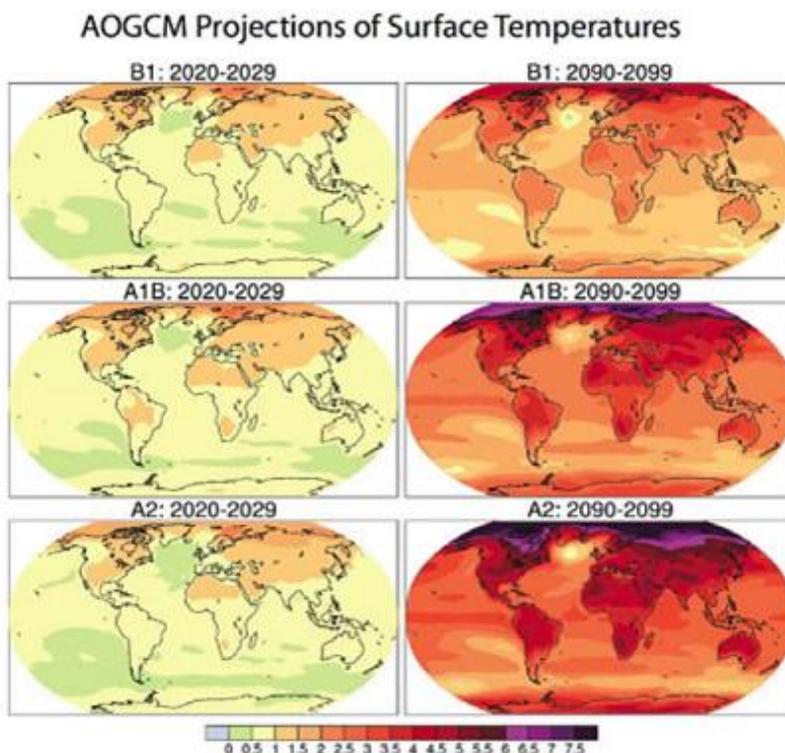


SIMULAZIONE

Che cosa serve:

- un sistema dinamico completamente definito
- un orizzonte di simulazione (intervallo di tempo per il quale sono noti gli ingressi)
- funzioni di ingresso definite per tutto l'orizzonte
 1. $u(\cdot) \equiv 0$ simulazione del movimento libero
 2. $u(\cdot) \equiv \underline{u}(\cdot)$ nota (spesso = manipolabile)
 3. $u(\cdot) \equiv$ campione generato in modo sintetico (metodo Montecarlo)
 - si determina una distribuzione statistica dai dati reali
 - si generano, con il calcolatore, diverse sequenze di numeri (pseudo)casuali con caratteristiche statistiche assegnate (es. media, varianza, autocorrelazione, distribuzione,...uguali a quelle dei possibili ingressi)
 - si simula il sistema in corrispondenza di ciascuna sequenza
 - si valutano le statistiche più significative ottenute
- un algoritmo per la discretizzazione dei sistemi continui
- (possibilmente) software dedicato



In figura sono rappresentate le variazioni della temperatura terrestre all'inizio e alla fine del XXI secolo.

Ogni riga di figure è relativa ad un diverso scenario di emissione; i cambiamenti sono misurati rispetto alle temperature medie del periodo 1980-1999 e a colori più scuri corrispondono aumenti di temperatura maggiori (come indicato nella legenda cromatica); i pannelli a sinistra sono riferiti al periodo 2020-2029 e quelli a destra al periodo 2090-2099. Queste valutazioni sono state ricavate dall'IPCC simulando vari modelli climatici per ciascuno dei diversi scenari di emissione considerati. I valori riportati per ogni scenario sono le medie tra i risultati dei diversi modelli.

METODI DI DISCRETIZZAZIONE

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad \text{Sistema di equazioni differenziali}$$

$$\mathbf{u}(\cdot) \text{ assegnato} \Rightarrow \frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}, t)$$

In generale, su calcolatore,

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}) \Rightarrow \tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \Delta t \mathbf{f}^*(\tilde{\mathbf{x}}_k, \tilde{\mathbf{x}}_{k+1})$$

dove:

- Δt è il passo di discretizzazione
- k (intero e positivo) è l'indice dei passi
- la derivata è sostituita con il rapporto incrementale $(\tilde{\mathbf{x}}_{k+1} - \tilde{\mathbf{x}}_k) / \Delta t$
- $\tilde{\mathbf{x}}_k$ è il valore calcolato all'istante $\mathbf{x}(k\Delta t)$ in generale diverso da $\mathbf{x}(t)$
- $\mathbf{f}^*(\tilde{\mathbf{x}}_k, \tilde{\mathbf{x}}_{k+1})$ è una combinazione convessa ($\sum_i w_i \mathbf{f}_i$ con $\sum_i w_i = 1$) con valori di \mathbf{f} calcolati in istanti diversi.

metodi ESPLICITI: \mathbf{f}^* dipende solo dai valori (calcolati) fino a k

metodi IMPLICITI: \mathbf{f}^* dipende anche dai valori (incogniti) in $k+1$

metodi A UN PASSO: \mathbf{f}^* dipende solo dal valore (calcolati) in k

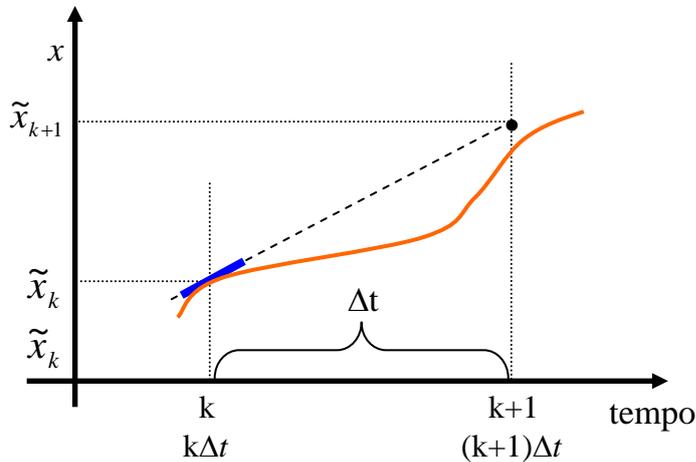
metodi A PIU' PASSI: \mathbf{f}^* dipende da più valori (calcolati) fino a k

metodo ESPLICITO di Eulero (a un passo)

E' il metodo più semplice e intuitivo $\Rightarrow f^* = f$

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \Delta t \mathbf{f}(\tilde{\mathbf{x}}_k)$$

Esempio monodimensionale:



In pratica, la derivata calcolata nell'istante $k\Delta t$ viene estrapolata per tutto l'intervallo Δt stesso

metodo ESPLICITO di Runge-Kutta (a un passo)

Richiede 4 valutazione di f per ogni passo e garantisce errori di troncamento dell'ordine di Δt^4 .

E' il metodo più usato e incluso in tutti i linguaggi e package di simulazione perché rappresenta un buon compromesso tra precisione e velocità.

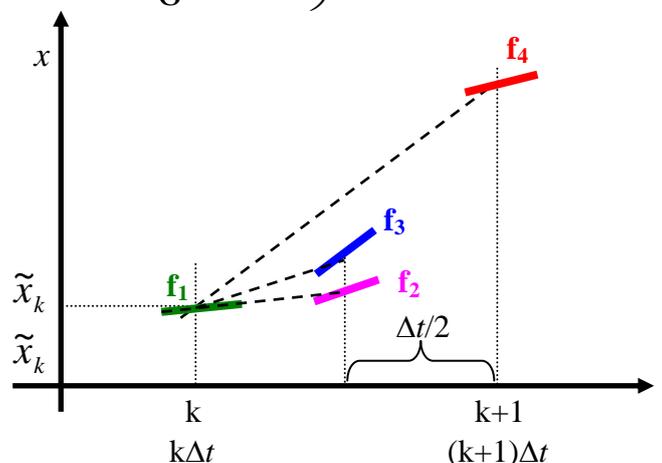
$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \Delta t \left(\frac{1}{6} \mathbf{f}_1(\tilde{\mathbf{x}}_k) + \frac{2}{6} \mathbf{f}_2(\tilde{\mathbf{x}}_k) + \frac{2}{6} \mathbf{f}_3(\tilde{\mathbf{x}}_k) + \frac{1}{6} \mathbf{f}_4(\tilde{\mathbf{x}}_k) \right)$$

con:

$$\mathbf{f}_1(\tilde{\mathbf{x}}_k) = \mathbf{f}(\tilde{\mathbf{x}}_k); \quad \mathbf{f}_2(\tilde{\mathbf{x}}_k) = \mathbf{f}\left(\tilde{\mathbf{x}}_k + \frac{\Delta t}{2} \mathbf{f}(\tilde{\mathbf{x}}_k)\right)$$

$$\mathbf{f}_3(\tilde{\mathbf{x}}_k) = \mathbf{f}\left(\tilde{\mathbf{x}}_k + \frac{\Delta t}{2} \mathbf{f}\left(\tilde{\mathbf{x}}_k + \frac{\Delta t}{2} \mathbf{f}(\tilde{\mathbf{x}}_k)\right)\right)$$

$$\mathbf{f}_4(\tilde{\mathbf{x}}_k) = \mathbf{f}\left(\tilde{\mathbf{x}}_k + \Delta t \mathbf{f}\left(\tilde{\mathbf{x}}_k + \frac{\Delta t}{2} \mathbf{f}\left(\tilde{\mathbf{x}}_k + \frac{\Delta t}{2} \mathbf{f}(\tilde{\mathbf{x}}_k)\right)\right)\right)$$



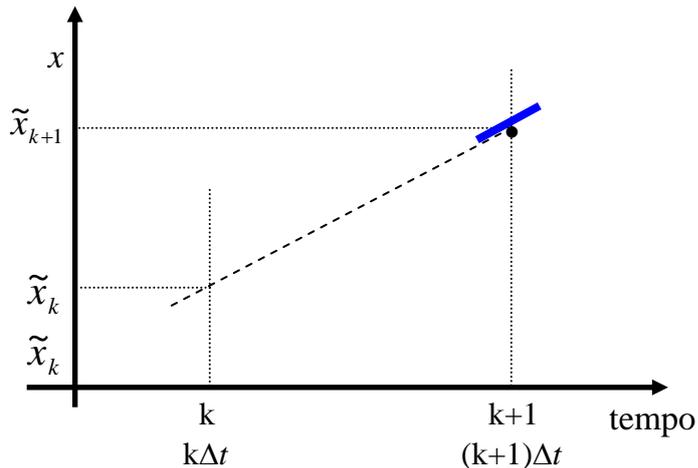
metodo IMPLICITO della derivata destra (a un passo)

Analogo a quello di Eulero, ma utilizza la derivata al termine del passo (incognita) invece di quella all'inizio (nota).

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \Delta t \mathbf{f}(\tilde{\mathbf{x}}_{k+1})$$

Attenzione: è incognita

Esempio monodimensionale:



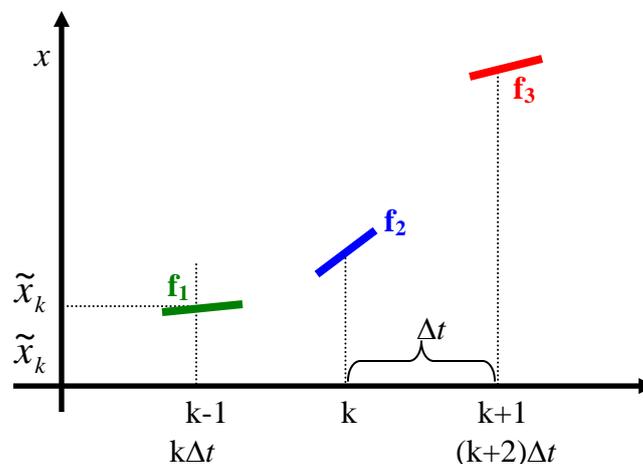
Il metodo definisce ad ogni passo un sistema di equazioni algebriche che va risolto con un opportuno metodo iterativo.

Un metodo a più passi (predictor-corrector)

Utilizza anche informazioni relative al passo $k-1$

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + 2\Delta t \left(\frac{1}{6} \mathbf{f}(\tilde{\mathbf{x}}_{k-1}) + \frac{4}{6} \mathbf{f}(\tilde{\mathbf{x}}_k) + \frac{1}{6} \mathbf{f}(\tilde{\mathbf{x}}_{k+1}) \right)$$

incogniti



Metodi adattativi

Quando il movimento da calcolare subisce rapide variazioni, è difficile definire un passo corretto di discretizzazione:

- passo breve \Rightarrow elevata precisione, ma lunghi tempi di calcolo
- passo lungo \Rightarrow tempi di calcolo più brevi, ma scarsa precisione

Si preferisce quindi far dipendere il passo Δt dalla variazione calcolata:

Es.

se $\tilde{x}((k+1)\Delta t) > 1.5\tilde{x}(k\Delta t) \Rightarrow \Delta t \rightarrow \Delta t/2$

se $\tilde{x}((k+1)\Delta t) < 1.1\tilde{x}(k\Delta t) \Rightarrow \Delta t \rightarrow 1.2\Delta t$

Vantaggi: il passo si adatta alla variabilità del movimento

Svantaggi: il numero di passi per simulare un certo orizzonte non è più calcolabile a priori.

- Tutti i metodi precedenti possono essere resi adattativi
- Le modalità con cui il passo viene legato alla variabilità dei valori calcolati possono essere diverse.
- Il fatto di non conoscere il numero di passi significa non essere in grado di prevedere a priori né la durata della simulazione, né l'occupazione di memoria dei valori calcolati.
- Inoltre, è necessario archiviare sia i valori delle variabili, sia quelli dell'istante di tempo (che non è più un multiplo di Δt) a cui si riferiscono.

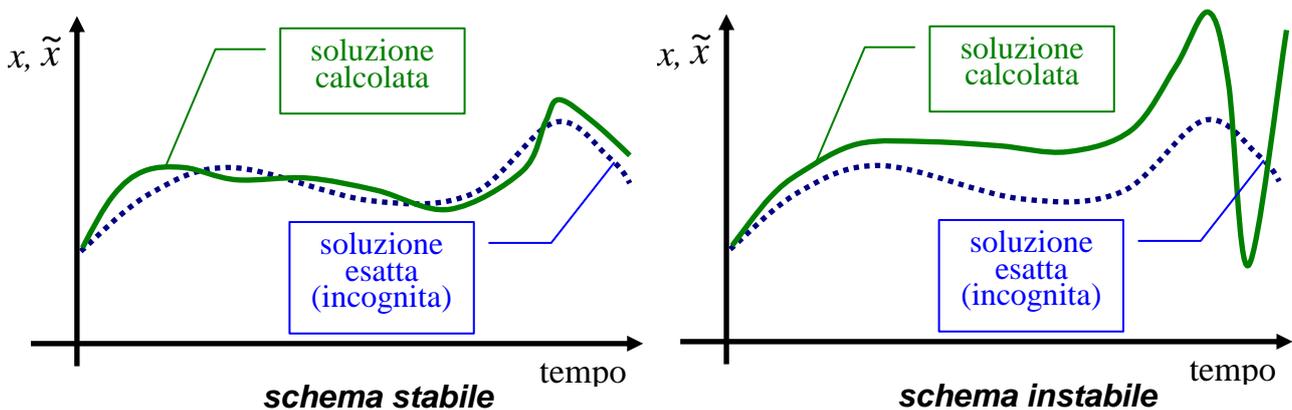
DIFFERENZE E CONCLUSIONI

La soluzione calcolata è in generale diversa da quella esatta.

Più è piccolo Δt ,

- più il rapporto incrementale si avvicina alla derivata,
- più lungo è il tempo di calcolo,
- più ci sono errori di troncamento.

E' essenziale comunque che l'errore $\varepsilon_k = x(k\Delta t) - \tilde{x}_k$ non si accumuli nel tempo (schema stabile), altrimenti lo schema è instabile.



ATTENZIONE: la soluzione esatta è incognita, quindi sarebbe necessaria una garanzia *a priori* della stabilità del metodo.

Analizziamo un metodo esplicito e uno implicito a un passo in un caso ipotetico in cui sia nota la soluzione esatta.

DINAMICA DELL'ERRORE

Consideriamo l'equazione (lineare, monodimensionale)

$$\dot{x} = -\lambda x \quad \text{con } \lambda > 0$$

la cui soluzione analitica è **nota**

$$x((k+1)\Delta t) = e^{-\lambda\Delta t} x(k\Delta t) = (1 - \lambda\Delta t)x(k\Delta t) + (\lambda^2\Delta t^2 / 2! + \dots)x(k\Delta t)$$

mentre la soluzione calcolata con il **metodo di Eulero** è

$$\tilde{x}_{k+1} = \tilde{x}_k + \Delta t(-\lambda\tilde{x}_k) = (1 - \lambda\Delta t)\tilde{x}_k$$

Sottraendo la seconda equazione alla prima:

$$\varepsilon_{k+1} = (1 - \lambda\Delta t)\varepsilon_k + (\lambda^2\Delta t^2 / 2! + \dots)x(k\Delta t) \quad \text{dinamica dell'errore}$$

quindi l'errore è rappresentato da un modello lineare e non si accumula se $\varepsilon_{k+1} < \varepsilon_k$

$$|1 - \lambda\Delta t| < 1, \quad \text{cioè se } \Delta t < 2/\lambda$$

La soluzione calcolata con il **metodo della derivata destra** è

$$\tilde{x}_{k+1} = \tilde{x}_k + \Delta t(-\lambda\tilde{x}_{k+1}),$$

che, risolta rispetto a \tilde{x}_{k+1} , dà $\tilde{x}_{k+1} = \tilde{x}_k / (1 + \lambda\Delta t)$

Sottraendo le due equazioni otteniamo:

$$\varepsilon_{k+1} = \frac{1}{1 + \lambda\Delta t} \varepsilon_k + \left(e^{-\lambda\Delta t} - \frac{1}{1 + \lambda\Delta t} \right) x(k\Delta t)$$

anche in questo caso l'errore è rappresentato da un modello

lineare $\varepsilon_{k+1} = A\varepsilon_k + Bu_k$ **e non si accumula** ($\varepsilon_{k+1} < \varepsilon_k$)

(incondizionatamente stabile) perché $|1/(1 + \lambda\Delta t)| < 1 \forall \Delta t$.

CONCLUSIONE: se il movimento calcolato è instabile, prima di trarre conclusioni, è opportuno provare a rifare la simulazione cambiando metodo o riducendo il passo. Se anche per passi decisamente piccoli, rispetto alla velocità con cui si immagina che evolva il sistema, si ottiene un'instabilità si può attribuirlo al movimento con una certa sicurezza.

ERRORI DI TRONCAMENTO: Esempio

Simuliamo il sistema

$$\frac{dx}{dt} = 0,2x - 0,3x^2 + y$$

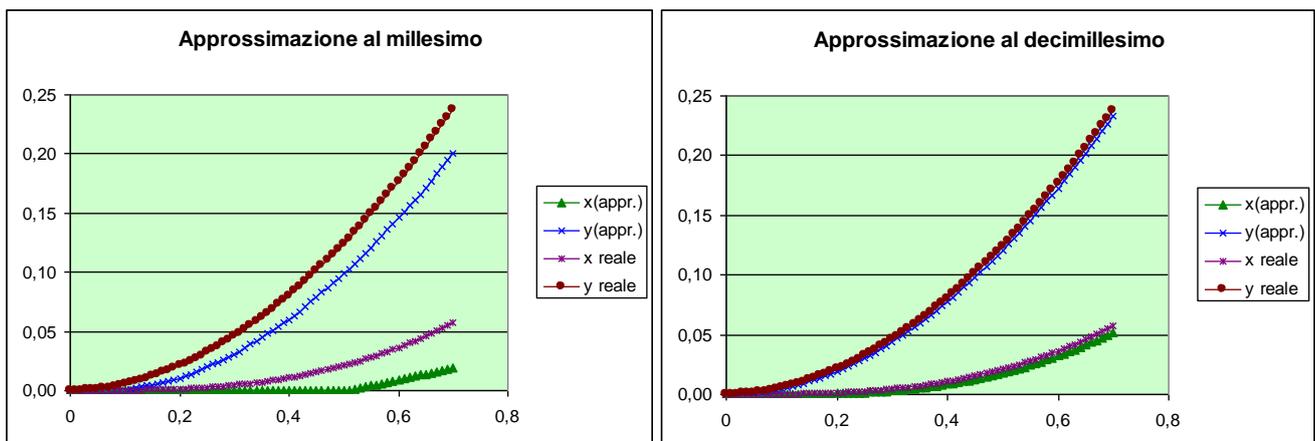
$$\frac{dy}{dt} = 0,5x - 0,1y + \text{sen}(t)$$

con il metodo di Eulero e determiniamo l'effetto di differenti troncamenti dei valori calcolati delle variabili di stato.

Utilizziamo il solito foglio Excel, sul quale creiamo due colonne per i valori calcolati con il metodo di Eulero $[x(k+1)=x(k)+\Delta t f(x(k),u(k))]$ e altre due per i valori troncati ad un certo decimale. Un modo per ottenere questi ultimi (normalmente la precisione del calcolo è quella interna del programma e quindi estremamente elevata), è prendere la parte intera (INT()) del valore moltiplicato per 10^i , dove i è la precisione che vogliamo ottenere in termini di cifre decimali, e dividere nuovamente l'intero ottenuto per 10^i .

Naturalmente, per simulare correttamente l'effetto del troncamento, occorre che le equazioni del metodo di Eulero siano applicate ad ogni passo ai valori troncati così ottenuti.

Le figure seguenti mostrano i risultati con passo di discretizzazione pari a 0,01 e i troncamenti rispettivamente pari a 3 (troncamento al millesimo) e 4 (al decimillesimo). Per confronto è mostrato anche il risultato "reale" (cioè quantizzato alla massima precisione possibile).



Già entro 0,7 unità di tempo, la differenza tra i risultati è evidente: l'errore massimo nel primo caso arriva al 100% del valore massimo di x , mentre nel secondo è circa il 50% (ma x ha inizialmente valori molto piccoli). L'errore massimo sulla y è invece del 44% nel primo caso e solo del 7% nel secondo.