Introduzione al Calcolo Scientifico

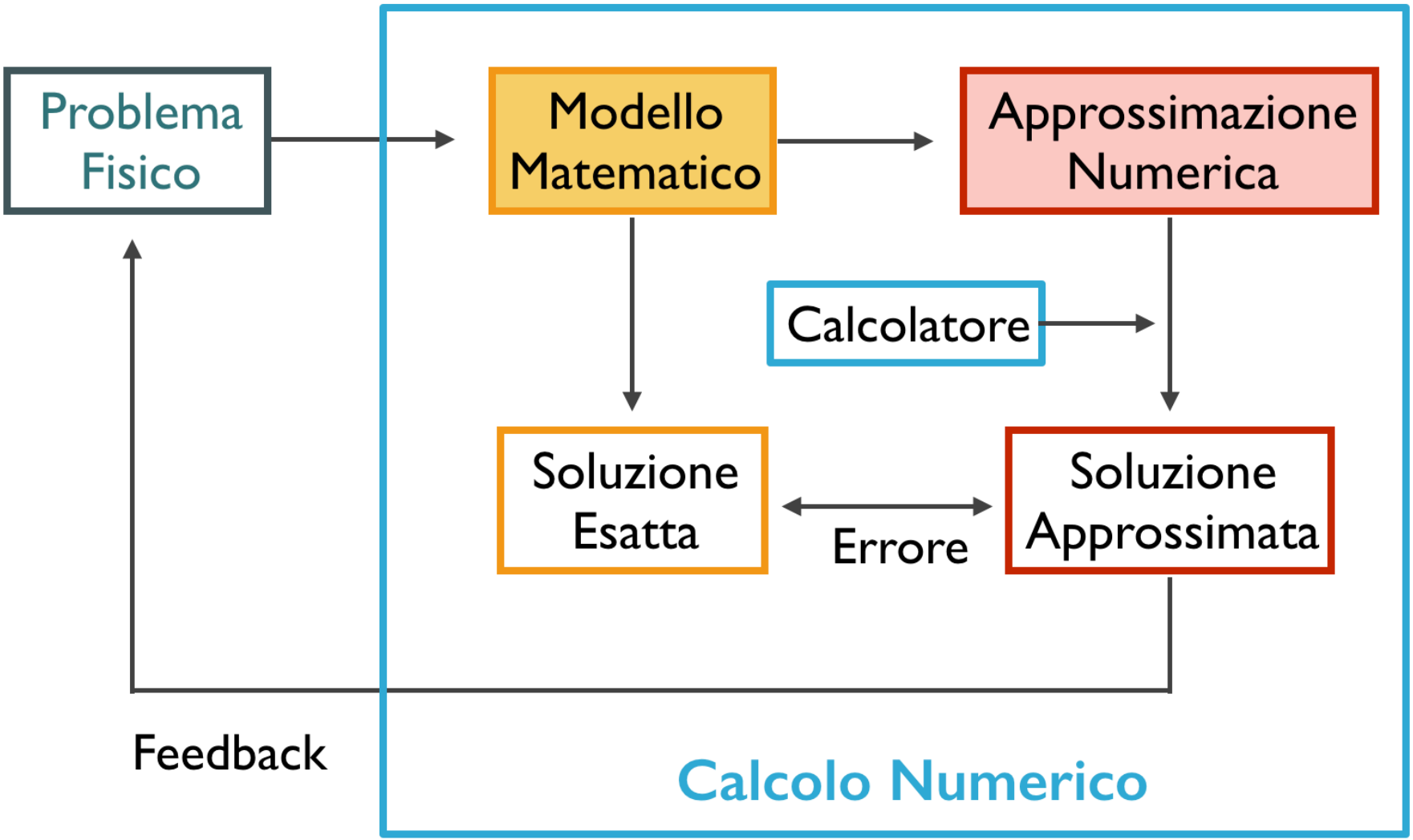
Calcolo Numerico ed Elementi di Analisi

Andrea Manzoni

POLITECNICO
MILANO 1863

MOX - Dipartimento di Matematica
POLITECNICO DI MILANO

Secondo Semestre 2021–22


Problema Fisico
Modello Matematico
Approssimazione Numerica
Calcolatore
Soluzione Esatta
Errore
Soluzione Approssimata
Feedback
Calcolo Numerico

Introduzione

1. Dappertutto si nascondono problemi matematici...
2. Dal problema alla soluzione
3. Dal problema matematico al problema numerico
4. Rappresentazione macchina dei numeri reali

Dappertutto si nascondono problemi matematici...

Esempio 1

Un circuito elettrico a corrente continua (CC) con solo una batteria e resistori fornisce un **sistema di equazioni lineari**. Per determinare la corrente e la caduta di tensione su ciascun resistore nel circuito, abbiamo solo bisogno della legge di Ohm e della legge di Kirchhoff delle tensioni (o delle correnti):

Legge di Ohm

Quando una corrente di I ampere scorre attraverso un resistore di R ohm, la caduta di tensione, V, in volt attraverso il resistore è

$$V = I\,R.$$

Legge di Kirchhoff delle tensioni

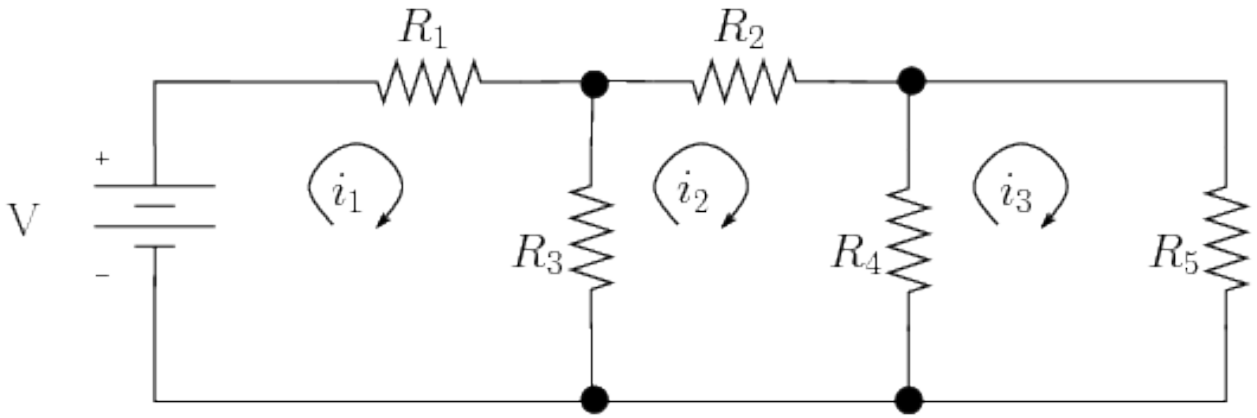
La somma algebrica delle tensioni lungo una linea chiusa (con il segno appropriato in funzione del verso di percorrenza della maglia stessa) è pari a zero.

Legge di Kirchhoff delle correnti

In ogni nodo del circuito, la somma delle correnti entranti è uguale alla somma delle correnti uscenti.

Dappertutto si nascondono problemi matematici...

Sfruttando la legge di Kirchhoff delle tensioni, giungiamo facilmente a un **sistema di equazioni lineari**...



In questo caso, le tre equazioni corrispondenti alle tre maglie in figura si possono esprimere in funzioni delle correnti i_1, i_2, i_3, che fluiscono in ciascuna maglia:

$$\begin{cases} i_1 R_1 + (i_1 - i_2) R_3 = v \\ i_2 R_2 + (i_2 - i_3) R_4 + (i_2 - i_1) R_3 = 0 \\ i_3 R_5 + (i_3 - i_2) R_4 = 0 \end{cases}$$

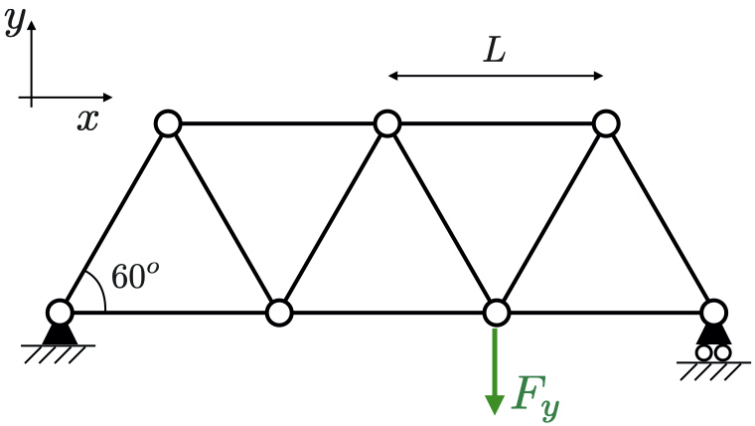
In notazione matriciale, si ottiene

$$\begin{bmatrix} (R_1 + R_3) & -R_3 & 0 \\ -R_3 & (R_2 + R_3 + R_4) & -R_4 \\ 0 & -R_4 & (R_4 + R_5) \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix}$$

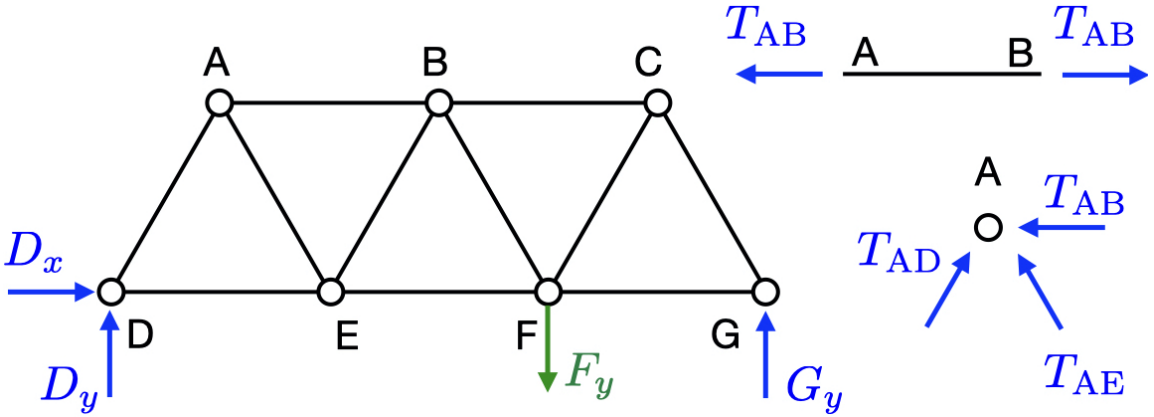
Dappertutto si nascondono problemi matematici...

Esempio 2

Consideriamo il problema della statica di un ponte a travi composto da aste e giunti.



Vogliamo determinare le reazioni vincolari D_x, D_y, G_y e le forze interne alle aste (T_{AB}, etc.), noti i vincoli e la forza assegnata F_y. Tale problema si può riscrivere come un **sistema di equazioni lineari**.

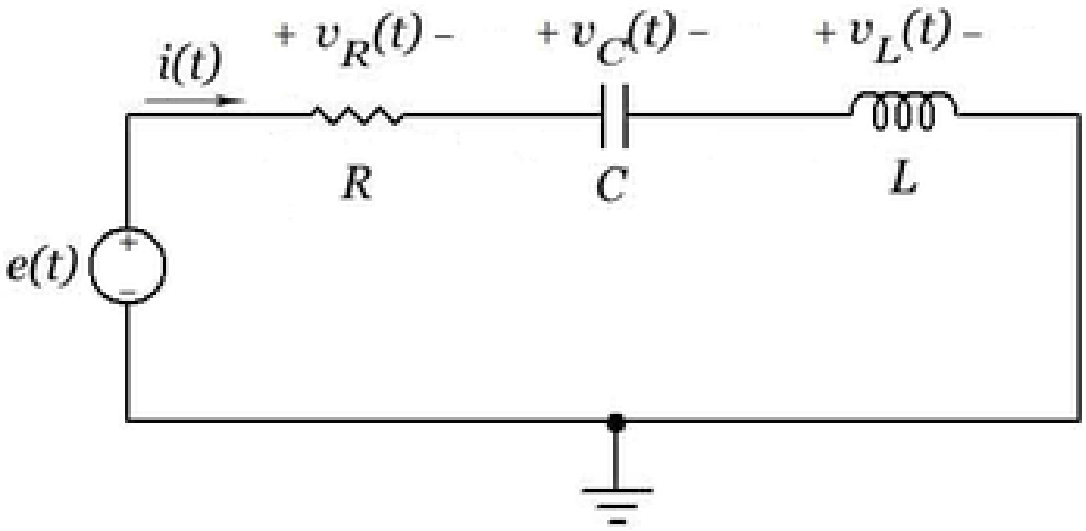


$$
\begin{aligned}
D_x &= 0 \\
D_y + G_y &= F_y \\
3L\,G_y &= 2L\,F_y \\
\tfrac{1}{2}\,T_{\mathsf{AD}} - \tfrac{1}{2}\,T_{\mathsf{AE}} + T_{\mathsf{AB}} &= 0 \\
\tfrac{\sqrt{3}}{2}\,T_{\mathsf{AD}} + \tfrac{\sqrt{3}}{2}\,T_{\mathsf{AE}} &= 0 \\
\vdots &= \vdots
\end{aligned}
$$

Dappertutto si nascondono problemi matematici...

Esempio 3

Un circuito RLC con condensatori e induttori oltre a resistori è descritto da un **sistema di equazioni differenziali ordinarie**.



Ad esempio, nel caso del circuito RLC in serie in figura, applicando la legge di Kirchhoff delle tensioni si ottiene:

$$v_R(t) + v_L(t) + v_C(t) = e(t)$$

e sostituendo le relazioni costitutive degli elementi:

$$Ri(t) + L \cdot \frac{di(t)}{dt} + \frac{1}{C} \int_0^t i(t)\, dt = e(t).$$

Dappertutto si nascondono problemi matematici...

Tenendo presente che come generatore di tensione costante $e(t) = e_0$, derivando una volta rispetto a t e dividendo per l'induttanza L si può riscrivere l'equazione in forma differenziale[1]:

$$\frac{R}{L} \cdot \frac{di(t)}{dt} + \frac{d^2 i(t)}{dt^2} + \frac{1}{LC} \cdot i(t) = 0.$$

Definendo poi i due parametri:

$$\alpha = \frac{R}{2L} \qquad \text{costante di smorzamento}$$

$$\omega_0 = \frac{1}{\sqrt{LC}} \qquad \text{pulsazione di risonanza}$$

si ottiene infine

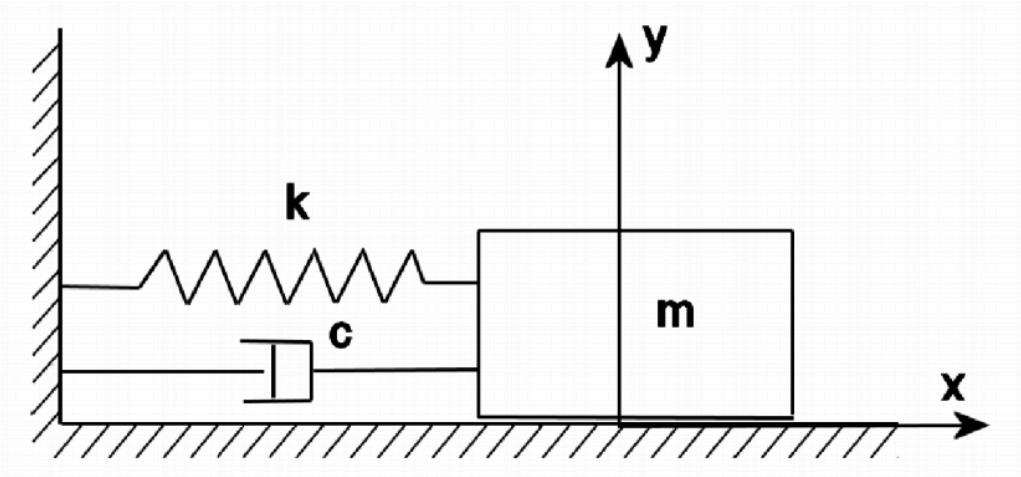
$$\frac{d^2 x(t)}{dt^2} + 2\alpha \cdot \frac{dx(t)}{dt} + \omega_0^2 \cdot x(t) = 0$$

[1] La presenza di un generatore costante non influisce sulle equazioni: la soluzione dell'equazione è la stessa di quella senza generatore, come se fosse in evoluzione libera.

Dappertutto si nascondono problemi matematici...

Esempio 4

Si consideri il sistema massa-molla-smorzatore



La molla esercita una forza proporzionale allo spostamento dalla posizione di riposo $F_k = -kx$.

Se aggiungiamo al sistema una forza di attrito proporzionale alla velocità $F_c = -cx'$, e scriviamo il bilancio delle forze $F_k + F_c = mx''$, si ottiene l'**equazione differenziale di ordine 2** dell'oscillatore smorzato:

$$mx'' + cx' + kx = 0.$$

Dividendo per m e ponendo $\gamma = c/m$, $\omega^2 = k/m$ si ottiene:

$$x'' + \gamma x' + \omega^2 x = 0.$$

Dappertutto si nascondono problemi matematici...

Esempio 5

Si consideri il problema di Keplero inerente due corpi puntiformi aventi masse m_1 e m_2 e posizioni $\mathbf{r}_1(t)$ e $\mathbf{r}_2(t)$ al tempo t in un sistema di riferimento inerziale e che interagiscono tramite la forza gravitazionale.

Assumiamo che il moto dei due corpi avvenga in un piano: $\mathbf{r}_i(t) = r_{i,x}(t)\,\hat{\mathbf{x}} + r_{i,y}(t)\,\hat{\mathbf{y}}$, per $i = 1, 2$.

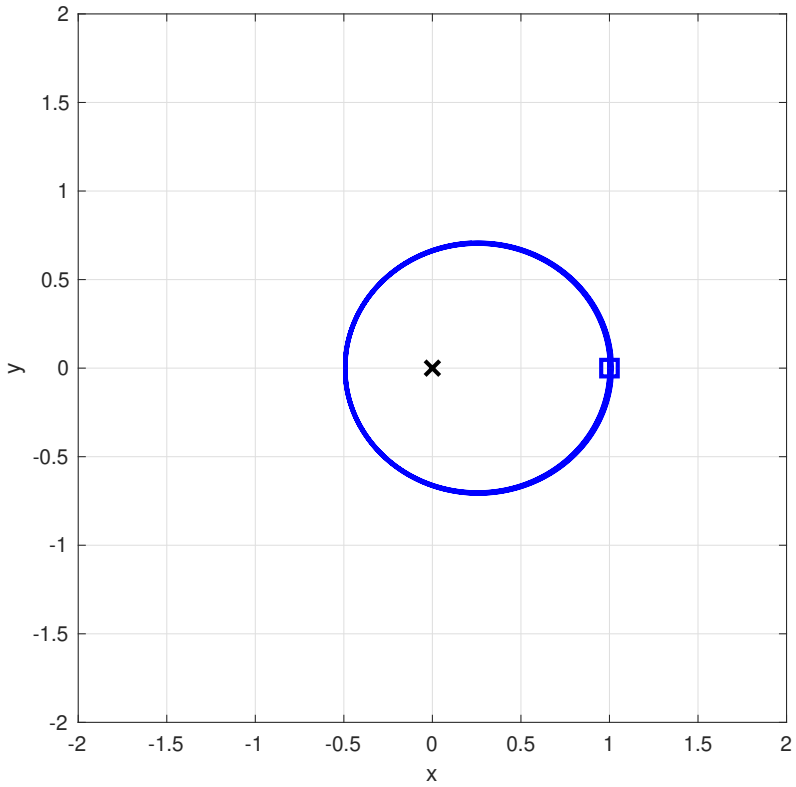
Vogliamo determinare l'orbita di Keplero del corpo 2 assimilabile al pianeta Terra, ovvero $\mathbf{r}_2(t)$, assumendo che $\mathbf{r}_1(t) = \mathbf{0}$ per ogni $t \geq t_0$, essendo il corpo 1 il Sole di massa $m_1 \gg m_2$.

Ponendo dunque $m = m_1$, $\mathbf{r}(t) = \mathbf{r}_2(t)$, il modello matematico che descrive il fenomeno è rappresentato dal seguente **sistema di equazioni differenziali ordinarie del secondo ordine** in forma adimensionale:

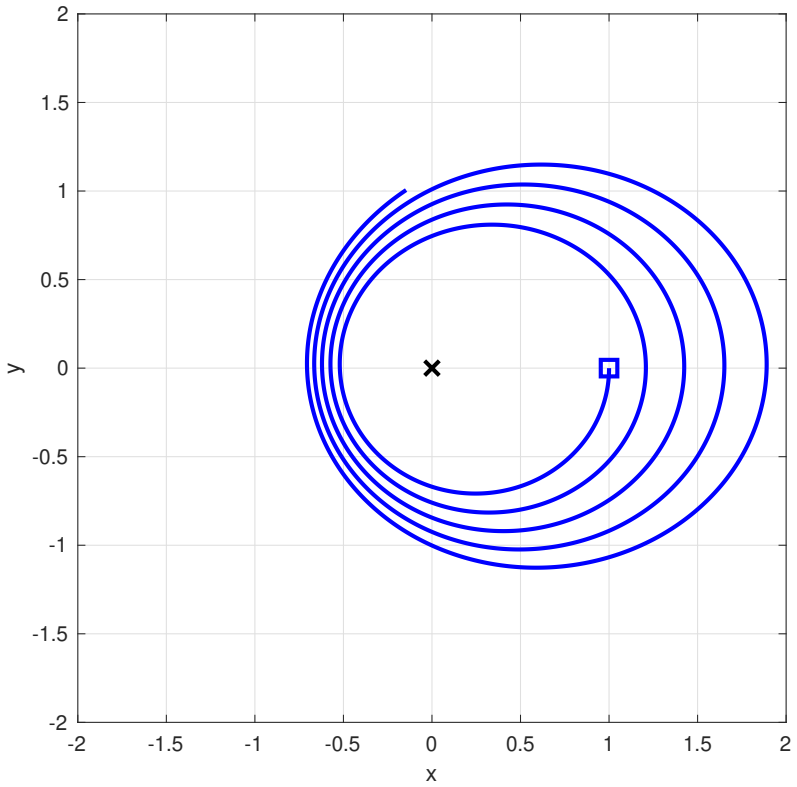
$$\begin{cases} \mathbf{r}''(t) = -4\,\pi^2\,\dfrac{\mathbf{r}(t)}{\|\mathbf{r}(t)\|^3} & t \in (t_0, t_f), \\ \mathbf{r}(t_0) = \mathbf{r}_0, \\ \mathbf{r}'(t_0) = \mathbf{v}_0, \end{cases}$$

dove $\mathbf{r}_0$ è la posizione iniziale e $\mathbf{v}_0$ la velocità iniziale del corpo 2; $\mathbf{r}(t)$ esprime la posizione della Terra rispetto al Sole in unità astronomiche.

Dappertutto si nascondono problemi matematici...



Orbita corrispondente alle soluzioni approssimate $r_x(t)$ e $r_y(t)$ ottenute tramite il metodo di Eulero in avanti, $h = 10^{-5}$ (caso **stabile**)



Orbita corrispondente alle soluzioni approssimate $r_x(t)$ e $r_y(t)$ ottenute tramite il metodo di Eulero in avanti, $h = 10^{-3}$ (caso **instabile**)

Dappertutto si nascondono problemi matematici...

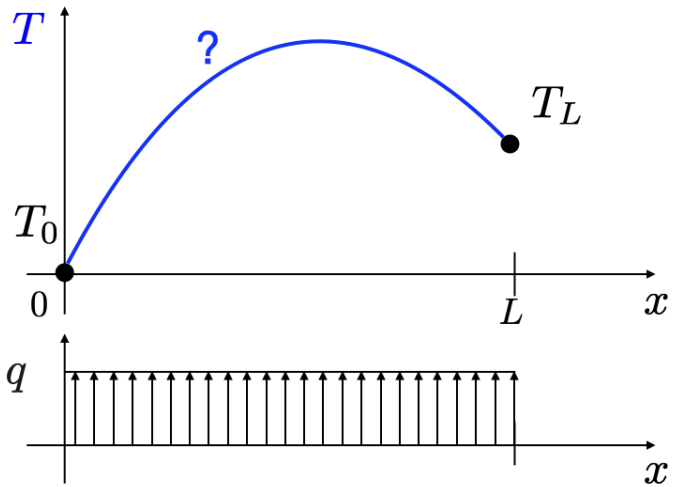
Esempio 6

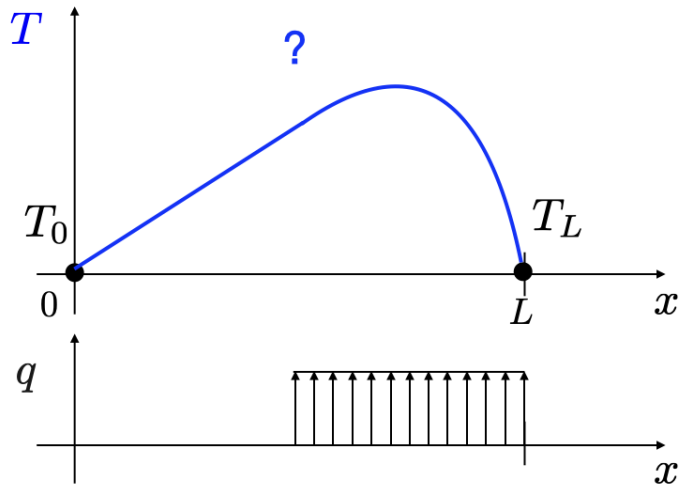
Si consideri un problema di trasmissione del calore stazionario in una barra metallica.

Il problema consiste nel determinare la *temperatura* $T(x)$ in ogni punto di coordinata $x \in (0, L)$, sapendo che la temperatura della barra è nota agli estremi del filo (con valori T_0 e T_L), che è assegnata una sorgente di calore distribuita $f(x)$ e che la conduttività termica k del filo è nota.

Il modello matematico è rappresentato dal seguente **problema ai limiti** (un caso particolare di equazione alle derivate parziali del secondo ordine):

$$\begin{cases} -k\,T''(x) = q(x) & x \in (0, L), \\ T(x = 0) = T_0, \\ T(x = L) = T_L. \end{cases}$$





Dal problema alla soluzione

1. Problema reale: determinare la configurazione di equilibrio di una struttura (ovvero il suo spostamento u) dato un certo carico f.

Assumiamo che:

- la struttura sia uni-dimensionale, descritta da una linea elastica di
- lunghezza unitaria, massa trascurabile, fissata agli estremi,
- il regime sia quello di piccoli spostamenti
- la struttura sia soggetta a una forza trasversale di intensità f (per unità di lunghezza).

2. Modello Matematico: trovare $u = u(x)$ tale che

$$\begin{aligned} -u''(x) &= f(x), \qquad 0 < x < 1 \\ u(0) &= 0, \quad u(1) = 0. \end{aligned} \tag{1}$$

La funzione u descrive lo spostamento in direzione verticale della corda rispetto alla configurazione non caricata $u = 0$.

Inoltre, siamo interessati a calcolare il lavoro delle forze esterne,

$$W = \int_0^1 g(x)\,dx, \qquad \text{con} \quad g(x) = f(x)u(x) \tag{2}$$

(parliamo di **problema matematico** o problema nel continuo, e soluzione esatta)

Se f è costante, $c > 0$, il problema (1) può essere risolto analiticamente,

$$u(x) = \frac{c}{2} x(1-x)$$

e anche la valutazione dell'integrale (2) è banale, poiché

$$W = \int_0^1 c\frac{c}{2} x(1-x)dx = c^2/12.$$

In generale, la soluzione del problema (1) può essere sempre caratterizzata mediante un integrale che coinvolge f: per ogni $f \in C^0([0,1])$ esiste un'unica soluzione $u \in C^2([0,1])$ data da

$$u(x) = x \int_0^1 (1-s)f(s)ds - \int_0^x (x-s)f(s)ds. \tag{3}$$

Quando f non è una funzione elementare, la valutazione dell'integrale in (3) (così come in (2)) può diventare molto complicata, se non addirittura impossibile.

3. Discretizzazione Numerica: ogni passaggio al limite (e quindi ogni derivata e ogni integrale) deve essere approssimato, discretizzato.

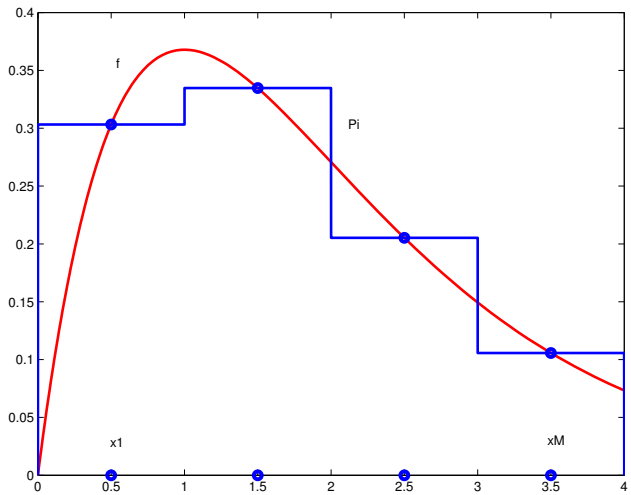
Approssimazione di integrali

Possiamo approssimare l'integrale in (2) introducendo su $[0, 1]$ una partizione in sottointervalli

$$I_j = [x_{j-1}, x_j], \qquad j = 1, \ldots, N$$

di uguale lunghezza $h = 1/N$, dati i nodi $x_j = jh$, e valutando (*formula del punto medio composita*, applicata all'integranda g)

$$W_N = \frac{1}{N} \sum_{j=1}^{N} g(\bar{x}_j), \qquad \text{dove} \quad \bar{x}_j = \frac{x_{j-1} + x_j}{2}$$



Per N sufficientemente grande, $W_N \to W$; l'errore $|W - W_N|$ si comporta come $1/N$: per essere accurati alla terza cifra decimale, occorrono circa $N = 100$ nodi.

Approssimazione delle derivate

Un possibile modo di discretizzare il problema è assumere che l'equazione differenziale sia soddisfatta in un insieme di punti x_j interni a $(0,1)$ (*metodo delle differenze finite*)

$$-u''(x_j) = f(x_j), \qquad j = 1, \ldots, N.$$

Possiamo approssimare queste N equazioni sostituendo alla derivata seconda la quantità (*differenza finita*)

$$\delta^2 u(\bar{x}) = \frac{u(\bar{x}+h) - 2u(\bar{x}) + u(\bar{x}-h)}{h^2}.$$

Come vedremo, se $u : [0,1] \to \mathbb{R}$ è una funzione sufficientemente regolare in un intorno di un generico punto $\bar{x} \in (0,1)$, $\delta^2 u(\bar{x})$ fornisce un'approssimazione di $u''(\bar{x})$ di ordine 2 rispetto ad h, ovvero l'errore $|u''(\bar{x}) - \delta^2 u(\bar{x})|$ si comporta come $1/N^2$.

(parliamo di **problema numerico** o problema discreto e soluzione discreta)

Possiamo approssimare il problema (1) come: trovare $\{u_j\}_{j=1}^N$ tale che

$$\begin{cases} -\dfrac{u_{j+1} - 2u_j + u_{j-1}}{h^2} = f(x_j), & j = 1, \ldots, N \\ u_0 = u_{N+1} = 0. \end{cases} \tag{4}$$

dove u_j indica l'approssimazione di $u(x_j)$. Le equazioni $(4)_1$ forniscono un sistema lineare

$$A\mathbf{u} = \mathbf{f}, \tag{5}$$

dove $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_{N-1}), f(x_N))^T$, $\mathbf{u} = (u_1, \ldots, u_N)^T$ è il vettore delle incognite, e A è la matrice (tridiagonale)

$$A = \frac{1}{h^2}\texttt{tridiag}(-1, 2, 1) = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & 0 & \ldots & 0 \\ -1 & 2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & -1 & 0 \\ \vdots & & -1 & 2 & -1 \\ 0 & \ldots & 0 & -1 & 2 \end{bmatrix}. \tag{6}$$

Se $f \in C^2([a, b])$, l'errore tra la soluzione esatta e quella approssimata è tale che

$$\max_{j=0,\ldots,N+1} |u(x_j) - u_j| \leq \frac{h^2}{96} \max_{x\in[a,b]} |f''(x)|$$

1. Dal problema matematico al problema numerico

1. Il problema matematico e il numero di condizionamento
2. Il problema numerico
3. Gli inevitabili errori

Il problema matematico e il numero di condizionamento

Da un punto di vista astratto, un *problema matematico* (PM) può essere espresso nella forma: trovare x tale che

$$F(x, d) = 0, \tag{7}$$

dove

- $d \in \mathcal{D}$ rappresenta l'insieme dei dati del problema,
- $x \in \mathcal{X}$ la soluzione,
- F il legame (funzionale) tra dati e soluzione,
- $\mathcal{X}$ e $\mathcal{D}$ sono due opportuni spazi.

Un problema di questo tipo si ottiene modellizzando un *problema fisico* (PF), la cui soluzione – chiamata anche *soluzione fisica* – viene indicata con x_f; l'errore di modello,

$$e_m = x_f - x$$

tiene conto di quelle componenti/caratteristiche del PF non rappresentate nel PM. Non ci occuperemo di questa componente di errore.

Esempio

Consideriamo il problema fisico dell'equilibrio della corda elastica, fissata agli estremi. Sotto le ipotesi fatte in precedenza, il problema matematico è: trovare lo spostamento u *(la soluzione),* $u = u(\xi)$ *tale che*

$$F(u, d) = u - \xi \int_0^1 (1 - s) f(s) ds + \int_0^\xi (\xi - s) f(s) ds = 0,$$

i cui dati sono $d = (0, 1, f)$*; in questo caso* $\mathcal{D} = \mathbb{R} \times \mathbb{R} \times C^0([0, 1])$*, mentre* $\mathcal{X} = C^2([0, 1])$.

Definizione

Il problema matematico (7) *è* **ben posto** *se esso ammette un'unica soluzione* $x \in \mathcal{X}$ *che dipende con continuità dai dati* $d \in \mathcal{D}$.

Dipendenza continua dai dati significa che piccole perturbazioni sui dati $d \in \mathcal{D}$ inducono *piccole* variazioni nella soluzione $x \in \mathcal{X}$.

Un problema che non soddisfa queste proprietà è detto mal posto o instabile[2].

Esempio

Trovare il numero di radici reali di un polinomio non è un problema ben posto. Ad esempio, in

$$F(x, d) = x^4 - x^2(2d - 1) + d(d - 1) = 0$$

il numero di radici reali subisce una discontinuità al variare di d *in* $\mathbb{R}$*; si hanno* 4 *radici reali se* $d \geq 1$*,* 2 *se* $d \in (0, 1)$*, invece non ci sono radici reali se* $d < 0$.

Tuttavia, problemi matematici che sono ben posti possono mostrare una *grande* variazione nella soluzione anche in presenza di *piccole perturbazioni* sui dati.

[2] Useremo a volte i termini *ben posto* e *stabile* in modo intercambiabile.

Sia δd una perturbazione sui dati (ammissibile, tale cioè che $d + \delta d \in \mathcal{D}$) e sia δx la conseguente variazione nella soluzione, in modo che si abbia

$$F(x + \delta x, d + \delta d) = 0.$$

Definizione

Il numero di condizionamento (relativo) di un problema $F(x; d) = 0$ è

$$K(d) := \sup \left\{ \frac{\|\delta x\|/\|x\|}{\|\delta d\|/\|d\|} \quad \forall \delta d \ : \ d + \delta d \in \mathcal{D} \right\}.$$

Per definizione, $K(d) \geq 1$.

- Se $K(d)$ è *piccolo* per ogni dato ammissibile d il PM (7) è ben condizionato;
- se $K(d)$ è *grande*, il problema si dice *mal condizionato*.

Se un problema non è ben posto perché la soluzione non dipende con continuità dai dati, tale problema risulterà anche mal condizionato.

Remark

La proprietà di un problema di essere ben condizionato è indipendente dal metodo numerico usato per approssimarlo.

Esempio

Consideriamo il problema di moltiplicare due numeri reali,

$$F(x, \mathbf{d}) = x - d_1 d_2 = 0$$

essendo $\mathbf{d} = (d_1, d_2)^T$ *e perturbiamone i dati prendendo*

$$\tilde{\mathbf{d}} = \mathbf{d} + \delta\mathbf{d} = (\tilde{d}_1, \tilde{d}_2)^T = (d_1(1+\epsilon), d_2(1+\epsilon))^T;$$

in questo caso

$$\|\delta\mathbf{d}\| = \epsilon\|\mathbf{d}\|$$

e

$$\frac{|\delta x|}{|x|} = \frac{|\tilde{d}_1\tilde{d}_2 - d_1 d_2|}{|d_1 d_2|} = \frac{|d_1 d_2(1+\epsilon)^2 - d_1 d_2|}{|d_1 d_2|} = (1+\epsilon)^2 - 1 = \epsilon^2 + 2\epsilon.$$

Per piccole perturbazioni ($\varepsilon \ll 1$ *) possiamo trascurare il fattore* $\epsilon^2 \ll 2\epsilon$ *e considerare* $K(d) = 2$*: il problema di moltiplicare due numeri reali è ben condizionato. Si ha cioè*

$$\frac{|\delta x|/|x|}{\|\delta\mathbf{d}\|/\|\mathbf{d}\|} = \frac{|\tilde{d}_1\tilde{d}_2 - d_1 d_2|/|d_1 d_2|}{\|\delta\mathbf{d}\|/\|\mathbf{d}\|} \leq 2.$$

Esempio

Consideriamo ora il problema di sottrarre due numeri reali,

$$F(x, \mathbf{d}) = x - (d_1 - d_2) = 0.$$

Perturbando i dati $\mathbf{d} = (d_1, d_2)^T$ *come nell'esempio precedente, per cui* $\|\delta\mathbf{d}\| = \epsilon\|\mathbf{d}\|$*; ora*

$$\frac{|\delta x|}{|x|} = \frac{|(\tilde{d}_1 - \tilde{d}_2) - (d_1 - d_2)|}{|d_1 d_2|} = \frac{|d_1\epsilon - d_2\epsilon|}{|d_1 - d_2|} \leq \frac{|d_1| + |d_2|}{|d_1 - d_2|}\epsilon$$

Se $sign(d_1) = -sign(d_2)$ *(ovvero il problema è quello di addizionare due numeri) si ha* $K(d) = 1$*; il problema di addizionare due numeri è ben condizionato.*
Se invece i segni sono gli stessi, e $d_1 \approx d_2$*, allora*

$$K(d) = \frac{|d_1| + |d_2|}{|d_1 - d_2|}$$

diventa molto grande: il problema di addizionare due numeri NON è ben condizionato.

Ad esempio, se $d_1 = 1/51$ *e* $d_2 = 1/52$*; allora* $K(d) = 103$*. Infatti, se* $\tilde{d}_1 = 0.196 \cdot 10^{-1}$ *e* $\tilde{d}_2 = 0.192 \cdot 10^{-1}$*, si ottiene* $\tilde{d}_1 - \tilde{d}_2 = 0.400 \cdot 10^{-3}$*, molto diverso dal risultato esatto* $d_1 - d_2 = 0.377 \cdot 10^{-3}$*; il grande numero di condizionamento si riflette nel fatto che la soluzione sia incline a errori grandi dovuti a cancellazione.*

Perché valutare la propagazione degli errori dovuti a piccole perturbazioni?

Principio di Wilkinson

Il risultato di un'operazione numerica al calcolatore (o in aritmetica floating point) equivale al risultato della medesima operazione in aritmetica esatta effettuata su dati affetti da una (piccola) perturbazione.

Il problema numerico

Consideriamo un PM che sia ben posto. Un **METODO NUMERICO** per la risoluzione approssimata di (7) consiste nella costruzione di una *successione* di problemi numerici (o approssimati): trovare $x_h \in \mathcal{X}_h$ tale che

$$F_h(x_h, d_h) = 0 \tag{8}$$

dipendenti da un certo *parametro* h (a seconda dei casi, si usa h oppure n per indicizzare tali problemi; solitamente, $h \to 0 \Leftrightarrow n \to \infty$).

Esempio

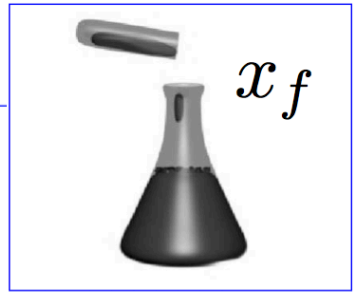
Se consideriamo il PM

$$F(x; d) = x - \int_a^b f(t)dt = 0, \qquad d = \{a, b, f\}$$

possiamo considerare il seguente problema numerico (per un determinato $h > 0$) per approssimare l'integrale

$$F_h(x_h; d_h) = x_h - h \sum_{k=1}^{n} f\left(\frac{t_k + t_{k+1}}{2}\right), \qquad n \geq 1 \tag{9}$$

dove $h = (b-a)/n$ *e* $t_k = a + (k-1)h$, $k = 1, \ldots, n+1$



Problema Fisico

Problema Matematico

$$x = \int_0^T \phi(t)dt$$

Problema Numerico

$$x_h = \sum_k \phi(t_k)\alpha_k$$

Gli inevitabili errori

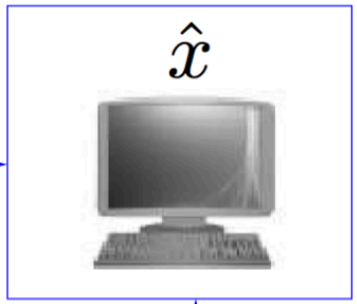
e_m

Problema Fisico

Problema Matematico

$$x = \int_0^T \phi(t)dt$$

e_c



Problema Numerico

e_t

$$x_h = \sum_k \phi(t_k)\alpha_k$$

e_a

e_m=errore di modello, e_t=errore di troncamento,

e_a=errore di arrotondamento, e_c=errore computazionale

$(e_c = e_t + e_a)$, (tipicamente $|e_a| \ll |e_t|$, $x_h \approx \widehat{x}$)

Problema Numerico: concetti fondamentali

1. Quando il problema numerico risulta ben posto (o stabile)?
   (**stabilità**)
2. Quando il problema numerico *riproduce* correttamente il problema matematico?
   (**consistenza**)
3. Sotto quali condizioni $x_h \to x$ per $h \to 0$?
   Ovvero, l'errore $e_c = e_c(x_h) := \|x - x_h\|$ è tale che $e_c \to 0 \quad$ per $h \to 0$?
   (**convergenza**)
4. Possiamo garantire che l'errore e_c sia tale che $e_c = e_c(x_h) \leq C\, h^p$?
   (**convergenza** di ordine p)
5. Quali relazioni intercorrono tra i concetti di stabilità e convergenza?
6. Come è possibile esprimere il numero di condizionamento di un problema numerico?
7. Come possiamo controllare la propagazione degli errori di arrotondamento sul risultato di operazioni al calcolatore (e dunque su una simulazione numerica)?

Definizione (Stabilità di un metodo numerico)

Il problema numerico (8) *è ben posto (o* **stabile***) se per ogni* $h > 0$ *fissato esso ammette un'unica soluzione* $x_h \in \mathcal{X}_h$ *che dipende con continuità dai dati* $d_h \in \mathcal{D}_h$.

Definizione (Consistenza di un metodo numerico)

Se il dato $d \in \mathcal{D}$ *del PM è ammissibile per* $F_h(\cdot;\cdot)$ *(ovvero* $d \in \mathcal{D}_h$*), il metodo numerico* $F_h(x_h; d_h) = 0$ *è* **consistente** *se*

$$F_h(x; d) = F_h(x; d) - F(x; d) \to 0 \qquad \text{per } h \longrightarrow 0,$$

Un metodo è detto **fortemente consistente** *se* $F_h(x; d) = 0 \ \forall h > 0$ *e non solo per* $h \to 0$.

Definizione (Convergenza di un metodo numerico)

Il metodo numerico $F_h(_h; d_h) = 0$ *è* **convergente** *se l'errore*

$$e_c = e_c(x_h) := \|x - x_h\|$$

è tale che $e_c \to 0$ *per* $h \to 0$. *Se l'errore* e_c *può essere limitato in funzione di* h *come*

$e_c = e_c(x_h) \leq C\, h^p$, *per qualche* $p > 0$ *e* C *independente da* h *e* p, *il metodo numerico si dice convergente di ordine* p.

- Se il problema matematico (7) è ben posto, una condizione *necessaria* affinché il problema numerico (8) sia convergente è che esso sia stabile.

$$\text{convergenza} \quad \Rightarrow \quad \text{stabilità}$$

- La stabilità di un metodo numerico diventa condizione *sufficiente* per la convergenza se il problema numerico (8) è anche consistente con il problema matematico.

$$\text{stabilità} \quad \Rightarrow \quad \text{convergenza} \quad \text{(se PN consistente con PM)}$$

Definizione

Il numero di condizionamento (relativo) di un problema numerico $F_h(x_h; d_h) = 0$ *è*

$$K_h(d_h) := \sup \left\{ \frac{\|\delta x_h\|/\|x_h\|}{\|\delta d_h\|/\|d_h\|} \quad \forall \delta d_h \ : \ d_h + \delta d_h \in \mathcal{D}_h \right\}.$$

Se $K_h(d_h)$ *è* piccolo *il problema numerico* (8) *è ben condizionato; viceversa, il problema è mal condizionato.*

Spesso ci interesserà derivare **stime dell'errore** della forma

$$\|x - x_h\| \leq \Phi(x_h) \qquad \text{(stime a posteriori)}$$

per un'opportuna funzione Φ che può dipendere anche da h e K_h. A volte è possibile derivare stime a priori, in cui lo stimatore non dipende dalla soluzione numerica x_h.

3. Rappresentazione macchina dei numeri reali

1. Numeri floating-point
2. Aritmetica floating-point
3. Effetti degli errori di arrotondamento

Numeri Floating-point

Un calcolatore può gestire solo un numero finito di numeri ed effettuare un numero finito di operazioni: l'insieme dei numeri reali $\mathbb{R}$ è dunque rappresentato da un insieme *finito* di numeri macchina $\mathbb{F} = \{-\tilde{a}_{min}, \ldots, \tilde{a}_{max}\}$, detti numeri floating point[3]

$$\text{es.: } 1/3 = 0.\overline{3} \longrightarrow 0.33333333333333333333.$$

Ogni numero reale $x \in \mathbb{R}$ è dunque rappresentato da un numero arrotondato $fl(x) \in \mathbb{F}$.

Un numero floating point $x \in \mathbb{F}(\beta, t, L, U)$ viene rappresentato come:

$$x = (-1)^s \cdot m \cdot \beta^{e-t},$$

dove:

- $s = \{0, 1\}$ è il suo segno;
- $\beta \geq 2$ è un numero intero positivo, detto base;
- $m = a_1 a_2 \ldots a_t$ è la mantissa $(0 < m < \beta^t - 1)$ la cui lunghezza t è il numero massimo di cifre a_i: $0 \leq a_i \leq \beta - 1$, con $a_1 \neq 0$, che vengono memorizzate (anche dette *cifre significative*);
- e è l'esponente, tale che $e \in [L, U]$, con $L < 0$ e $U > 0$.

[3] La posizione della virgola non è infatti fissata.

Il numero di cifre dell'esponente definisce il range dei numeri macchina, il numero di cifre della mantissa la sua precisione.

Poiché

$$0.a_1 a_2 \dots a_t = a_1 a_2 \dots a_t \cdot \beta^{-t}$$

una rappresentazione equivalente dei numeri floating point è data da

$$x = (-1)^s \beta^e \sum_{i=1}^{t} a_i \beta^{-i}$$

Per rendere unica la rappresentazione di un numero macchina $fl(x)$ – osserviamo che $1.2345 \cdot 10^3 = 0.0012345 \cdot 10^6$ – richiediamo che (numeri normalizzati)

$$a_1 \neq 0, \qquad m \geq \beta^{t-1}.$$

Si osservi che $0 \notin \mathbb{F}$ (poiché $a_1 = 0$) e dunque occorre gestirlo separatamente.

Esempi

- $-3.154 \cdot 10^5 = -0.3154 \cdot 10^6 = (-1)^1 \cdot 10^6 \left(\frac{3}{10^1} + \frac{1}{10^2} + \frac{5}{10^3} + \frac{4}{10^4}\right)$
  ovvero $s = 1$, $\beta = 10$, $m = 3154$, $e = 6$, $t = 4$
- $(4.25)_{10} = (100.01)_2 = 1.0001 \cdot 2^2 = 0.10001 \cdot 2^3$
  ovvero $s = 0$, $\beta = 2$, $m = 10001$, $e = 3_{10} = 11_2$, $t = 5$

Nel caso dell'insieme $\mathbb{F}(\beta, t, L, U)$:

- l'errore di arrotondamento (roundoff) è l'errore relativo che si commette sostituendo $x \in \mathbb{R} \setminus \{0\}$ con il suo rappresentante $fl(x) \in \mathbb{F}$ è limitato da

$$\frac{|x - fl(x)|}{|x|} \leq \frac{1}{2}\varepsilon_M, \qquad x \neq 0,$$

  essendo $\frac{1}{2}\varepsilon_M$ l'*unità di arrotondamento* (ovvero il massimo errore relativo che la macchina può commettere nella rappresentazione di un numero).

- Poiché L ed U sono finiti, non è possibile rappresentare numeri in valore assoluto arbitrariamente piccoli o grandi. Il più piccolo e il più grande numero positivo sono

$$x_{min} = \beta^{L-1}, \qquad x_{max} = \beta^U \left(1 - \beta^{-t}\right);$$

- non è quindi possibile rappresentare alcun numero (a parte lo zero) minore in valore assoluto di x_{min}, o maggiore di x_{max};

- il valore detto epsilon macchina

$$\varepsilon_M = \beta^{1-t}$$

  rappresenta la distanza tra 1 e il più piccolo numero floating point maggiore di 1, ovvero il più piccolo numero reale maggiore di zero tale per cui $fl(1 + \varepsilon_M) > 1$;

- la cardinalità di $\mathbb{F}(\beta, t, L, U)$ (considerando solo i positivi), escluso lo zero, è data da

$$card\,(\mathbb{F}) = 2(\beta - 1)\beta^{t-1}(U - L + 1).$$

Mostriamo ad esempio che, per i numeri $\mathbb{F}(\beta, t, L, U)$ rappresentati nella forma

$$x = (-1)^s \cdot m \cdot \beta^{e-t} = (-1)^s \beta^e \sum_{i=1}^{t} a_i \beta^{-i}$$

si ha che:

- $x_{min} = \beta^{L-1}$.
  Scegliendo $s = 0$, $e = L$, $a_1 = 1$, $a_2 = \ldots = a_t = 0$, si trova

  $$x_{min} = (-1)^0 \beta^L \sum_{i=1}^{t} a_i \beta^{-i} = \beta^L \cdot \beta^{-1} = \beta^{L-1};$$

  analogamente, la mantissa sarebbe stata pari a $m = 10\ldots0 = \beta^{t-1}$.
- $x_{max} = \beta^U \left(1 - \beta^{-t}\right)$.
  Scegliendo $s = 0$, $e = U$, $a_1 = \beta - 1$, $a_2 = \ldots = a_t = \beta - 1$, si trova

  $$x_{max} = (-1)^0 \beta^U \sum_{i=1}^{t} (\beta - 1)\beta^{-i} = \beta^U (\beta - 1) \sum_{i=1}^{t} (1/\beta)^i;$$

  poiché (con $x \in (0, 1)$)

  $$\sum_{i=0}^{t} x^i = \frac{1 - x^{t+1}}{1 - x} \quad \Rightarrow \sum_{i=1}^{t} x^k = \frac{x}{1 - x}(1 - x^t)$$

  risulta (scegliendo $x = 1/\beta$) $x_{max} = \beta^U (\beta - 1) \frac{1/\beta}{1 - 1/\beta}(1 - \beta^{-t}) = \beta^U (1 - \beta^{-t})$.

- $\varepsilon_M = \beta^{1-t}$.

  Osserviamo che otteniamo 1 con $s = 0$, $e = 1$, $a_1 = 1$, $a_2 = \ldots = a_t = 0$:

$$1 = (-1)^0 \beta^1 \sum_{i=1}^{t} a_i \beta^{-i} = (-1)^0 \beta^1 \beta^{-1}$$

  il più piccolo numero floating point maggiore di 1 risulta dato da $s = 0$, $e = 1$, $a_1 = 1$, $a_2 = \ldots = a_{t-1} = 0$, $a_t = 1$. Di conseguenza,

$$1 + \varepsilon_M = (-1)^0 \beta^1 \sum_{i=1}^{t} a_i \beta^{-i} = (-1)^0 \beta^1 (\beta^{-1} + \beta^{-t}) = \beta^{1-t} + 1$$

  da cui

$$\varepsilon_M = 1 + \varepsilon_M - 1 = 1 + \beta^{1-t} - 1 = \beta^{1-t}.$$

Il caso $\beta = 2$ (numeri in base 2)

Poiché in base $\beta = 2$ necessariamente $a_1 = 1$, in questo caso non serve un bit per a_1, dato che tale cifra è nota a priori; occorrerà semplicemente memorizzare nella mantissa la frazione $\tilde{m} < 1$ del numero $1 + \tilde{m}$

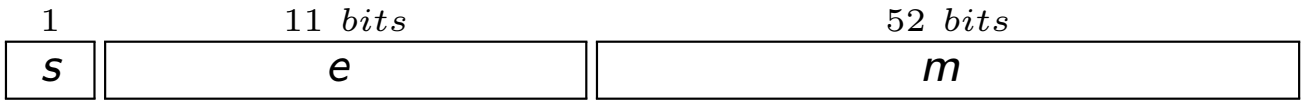
In base $\beta = 2$ si utilizza dunque la rappresentazione seguente[4]:

$$x = (-1)^s 2^e (1 + \tilde{m}), \qquad \tilde{m} = a_2 2^{-1} + a_3 2^{-2} + \ldots + a_{t+1} 2^{-t}$$

e la normalizzazione del numero avviene dunque sulla cifra delle unità, essendo

$$1 + \tilde{m} = \underset{=1}{a_1 2^{-0}} + a_2 2^{-1} + a_3 2^{-2} + \ldots + a_{t+1} 2^{-t}$$

Lo standard floating point **IEEE double precision** (usato in Matlab) usa una stringa di $N = 64$ bits per rappresentare i numeri macchina e corrisponde a $\mathbb{F} = \mathbb{F}(2, 52, -1022, 1023)$:



dove in m si memorizzano le cifre $a_2, \ldots, a_{53}$, che corrispondono alla parte frazionaria $\tilde{m} \in [0, 1)$ del numero.

[4] Si ha $1.m = 1 + \tilde{m}$, ovvero $\tilde{m} = m \cdot 2^{-t}$

Si ha quindi, nel caso in cui si usino 52 bit per $\tilde{m}$ e 11 bit per e,

$$1 = (-1)^0 2^0 (1 + 00000...000) \quad \text{ovvero} \quad s = 0, e = 0, a_2 = \ldots = a_{53} = 0$$

Il più piccolo dei numeri macchina maggiori di 1 è invece dato da

$$(-1)^0 2^0 (1 + 00000...001) \quad \text{ovvero} \quad s = 0, e = 0, a_2 = \ldots = a_{52} = 0, a_{53} = 1$$

Il più piccolo numero rappresentabile si ottiene invece per

$$(-1)^0 2^{-(11\ldots1)_2} (1 + 0) \text{ovvero} \quad s = 0, e = -(2^{10} - 1), a_2 = \ldots = a_{53} = 0.$$

Inoltre, per non memorizzare il segno dell'esponente, si memorizza in c l'esponente e maggiorato di 1023: in questo modo $0 < c < 2047 = 2^{11} - 1$. Si ha dunque

$$x = (-1)^s \cdot 2^{c-1023} \cdot (1.m)_2$$

Esempio

0 100 0000 0001 1010 0000 0000 0000

coincide con

$$+1 \cdot 2^{1025-1023} \cdot (1 + \tfrac{1}{2^1} + \tfrac{1}{2^3}) = 6.5.$$

Nel caso della base $\beta = 2$, per effetto della normalizzazione sulla cifra delle unità, memorizzando nella mantissa solo la parte frazionaria $\tilde{m} \in [0, 1)$ del numero si otterrebbe

$$x_{min} = \beta^L, \qquad x_{max} = \beta^U(\beta - \beta^{-t})$$

$$\varepsilon_M = \beta^{-t}$$

Esempio (versione alternativa dell'Esempio 5.3)
Si consideri l'insieme dei numeri floating point $\mathbb{F}(2, 1, -1, 2)$, cioè tali per cui $\beta = 2$, $t = 1$ (numero di cifre della mantissa), $L = -1$ e $U = 2$. Di conseguenza, si ha

$$\varepsilon = \beta^{-t} = 1/2$$
$$x_{min} = \beta^L = (1.0)_2 \cdot 2^{-1} = 2^{-1}$$
$$x_{max} = \beta^U(\beta - \beta^{-t}) = (1.1)_2 \cdot 2^2 = 2^2(2 - 2^{-1}) = 6$$
$$card(\mathbb{F}) = 2(\beta - 1)\beta^{t-1}(U - L + 1) = 16$$

- I valori che l'esponente può assumere sono -1, 0, 1 e 2.
- I numeri reali positivi in $\mathbb{F}$ sono pertanto rappresentabili come

$$x = (1 + \tilde{m})2^e, \qquad \tilde{m} = (a_2)_2 \cdot 2^{-1}$$

e sono riportati nella tabella seguente:

e	-1	0	1	2
$1 + \tilde{m} = 1 + 0 \cdot 2^{-1} = 1$	1/2	1	2	4
$1 + \tilde{m} = 1 + 1 \cdot 2^{-1} = 3/2$	3/4	3/2	3	6

Esempio ($\mathbb{F}(2, 52, -1022, 1023)$)

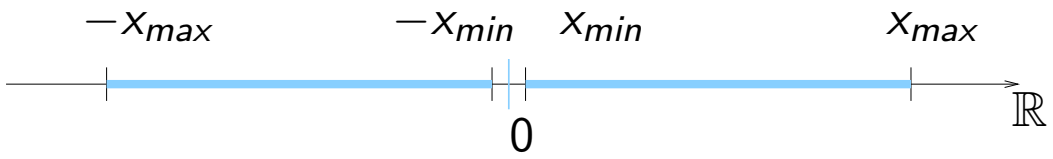
In MATLAB si ha $t = 52$ *e* $\varepsilon_M = 2^{-52} \approx 2.22 \cdot 10^{-16}$*; inoltre, usando i comandi* `realmin` *e* `realmax`*, si ha*

$$\begin{aligned} x_{min} &= (1.000\ldots000)_2 \cdot 2^{-1022} = 2.225073858507201 \cdot 10^{-308} \\ x_{max} &= (1.111\ldots111)_2 \cdot 2^{1023} = 2^{1023}(2 - 2^{-52}) = 1.797693134862316 \cdot 10^{+308}; \end{aligned}$$

- *53 cifre significative in base 2 corrispondono alle 15 cifre significative in base 10 nel* `format long` *di Matlab.*
- *un numero positivo maggiore di* x_{max} *produce una segnalazione di* overflow *e viene memorizzato nella variabile* `Inf` *.* `Inf` *è rappresentato prendendo* $c = 2047$ *e* $m = 0$*; soddisfa le relazioni* `1/Inf = 0` *e* `Inf+Inf = Inf`*.;*
- *analogamente, dà overflow un numero negativo minore di* $-x_{max}$*, memorizzato come* `-Inf`*;*
- *un numero positivo minore di* x_{min} *(o maggiore di* $-x_{min}$*) produce una segnalazione di* underflow *e viene trattato come 0;*
- *se un'operazione cerca di produrre un valore che non è definito nemmeno in* $\mathbb{R}$ *(come ad esempio* `0/0` *e* `Inf-Inf`*) il risultato è Not-a-Number,* `NaN`*, rappresentato prendendo* $c = 2047$ *e* $m \neq 0$*.*

I numeri floating-point non sono equispaziati, ma si addensano in prossimità del più piccolo numero rappresentabile.

$$[-x_{max}, -x_{min}] \cup \{0\} \cup [x_{min}, x_{max}]$$



Esempio

I numeri di $\mathbb{F}$ sono molto addensati vicino a x_{min} diventando sempre più radi all'avvicinarsi a x_{max}. Infatti, il numero di $\mathbb{F}$ immediatamente precedente x_{max} e immediatamente successivo a x_{min} sono rispettivamente

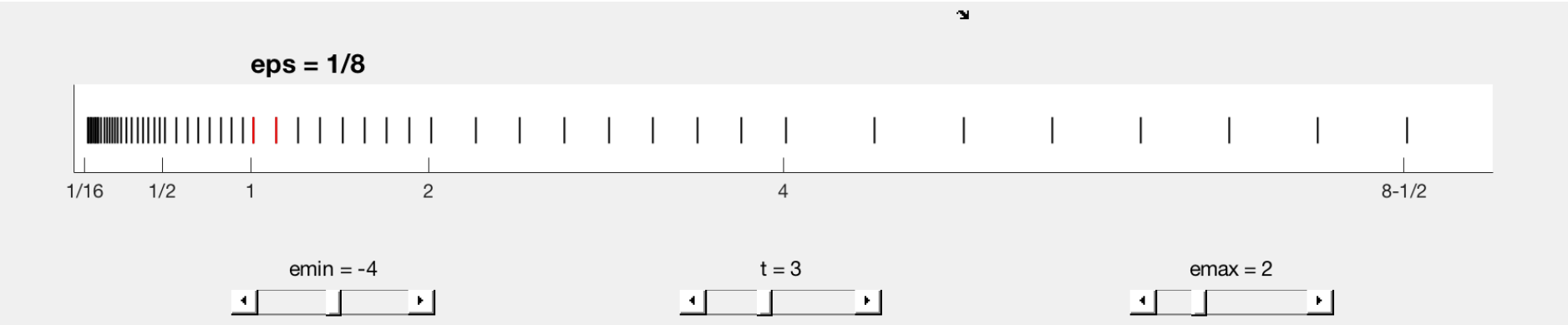
$$x^-_{max} = 1.797693134862315 \cdot 10^{+308}$$
$$x^+_{min} = 2.225073858507202 \cdot 10^{-308} :$$

$x^+_{min} - x_{min} \approx 10^{-323}$, $x_{max} - x^-_{max} \approx 10^{292}$ *(la distanza relativa è comunque piccola).*

Distribuzione dei numeri positivi in un sistema floating-point modello

Supponiamo di usare $\mathbb{F}(2, 3, -4, 2)$ ovvero

- $t = 3$ bit per rappresentare la mantissa m
- $-4 \leq e \leq 2$ per il range dell'esponente
- base 2



Si ha che in ogni intervallo della forma

$$2^e \leq x \leq 2^{e+1}$$

i numeri sono equispaziati, con un incremento di 2^{e-t}; ad esempio, se $e = 0$ e $t = 3$, la distanza tra i numeri compresi nell'intervallo $[1, 2]$ è $1/8$ (con double precision, questa spaziatura sarebbe 2^{-52}). Si noti che al crescere di e la spaziatura tra i numeri aumenta.

In questo caso $\varepsilon = \beta^{-t} = 2^{-3} = 1/8$ e

$$x_{min} = \beta^L = (1.000)_2 \cdot 2^{-4} = 2^{-4}$$

$$x_{max} = \beta^U(\beta - \beta^{-t}) = (1.111)_2 \cdot 2^2 = 2^2(2 - 2^{-3}) = 4(2 - 1/8) = 8 - 1/2.$$

Una nota sui numeri denormalizzati

Lo standard IEEE mette a disposizione anche *numeri denormalizzati* della forma

$$\tilde{x} = (-1)^s \cdot 2^{-1022} \cdot (0.m)_2$$

I numeri denormalizzati positivi appartengono all'intervallo `[realmin * eps, realmin]`.

Se un'operazione produce un numero strettamente positivo minore di `realmin * eps`, il risultato si trova nel cosiddetto range di underflow; siccome tale risultato non può essere rappresentato, ad esso viene assegnato valore zero.

Il calcolo con numeri denormalizzati è tuttavia da evitare in quanto può comportare significative perdite di precisione. Si consideri ad esempio il seguente codice:

```
res=pi*realmin/123456789101112
>> res = 5.68175492717433e-322
res2=res*123456789101112/realmin
>> res2 = 3.15248510554597
>> pi = 3.14159265358979
```

Il primo risultato è un numero denormalizzato, e non può essere più rappresentato con la miglior accuratezza possibile. Quando invertiamo le due operazioni e calcoliamo `res2`, otteniamo un risultato che contiene solo due cifre significative corrette.

In aritmetica floating point, intenderemo d'ora in poi di avere a che fare con numeri in formato double precision, secondo lo standard IEEE, normalizzati.

Come si opera con i numeri floating-point

Essendo $\mathbb{F}$ soltanto un sottoinsieme finito e discreto di $\mathbb{R}$, esse non godono di tutte le proprietà delle analoghe operazioni definite su $\mathbb{R}$. Ad esempio:

- la proprietà commutativa resta valida per addizione e moltiplicazione, ovvero $fl(x+y) = fl(y+x)$ e $fl(xy) = fl(yx)$
- la proprietà associativa e quella distributiva vengono violate
- non vale l'unicità dello 0: esiste infatti almeno un numero b diverso da 0 tale che $a + b = a$ ($a + b$ è in ogni caso uguale ad a se $b < \varepsilon_M$).

Esempio

L'associatività è violata quando si presenta una situazione di overflow o di underflow: prendiamo ad esempio $a = 1.0e+308$, $b = 1.1e+308$ *e* $c = -1.001e+308$; *troviamo*

$$a + (b + c) = 1.0990e+308, \qquad (a + b) + c = \texttt{Inf}.$$

*Come abbiamo già visto, ciò è quanto accade quando si sommano tra loro numeri che hanno all'incirca lo stesso modulo, ma segno opposto: il risultato della somma può essere assai impreciso (*cancellazione di cifre significative*).*

Ad esempio, calcolando $((1 + x) - 1)/x$ *(il risultato è 1* $\forall x \neq 0$*) con* `x = 1.e-15` *si ha* $y = fl((1 + fl(x)) - 1)/fl(x) = 1.1102$ *– e l'errore relativo è maggiore dell'*11%.

Effetti dell'arrotondamento (roundoff)

Effetti degli errori di arrotondamento / 1

```
% Esempio 1
a = 0.0;
for i = 1:10
  a = a + 0.1;
end
a == 1

ans =
  logical
   0
```

```
% Esempio 2
e = 1 - 3*(4/3 - 1)

e =
   2.2204e-16
```

```
% Esempio 3
b = 1e-16 + 1 - 1e-16;
c = 1e-16 - 1e-16 + 1;
b == c

ans =
  logical
   0
```

```
% Esempio 4 (cancellation)
sqrt(1e-16 + 1) - 1

ans =
     0
```

Effetti degli errori di arrotondamento / 2

Consideriamo le due equazioni

$$\begin{aligned} 17x_1 + 5x_2 &= 22 \\ 1.7x_1 + 0.5x_2 &= 2.2 \end{aligned}$$

Tale sistema ammette infinite soluzioni (risulta infatti singolare). Le istruzioni Matlab

```
A = [17 5; 1.7 0.5]
b = [22; 2.2]
```

forniscono invece

```
x= -1.0588
    8.0000
```

Perché? Il sistema è singolare, ma le equazioni sono consistenti tra loro (la seconda equazione risulta data da 0.1 volte la prima). La rappresentazione float di A non è singolare perché `A(2,1)` non è esattamente $17/10$ e quindi, per Matlab,

$$det(A) = 9.4369 \cdot 10^{-16} \neq 0.$$

Nella soluzione del sistema (metodo di eliminazione di Gauss), si sottrae dalla seconda un multiplo della prima; il moltiplicatore è `mu = 1.7/17`, numero float ottenuto troncando, anziché arrotondando, l'espansione di $1/10$ in base 2. `A` e `b` sono modificati prendendo

```
A(2,:) = A(2,:) - mu*A(1,:)
b(2) = b(2) - mu*b(1)
```

In aritmetica esatta, sia `A(2,2)` che `b(2)` sarebbero zero; in aritmetica floating point, diventano entrambi multipli non nulli di `eps`.

```
A(2,2) = (1/4)*eps
       = 5.5511e-17
b(2) = 2*eps
     = 4.4408e-16
```

Matlab riconosce il valore molto piccolo di `A(2,2)` e fornisce un messaggio di warning: `the matrix is close to singular`. Calcola quindi la soluzione della seconda equazione modificata dividendo un errore di roundoff per l'altro

```
x(2) = b(2)/A(2,2) =8
```

Questo valore viene poi sostituito nella prima equazione a dare

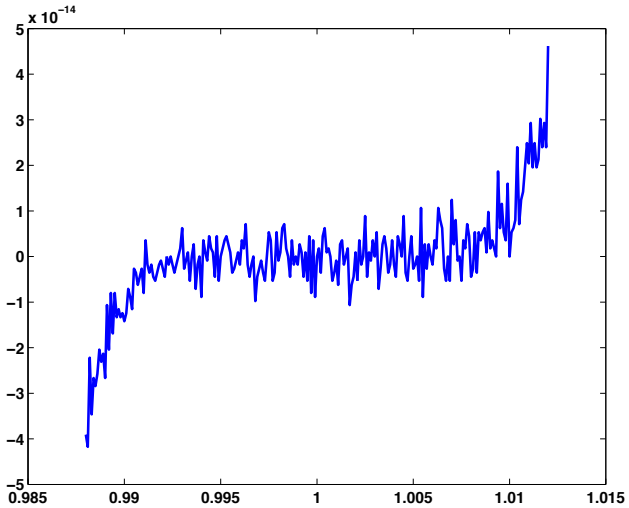
```
x(1) = (22 - 5*x(2))/17
     = -1.0588
```

Effetti degli errori di arrotondamento / 3

Proviamo a valutare nell'intervallo $[0.988, 1.012]$ la funzione

$$(x-1)^7 = x^7 - 7x^6 + 21x^5 - 35x^4 + 35x^3 - 21x^2 + 7x - 1.$$

```
x = 0.988:.0001:1.012;
y = x.^7-7*x.^6+21*x.^5-35*x.^4+35*x.^3-21*x.^2+7*x-1;
plot(x,y)
```



Il risultato è molto distante dall'essere un polinomio, a causa dell'errore di roundoff, e in particolare degli errori di *cancellazione* dovuti alle sottrazioni, dal momento che i piccoli valori di y sono ottenuti operando somme e differenze di numeri dell'ordine di $35 \cdot 1.012^4$. Se gli stessi valori fossero stati calcolati prendendo

```
y = (x-1).^7;
```

otterremmo il grafico di una funzione sostanzialmente nulla e costante nell'intorno di 1.

Gli errori di arrotondamento sono generalmente piccoli, tuttavia, se ripetuti all'interno di algoritmi lunghi e complessi, possono avere effetti catastrofici. Due casi eclatanti:

- esplosione del missile Ariane[5] (4 giugno 1996), causata dalla comparsa di un overflow nel computer di bordo
- mancata intercettazione di un missile Scud[6], (1 guerra del Golfo, 1991) e conseguente caduta su una caserma americana a causa di un errore di arrotondamento nel calcolo della traiettoria di un missile Patriot usato come sistema antimissile.

[5] ARIANE 5 / Flight 501 Failure. Report by the Inquiry Board (Chairman: Prof. J. L. LIONS) [originally appeared at http://www.esrin.esa.it/htdocs/tidc/Press/Press96/ariane5rep.html

[6] Report of the General Accounting office, GAO/IMTEC-92-26, entitled Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia

- missile Ariane

  il guasto alla base del fallimento fu dovuto ad una erronea conversione di un numero floating-point a 64 bit (correlato alla velocità orizzontale del razzo) in un intero a 16 bit: il numero risultante, maggiore del massimo intero rappresentabile (32768), provocò un overflow nella misura della velocità orizzontale con il conseguente spegnimento dei razzi ed esplosione del missile.

- missile Scud

  Il tempo, calcolato (24 bit, virgola fissa) in decimi di secondo nel computer della batteria del sistema antimissile Patriot, veniva poi moltiplicato per 10 per ottenere un tempo in secondi. Troncando e moltiplicando per 10 il numero 0.1 comportava l'insorgere di significativi errori[7] (0.000000095).

  Al momento dell'intercettazione la batteria, restata in funzione per 100 ore, aveva un tempo errato di circa 0.34 secondi, più che sufficiente a far mancare il bersaglio (la velocità di crociera di uno Scud è di circa 1676 m/s).

  Inoltre, nell'eseguire l'upgrade a sistema anti-missile, l'errore era stato eliminato da certe parti del codice e non da altre; questo contribuì in modo significativo al problema, evitando che le cancellazioni potessero compensare l'errore introdotto.

[7] Digitando ad esempio `t = 0.1` il valore memorizzato in `t` non è esattamente 0.1 perché esprimere tale frazione in binario richiede... una serie infinita! Infatti

$$\frac{1}{10} = \frac{1}{2^4} + \frac{1}{2^5} + \frac{0}{2^6} + \frac{0}{2^7} + \frac{1}{2^8} + \frac{1}{2^9} + \frac{0}{2^{10}} + \frac{0}{2^{11}} + \frac{1}{2^{12}} + \dots$$

Scelta di un metodo numerico

La scelta di un metodo numerico (PN) per approssimare la soluzione x di un PM deve tenere in considerazione le:

- proprietà matematiche del PM;
- l'efficenza computazionale in termini di:
  1. ordine di convergenza atteso dell'errore,
  2. i flops (= floating point operations) coinvolti nel calcolo,
  3. le prestazioni della CPU installata sul calcolatore,
  4. le modalità di accesso e la disponibilità della memoria di calcolo.

Supponiamo di indicare con n la dimensione del PN. Il numero di operazioni richieste dal calcolo della soluzione del PN può dipendere da n come segue:

dipendenza flops	$O(1)$ indipendente	$O(n)$ lineare	$O(n^\gamma)$ polinomiale	$O(\gamma^n)$ esponenziale	$O(n!)$ fattoriale

Esempio

Da un punto di vista teorico, se la matrice $A \in \mathbb{R}^{n\times n}$ è non singolare, la soluzione $\mathbf{x} \in \mathbb{R}^n$ del sistema lineare

$$A\mathbf{x} = \mathbf{b}$$

si può ottenere applicando la regola di Cramer:

$$x_i = \frac{\det(B_i)}{\det(A)},\ i = 1, \ldots, n,$$

dove $B_i \in \mathbb{R}^{n\times n}$ è la matrice ottenuta sostituendo la i-esima colonna di A con il vettore $\mathbf{b} \in \mathbb{R}^n$:

$$B_i = \begin{bmatrix} a_{11} & \ldots & b_1 & \ldots & a_{1n} \\ a_{21} & \ldots & b_2 & \ldots & a_{2n} \\ \vdots & & \vdots & & \vdots \\ a_{n1} & \ldots & b_n & \ldots & a_{nn} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \uparrow \\ i \end{matrix}$$

Sfortunatamente, la soluzione di un sistema lineare con questo metodo richiede

$$O(e(n+1)!) \text{ operazioni.}$$

Se $n = 100$, sono richieste $101!e \approx 2.56 \cdot 10^{160}$ operazioni aritmetiche.

Una macchina in grado di eseguire 10^{12} floating point operations (=flops) al secondo, cioè una potenza di calcolo di 1 TeraFlop, porterebbe a

$$\frac{2.56 \cdot 10^{160}}{10^{12}} = 2.56 \cdot 10^{148} \text{ secondi} \approx 8.11 \cdot 10^{140} \text{ anni}$$

In accordo con la teoria più accreditata, l'universo iniziò con il Big Bang circa $12.5(\pm) \cdot 10^9$ anni fa.

Occorrono dunque metodi più efficienti. La tecnica più celebre è il *metodo di eliminazione di Gauss*[8] che, come vedremo, richiede

$$O\left(\frac{2}{3}n^3\right) \text{ operazioni}$$

per risolvere un sistema lineare di dimensione n.

Un sistema di dimensione $n = 100$ viene dunque risolto in 10^{-6} secondi avendo a disposizione una potenza di calcolo di 1 TeraFlop, e in meno di un secondo su un qualsiasi laptop.

Dalla prossima lezione ci occuperemo di metodi numerici per risolvere sistemi lineari

[8] Carl Friedrich Gauss, 1777-1855; Theoria motus corporum coelestium in sectionibus conicis solem ambientium (1809)