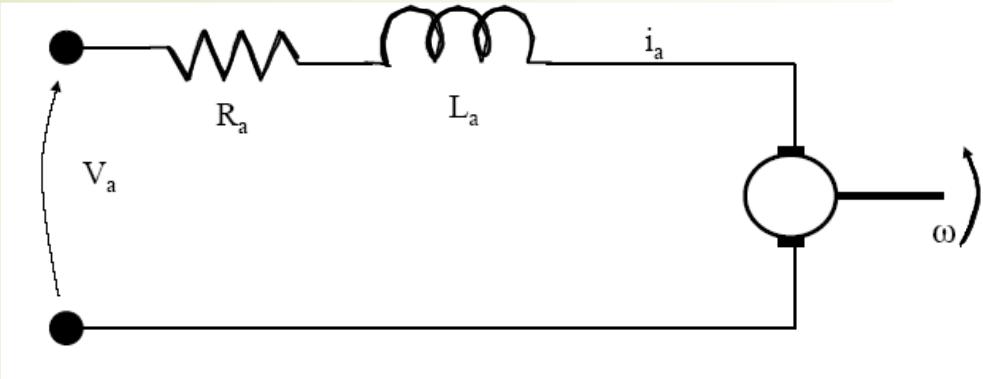
$$\dot{x}_1 = -x_1 + x_2$$

$$\dot{x}_2 = -x_1 - x_2 + u$$

Punti sempre da un regolatore.



# Esempio: motore in c.c.



Nelle ipotesi di

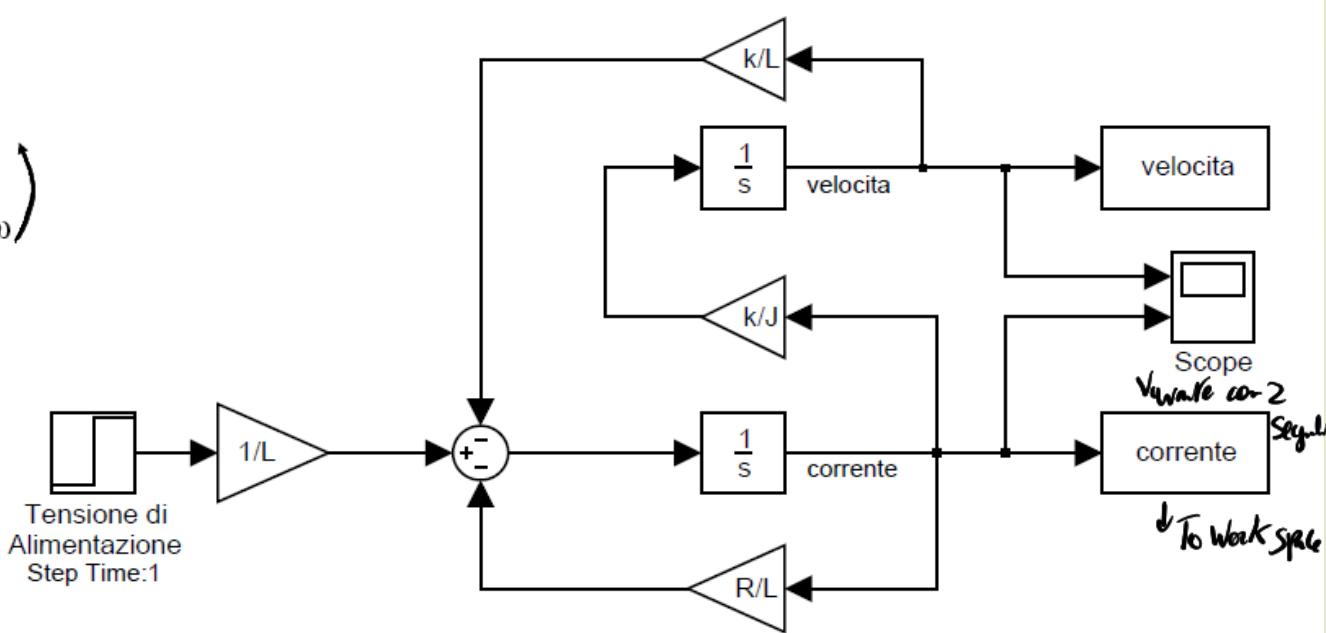
- ❖ assenza di attrito all'asse motore
- ❖ assenza di coppia di carico

le equazioni che governano il funzionamento del sistema sono

$$J \frac{d}{dt}(\omega) = ki$$

$$L \frac{d}{dt}(i) = -k\omega - Ri + V$$

$J$	6	$\text{kg m}^2$
$L$	1	$\mu\text{H}$
$R$	0.3	$\Omega$
$k$	0.5	



sistema è stiff se ci sono sia variabili stiff che smooth.

Proviamo algoritmo **ode45** e poi **ode15s** per sistemi stiff → Segnale stiff: dmo.es:onda quadrata  
Ci sono cambiam. repentinai

- Ritorno Anay nel workspace

- Decimation mi indica

NOTA: Corrente che fa MMN dipende dall'approssimaz. del solver.

Per dare il prezzo all'utente del quadtree da OAS prende un passo piccolo e se lo porta appresso.

SISTEMA STIFF: corrente dipende da velocità elettrica  
mentre velocità angolare non è stiff.

# Definizione del problema

Data l'equazione differenziale ordinaria del I ordine

$$\dot{x} = f(x, t); \quad t \in [T_0, T_f]; \quad x(T_0) = x_0$$

Sotto le opportune ipotesi l'unica soluzione esatta è  $x(t)$

Integrare numericamente l'equazione vuol dire

- individuare  $n$  istanti di tempo nell'intervallo di integrazione

$$T_0 = t_1 < t_2 < \dots < t_n = T_f$$

- e determinare  $n$  valori  $x_i$  che approssimino  $x(t_i)$  → Queste  $x(t_i)$  sono valori esatti, ma non approssimano con  $x_i$

Anziché risolvere il problema originale si risolve il problema approssimato

Non c'è  $x_{t_{i+1}}$ ?  $\rightarrow x(t_{i+1}) = x(t_i) + h_i \Delta(t_i, x, h, f)$  Termino generico che introduce approssimaz.

- che è facilmente risolvibile una volta definita  $\Delta$  e inizializzando  $x(t_1) = x(T_0) = x_0$  dell'eqaz. differenz.
- $h_i = t_{i+1} - t_i$  è il passo di integrazione

- se  $h_i = h = \text{cost.}$  → metodo di integrazione a passo fisso
- altrimenti → metodo di integrazione a passo variabile

Ci sono tutti a passo variabile su molti.

- i diversi metodi si differenziano per la scelta della funzione  $\Delta$

Ogni algoritmo di integraz. ha  $\Delta$ .

$x(t_0)$  mi serve per partire

$h$  indica la base dei rettangoli che approssimano area di derivata funzione

A passo variabile  $\Rightarrow$  combinaz. di alq. a passo fiss.

# Metodo di Eulero

funz. 1

Il metodo più semplice è a passo fisso ed utilizzando l'approssimazione della derivata con il rapporto incrementale: Metodo di Eulero

$$\dot{x} = f(x, t);$$



$$\frac{x_{i+1} - x_i}{h} \approx f(t_i, x_i)$$

$\dot{x}$  diretta rapporto tra.



$$x_{i+1} = x_i + h f(t_i, x_i)$$

↑ base ↑ altezza

$$h = t_{i+1} - t_i$$

all'ultima  $t_i$  ca cunpone  $x_i$ .

Indicando con  $x_i$  la soluzione approssimata  $x_i \approx x(t_i)$  essa è facilmente calcolabile a partire dalla condizione iniziale.

► Errore locale di troncamento (tra sol. esatta e approx)

$$e_i = x(t_i) - x_i$$

oggi trascurabile

► Errore di arrotondamento (tra sol. approx e sol. numerica con precisione della macchina)

$$e_a = x_i - \tilde{x}_i$$

► Errore globale è la sovrapposizione dei due precedenti

$$e_g = |x(t_i) - \tilde{x}_i| = |x(t_i) - x_i + x_i - \tilde{x}_i| \leq |x(t_i) - x_i| + |x_i - \tilde{x}_i|$$

Somma dei due errori

Nel prosieguo trascureremo l'errore di arrotondamento e considereremo  $x_i$  la soluzione numerica

# Definizioni di convergenza

Definito il vettore contenente i campioni della soluzione esatta

$$\mathbf{X}_v = [x(t_1), \dots, x(t_n)]$$

Ed il vettore contenente i campioni della soluzione numerica

$$\mathbf{X} = [x_1, \dots, x_n]$$

Un metodo di integrazione si dice convergente se

$$\lim_{n \rightarrow \infty} \|\mathbf{X}_v - \mathbf{X}\| = 0$$

aumentano i campioni

- cioè se la soluzione numerica tende a quella vera al crescere del numero di punti in cui la si calcola

# Metodi a passo fisso

Come migliorare il metodo di integrazione?

- L'idea è quella di sviluppare la soluzione  $f$  in serie di Taylor ed utilizzarla come  $\Delta$

$$f(t, x) + \frac{h}{2} f'(t, x) + \cdots + \frac{h^{m-1}}{m!} f^{(m-1)}(t, x)$$

Non c'è  $(x-x_0)^m$ ?  
E per a cosa serve?

$$\Delta(t_i, x_i, h, f) = f(t_i, x_i) + \frac{h}{2!} f'(t_i, x_i) + \cdots + \frac{h^{m-1}}{m!} f^{m-1}(t_i, x_i)$$

- per  $m = 1$  si riottiene il metodo di Eulero!
- Per  $m > 1$  la difficoltà consiste nel non poter valutare le derivate di  $f$

Difficoltà è calcolare derivate.

# Metodi a passo fisso: Runge-Kutta

l'idea è quella di valutare  $f$  anche in punti interni a  $[t_i, t_{i+1}]$

- Integrando il problema di partenza limitatamente ad un intervallo di integrazione, otteniamo

$$\dot{x} = f(x, t); \quad t \in [T_0, T_f]; \quad x(T_0) = x_0$$



$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f(\tau, x(\tau)) d\tau$$

- Ed approssimando l'integrale con il metodo rettangolare (i.e., con l'area del rettangolo avente come base l'intervallo di integrazione e come altezza il valore che la funzione assume al centro dell'intervallo)

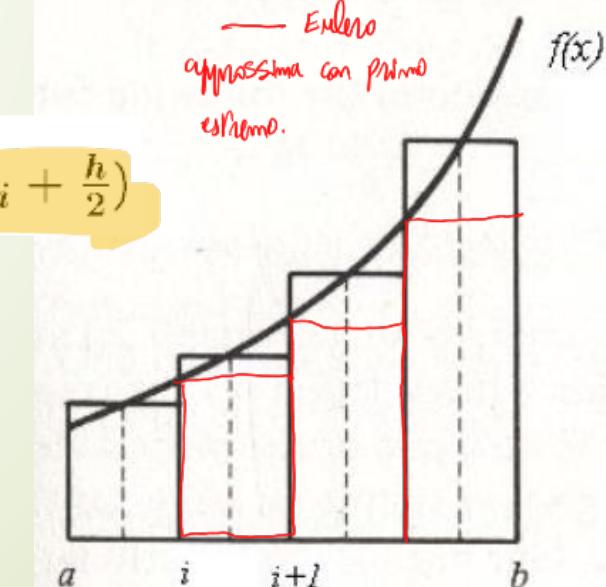
$$x(t_{i+1}) \approx x(t_i) + h f\left(t_i + \frac{h}{2}, x(t_i + \frac{h}{2})\right)$$

*conosco      non conosco*

Questa formula non è utilizzabile perché richiede un valore «futuro»  $x(t_i + \frac{h}{2})$

Si calcola con il metodo di Eulero tale quantità

$$x\left(t_i + \frac{h}{2}\right) \approx x_i + \frac{h}{2} f(t_i, x_i)$$



E si combina con la precedente ottenendo un metodo a due passi

# Metodi a passo fisso: Runge-Kutta

La combinazione consente di ottenere un metodo di integrazione a due stadi, che garantisce maggiore accuratezza (devo calcolare due valori di  $f$ , dove il secondo dipende dal primo)

• Equazioni disaccoppiate

$$\begin{aligned}k_1 &= f(t_i, x_i) \\k_2 &= f\left(t_i + \frac{h}{2}, x_i + h \frac{k_1}{2}\right) \\x_{i+1} &= x_i + hk_2\end{aligned}$$

Migliora di un po' accuratezza  
Si può anche accoppiare le equaz. e altre in sistema.

In generale, l'algoritmo Runge-Kutta può implementare fino ad  $m$  stadi (migliorando l'accuratezza, ma aumentando il tempo di calcolo)

si noti come il calcolo dei  $k_q$  può procedere in maniera ricorsiva a partire da  $k_1$

$$\begin{aligned}k_1 &= f(t_i, x_i) \\k_q &= f\left(t_i + ha_q, x_i + h \sum_{j=1}^{q-1} b_{qj} k_j\right), \quad q = 2, \dots, m \\x_{i+1} &= x_i + h \sum_{q=1}^m c_q k_q\end{aligned}$$

Appross. ricon-  
sivo. Migliora  
accuratezza

L'indice  $m$  rappresenta l'ordine del metodo ed indica quanti valori di  $f$  bisogna calcolare ad ogni passo, mentre il tipo è definito dai parametri

$$a_q, c_q, b_{qj}$$

Numero di stadi: numero dopo nome algoritmo.

# Metodi a passo fisso

metodo di Eulero modificato ( $m=2, q=2, j=1$ )

$$a_q = 1/2, c_q = 1, b_{qj} = 1/2$$

$$x_{i+1} = x_i + h f \left( t_i + \frac{1}{2}h, x_i + \frac{1}{2}h f(t_i, x_i) \right)$$

metodo di Heun ( $m=2, q=2, j=1$ )

$$a_q = 1, c_q = 1/2, b_{qj} = 1$$

$$x_{i+1} = x_i + \frac{h}{2} [f(t_i, x_i) + f(t_i + h, x_i + h f(t_i, x_i))]$$

metodo di Runge-Kutta del 4° ordine ( $m=4$ )

$$c_1 = 1/6, c_2 = 1/3, c_3 = 1/3, c_4 = 1/6$$

$$x_{i+1} = x_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$a_1 = a_2 = a_3 = a_4 = 1/2$$

$$k_1 = f(t_i, x_i); \quad k_2 = f \left( t_i + \frac{1}{2}h, x_i + \frac{1}{2}h k_1 \right);$$

$$b_{jq} = 1/2, \quad \forall j, q$$

$$k_3 = f \left( t_i + \frac{1}{2}h, x_i + \frac{1}{2}h k_2 \right); \quad k_4 = f \left( t_i + \frac{1}{2}h, x_i + \frac{1}{2}h k_3 \right);$$

le diverse scelte corrispondono a differenti metodi di "quadratura", cioè metodi numerici per il calcolo di un integrale definito (ad es. Heun → integrazione per trapezi)

# Metodi a passo fisso

Metodi impliciti → accoppiati o no

- nei metodi Runge-Kutta finora visti, i parametri  $k_q$  vengono calcolati con delle formule esplicite
- più in generale possono essere anche definiti implicitamente da un sistema di equazioni non lineari da risolvere ad ogni passo di integrazione

$$\begin{aligned} k_1 &= f(t_i, x_i) \\ k_q &= f \left( t_i + ha_q, x_i + h \sum_{j=1}^m b_{qj} k_j \right), \quad q = 2, \dots, m \\ x_{i+1} &= x_i + h \sum_{q=1}^m c_q k_q \end{aligned}$$

si noti come la sommatoria è estesa fino ad  $m$ , quindi non basata il calcolo ricorsivo a partire da  $k_1$

Sistema di equaz. accoppiate. Più lento nel calcolo e leggermente più preciso.

# Metodi a passo fisso multistep

= Quanti campioni precedenti  
uso per calcolare il valore di  $x_{i+1}$

## Metodi multistep

- i metodi visti finora, siano essi esplicativi o impliciti, calcolano  $x_{i+1}$  in funzione solo di  $x_i$  e non della soluzione a passi ancora precedenti, per questo sono detti metodi one step
- i metodi multistep invece calcolano la soluzione al passo  $i + 1$  usando i valori calcolati  $x_i, x_{i-1}, \dots, x_{i-r}$
- sorge subito il problema di come inizializzare l'algoritmo dal momento che è disponibile un'unica condizione iniziale  $x_0$ 
  - ❖ una soluzione è quella di partire con un metodo one step
    - Numero di operazioni da fare è maggiore. O mette a O i campioni precedenti o punto con un one step

# Metodi a passo fisso multistep

## ► Metodo Adams (funzioni polinomiali)

$$x_{i+1} = x_i + \sum_{j=0}^m \beta_j f(t_{i-j}, x_{i-j})$$

Combinazione di polinomi.

Sempre approssimabile con dei polinomi.

Funzione  $\Delta$ ; combinaz. di polinomi.

❖ è implicito se  $\beta_0 \neq 0$

## ► Metodi BDF (Backward Differentiation Formulas)

$$\sum_{k=1}^m \frac{1}{k} \nabla^k x_{i+1} - h f(t_{i+k}, x_{i+k})$$

$\nabla^k x_j = x_j - x_{j-k}$

ASK

- ❖ dove  $\nabla x_j = x_j - x_{j-1}$  è l'operatore di differenza all'indietro
- ❖ sono utilizzati per sistemi stiff

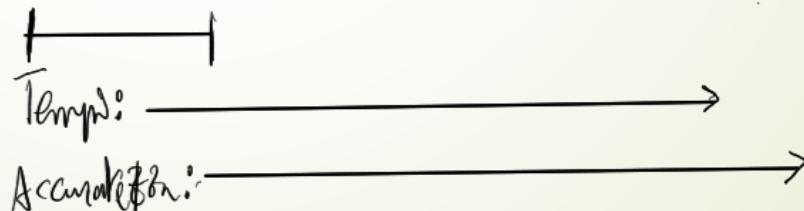
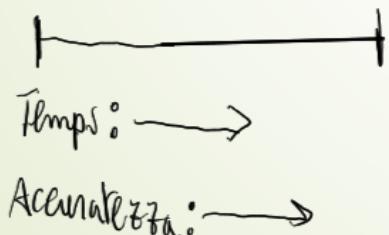
# Cenni sui metodi a passo variabile

## ► Motivazione

- ❖ la riduzione generalizzata del passo di integrazione per avere buona accuratezza comporta tempi di calcolo elevati!
- ❖ per sistemi stiff ciò è ancora più importante!

## ► L'idea di base

- ❖ si stima l'errore locale di troncamento con apposite tecniche
- ❖ se l'errore è più grande di una certa tolleranza si scarta la soluzione e si riduce il passo di integrazione
- ❖ altrimenti passa al calcolo del valore successivo con un passo di integrazione maggiore



# I metodi a passo variabile in Simulink

## ► ode45

- ❖ coppia di metodi Runge-Kutta esplicativi one step di ordine 4 e 5
- ❖ adatto ad un buon numero di problemi ma non stiff

## ► ode23

- ❖ coppia di metodi Runge-Kutta esplicativi one step di ordine 2 e 3
- ❖ è meno preciso del precedente ma più efficiente

## ► ode113

- ❖ combinazione di metodi multistep sia impliciti che esplicativi
- ❖ è adatto quando la funzione  $f(t,x)$  è molto complessa perché ne richiede la valutazione un numero di volte basso (ordine basso)

NON ACCOPPIATO

PASSO FISSO/variabile

→ SINGLE STEP/MULTISTEP

Combinano 2 metodi  
Runge a h e S step.  
Switch fra h e S a  
Secondo dell'esigenza

↪ ci vuole meno tempo.

Sistemi fortemente non lineari,

# I metodi a passo variabile in Simulink

- STIFF  $\uparrow$  da passo 1 a 5  
→ **ode15s** → Implanto: backward differentiation formulae
- ❖ metodo multistep basato sulle BDF
  - ❖ adatto ad un'ampia classe di problemi compresi quelli stiff
  - ❖ è il primo metodo da provare in alternativa a ode45 di default
- **ode23s** → Passo 2-3: numero di step
- ❖ metodo one step basato sulle BDF
  - ❖ è meno preciso del precedente ma più efficiente
- **ode23t e ode23tb**
- ❖ metodi basati sull'integrazione per trapezi
  - ❖ da usare per problemi moderatamente stiff quando ode15s risulta computazionalmente troppo oneroso

# I metodi a **passo fisso** in Simulink

- ▶ ode5
  - ❖ versione a passo fisso di ode45
- ▶ ode4
  - ❖ metodo di Runge-Kutta del 4° ordine
- ▶ ode3
  - ❖ versione a passo fisso di ode23
- ▶ ode2
  - ❖ metodo di Heun
- ▶ ode1
  - ❖ metodo di Eulero

*Quelli a passo variabile fanno al massimo  
di passi,*

# I parametri principali della simulazione

## Metodi a passo variabile

- ▶ **Max step size**
  - ❖ è il valore massimo del passo di integrazione
- ▶ **Min step size**
  - ❖ è il valore minimo del passo di integrazione
- ▶ **Initial step size**
  - ❖ con “auto” è calcolato in base alla derivata dello stato iniziale
- ▶ **Relative tolerance** *Capsice se allarga o restringe il passo*
  - ❖ è il valore di precisione desiderata per la soluzione
- ▶ **Absolute tolerance**
  - ❖ indica la soglia al di sotto della quale la soluzione è considerata nulla

## Metodi a passo fisso

- ▶ **Fixed step size**
  - ❖ è il valore del passo di integrazione