Calcolo Numerico

Stefano Sabatini

Capitolo 1

Teoria degli errori

1.1 Introduzione

Per tradurre un problema in linguaggio matematico che sia processabile su un calcolatore, si richiedono insiemi finiti di dati in input ed in output.
Bisogna spesso fare un passaggio intermedio per passare ad un modello discreto, ad esempio un integrale definito di $f(x)$ va discretizzato.
Trovare una sequenza finita di operazioni che svolgono il compito (*metodo numerico* o algoritmo).
Per esempio per risolvere un sistema lineare si usa l'eliminazione gaussiana (richiede $\frac{n^3}{3}$ operazioni) oppure la regola di Cramer ($n!$, è molto lunga), per individuare il numero si tiene conto di moltiplicazioni e divisioni (pesano di più di addizioni e sottrazioni).

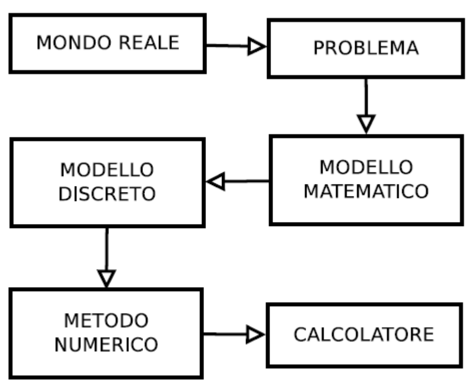

Figura 1.1: Schema della soluzione di un problema tramite calcolatore

1.2 Errori

Ci sono tre componenti di errore nel passaggio tra i vari passi del diagramma precedente, distinguibili dalle *sorgenti*. Il successivo diagramma invece specifica meglio gli ultimi due passi.

- Errori analitici: nascono nella formulazione del PROBLEMA, nella traduzione in MODELLO MATEMATICO o nella discretizzazione (MODELLO DISCRETO);
- Errori inerenti: sono errori di misurazione ad esempio, sono sempre presenti (DATI);
- Errori algoritmici: riguardano le approssimazioni che si adottano nell'algoritmo (RISULTATI INTERMEDI).

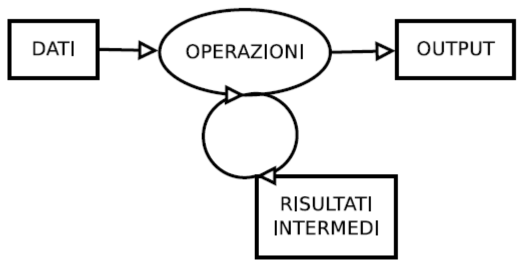

Figura 1.2: Schema di un algoritmo

Misure possibili:

- L'*errore assoluto* è la differenza fra valore ottenuto e valore atteso: $\delta = \tilde{x} - x$
- L'*errore relativo* invece è $\varepsilon = \frac{\tilde{x}-x}{x}$ ovvero $\tilde{x} = x\left(1+\varepsilon\right)$.

1.2.1 Errore inerente

Dati $x_1 ... x_n \in \mathbb{R}$ il problema ha come risultati attesi $y_1 ... y_m$. Il calcolatore usa la versione perturbata dei dati $\tilde{x}_1 ... \tilde{x}_n$, quindi $\tilde{x}_i = x_i\left(1+\varepsilon_i\right)$. Il valore atteso è $y_j = f_j\left(x_1 ... x_n\right)$ e il valore ottenuto è $\tilde{y}_j = f_j\left(\tilde{x}_1 ... \tilde{x}_n\right)$.
L'errore in output è $r_j = \frac{\tilde{y}_j - y_j}{y_j}$.
Nel caso particolare di un solo output abbiamo ε_i l'errore su x_i, l'errore sull'output $r = \frac{\tilde{y}-y}{y}$.
Consideriamo il *condizionamento del problema* $c = \frac{r}{\varepsilon_i}$, che effetto ha l'errore su un certo input sul valore finale? Se il rapporto è basso si dice che il problema è *ben condizionato*, quindi l'errore è dello stesso ordine di grandezza.

1.2.1.1 Esempio

$$\begin{cases} x + y = 2 \\ 1001x + 1000y = 2001 \end{cases}$$

L'input è costituito dalla matrice dei coefficienti $\begin{pmatrix} 1 & 1 \\ 1001 & 1000 \end{pmatrix}$ e dalla colonna dei termini noti $\begin{pmatrix} 2 \\ 2001 \end{pmatrix}$. Nella forma esatta (o attesa) l'output è $x = 1$, $y = 1$. Spostiamo un dato in input di poco, $\frac{1}{100}$ (sostituiamo a x nella prima equazione $\left(1 + \frac{1}{100}\right)x$), la soluzione si sposta di poco?

No, la soluzione esatta del sistema perturbato è $\tilde{x} = -\frac{1}{9}$, $\tilde{y} = \frac{1901}{900}$, l'errore relativo in output è $r = \frac{\frac{1901}{900}-1}{1} = \frac{1001}{900} > 1$. Il problema è mal condizionato (la soluzione è variata di tanto), si vede che $\varepsilon_i = \frac{1}{100}$ e il condizionamento vale $\frac{r}{\varepsilon_i} = \frac{1001}{900} \cdot 100$.

1.2.1.2 Formula per il condizionamento

Con un dato in input ed uno in output si puo' ricavare una formula per il condizionamento.
$x \longmapsto y = f(x)$ $\varepsilon = \frac{\tilde{x}-x}{x}$ $r = \frac{f(\tilde{x})-f(x)}{f(x)}$
$c = \frac{f(\tilde{x})-f(x)}{f(x)} \cdot \frac{x}{\tilde{x}-x}$

$$c \rightarrow \frac{xf'(x)}{f(x)} \text{ (\textit{coefficiente di amplificazione}) al limite per } \tilde{x} \rightarrow x$$

1.2.1.3 Esempio

$f(x) = 1 - \cos x \Longrightarrow f'(x) = \sin x$
$c = \frac{x \sin x}{1-\cos x}$
$\lim_{x\to 0} c = \lim_{x\to 0} \frac{x \sin x}{1-\cos x} = 2$ Con un piccolo errore mi devo aspettare un errore di circa 2ε sull'output.
Il condizionamento vero e proprio non è direttamente calcolabile, però possiamo fare una stima: nel caso di singola funzione l'abbiamo stimato ($c = \frac{xf'(x)}{f(x)}$) e quindi possiamo ricavare l'errore atteso sui risultati $r = c\varepsilon$; nel caso di più variabili in input, abbiamo $\frac{x_i \frac{\partial f}{\partial x_i}(x_1...x_n)}{f(x_1...x_n)}$, dove $\frac{\partial f}{\partial x_i}$ rappresenta la *derivata parziale*.

1.2.2 Rappresentazione dei numeri in un calcolatore

$B = 10$	$45.61 = 4 \cdot 10^1 + 5 \cdot 10^0 + 6 \cdot 10^{-1} + 1 \cdot 10^{-2}$
$B = 2$	$101.011 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$
$B = 16$	$5C.2F = 12 \cdot 16^0 + 5 \cdot 16^1 + 2 \cdot 16^{-1} + 15 \cdot 16^{-2}$

Tabella 1.1: Esempi di rappresentazione in diverse basi

1) Rappresentazione in *virgola fissa* (fixed point): $\sum_{i=-s}^{m} d_i B^i$ numeri del tipo $d_m...d_2 d_1 d_0 d_{-1} d_{-2}...d_{-s}$, se si presentano numeri fuori dal minimo/massimo rappresentabile si ha underflow/overflow.
2) Rappresentazione in *virgola mobile* (floating point): $0.d_1...d_r \cdot B^p$ $\forall i$ tale che $d_i \in \mathbb{N}$, $0 \leqslant d_i \leqslant B-1$, $p \in \mathbb{Z}$, $-m \leqslant p \leqslant M$ (altrimenti overflow o underflow), $d_1 \neq 0$ (quest'ultimo perchè per evitare ambiguità di rappresentazione la prima cifra decimale deve essere diversa da zero).

1.2.2.1 Numeri di macchina

Rappresentati dalla funzione

$$\Im(B,t,m,M) = \left\{ x \in \mathbb{R} : x = \pm f B^p, -m \leqslant p \leqslant M, f = \sum_{i=1}^{t} d_i B^{-i} \right\}$$

Per Matlab e per la doppia precisione del C ad esempio vale $\Im(2, 52, 1024, 1023)$: 1 bit per il segno, 11 per l'esponente (caratteristica) e 52 per la mantissa (f). Quando si converte un numero reale, esso diventa un numero di macchina.
Ci sono due strategie di conversione:
-*Troncamento* (Chopping): ignora le cifre che escono dallo spazio di rappresentazione.
-*Arrotondamento* (Rounding): controlla se la prima cifra fuori da t è minore o maggiore della metà della base, se è maggiore aggiungo una unità nell'ultima cifra disponibile.
Esempio: $B = 10$ $t = 2$ $x = 0.995$
Per troncamento diventa $\tilde{x} = 0.99 \cdot 10^0$
Per arrotondamento diventa $\tilde{x} = 0.99 + 0.01 = 1 = 0.10 \cdot 10^1$
Rappresentando un numero per arrotondamento, il *massimo errore assoluto* commesso è

$$|\tilde{x} - x| \leqslant \frac{B^{p-t}}{2}$$

L'*errore relativo assoluto* è definito *precisione di macchina*:

$$\varepsilon = \frac{\tilde{x} - x}{x} \Longrightarrow |\varepsilon| = \frac{|\tilde{x} - x|}{|x|} \leqslant \frac{1}{2}\frac{B^{p-t}}{B^{p-1}}$$

$|x| = fB^p \geqslant B^{-1}B^p$ La mantissa è minorata dal numero con tutti 0 tranne la prima cifra.

$$|\varepsilon| \leqslant \frac{1}{2}B^{-t+1} = u$$

E' una costante propria della macchina, ad ogni calcolo ci si attende sempre questo errore. Nel caso di MATLAB è $u = \frac{1}{2} \cdot 2^{-51} = 2^{-52} \cong 2.2 \cdot 10^{-16}$.
L'errore inerente atteso sarà come minimo $c \cdot u$.

1.2.2.2 Sulla precisione di macchina

Supponiamo che $\varepsilon > 0$ sia piccolo $\mapsto x = 1 + \varepsilon$ può essere rappresentato come $\tilde{x} = 1$?
Se $\varepsilon < u \Rightarrow \tilde{x} = 1$ e $\varepsilon_x = \frac{\tilde{x}-x}{x} = \frac{1-(1+\varepsilon)}{1+\varepsilon} = -\frac{\varepsilon}{1+\varepsilon}$, è possibile perché in modulo $< u$!
Se $\varepsilon \geqslant u \Rightarrow \tilde{x} > 1$ perché ε_x sarebbe troppo grande.
Nel caso che ε sia minore della precisione di macchina, non viene sentito nella somma con 1.

1.2.2.3 Esempio

$B = 10,\ u = 5 \cdot 10^{-t} \quad 1 = 0.\underbrace{100...0}_{t} \cdot 10^1$
Supponiamo di sommare u ad 1 (dopo averlo opportunamente scalato).
$u = 0.\underbrace{0...0}_{t}5 \cdot 10^1$ 5 è la cifra subito oltre la scala, allora arrotonda per eccesso e scrive $\underbrace{0...01}_{t} \cdot 10^1$
Il calcolatore procede alla somma e ottiene $\tilde{x} = 0.10...01 \cdot 10^1$.
Se al posto di 5 ci fosse stato 4, avrebbe arrotondato a 0.
Questo è il più piccolo numero che viene "sentito" in questa operazione, da non confondere col più piccolo numero che viene rappresentato.

1.2.3 Errore algoritmico

Supponiamo di avere in input $x_1 ... x_n$ e prendiamo un algoritmo in cui ci siano errori nelle operazioni sui dati in input ed errore conseguente di arrotondamento sull'output.
Calcoliamo gli errori che viziano le quattro operazioni fondamentali

1.2.3.1 Addizione

$y = a + b \longrightarrow \tilde{y} = \left(\tilde{a} + \tilde{b}\right)(1 + \varepsilon_y)$
dove $\tilde{a} = a(1 + \varepsilon_a)$, $\tilde{b} = b(1 + \varepsilon_b)$ e $|\varepsilon_y| \leq u$ è l'errore relativo di rappresentazione
$\tilde{y} = (a(1 + \varepsilon_a) + b(1 + \varepsilon_b))(1 + \varepsilon_y)$
$\varepsilon_{a+b} = \frac{\tilde{y}-y}{y} = \frac{(a(1+\varepsilon_a)+b(1+\varepsilon_b))(1+\varepsilon_y)+(a+b)}{a+b} \doteq$
Per semplificare i conti faccio l'*analisi al primo ordine*, tralasciando dei pezzi che influiscono poco sul risultato, in particolare i prodotti fra errori.
$\doteq \varepsilon_y + \varepsilon_a \frac{a}{a+b} + \varepsilon_b \frac{b}{a+b}$

$$\varepsilon_{a+b} \doteq \varepsilon_y + \varepsilon_a \frac{a}{a+b} + \varepsilon_b \frac{b}{a+b}$$

Ad ogni somma c'è un errore che dipende da tre oggetti, gli errori, che non sono dello stesso ordine di grandezza. I rapporti $\frac{a}{a+b}$ e $\frac{b}{a+b}$ si chiamano *coefficienti di amplificazione*.
Consideriamo questi coefficienti:

- nel caso che a, b siano concordi, il coefficiente è minore o uguale ad 1 ($c_{amp} \leqslant 1$) e l'errore potrebbe ridursi (sommare numeri di segno concorde non è pericoloso);
- nel caso che a, b siano discordi e ad esempio $a \cong -b$ il coefficiente diventa molto grande ($c_{amp} \gg 1$), questo caso prende il nome di *cancellazione* ed è fortemente da evitare.

Esempio Dati $a = 0.123456$ e $b = -0.123454$, con $B = 10$ e $t = 6$ poniamoci nel caso di cancellazione.
$a + b = 0.000002 \longrightarrow 0.200000 \cdot 10^{-5}$
Gli zeri non sono affidabili (provengono da cifre fuori scala), l'errore è stato amplificato di circa 100000 volte!

1.2.3.2 Sottrazione

La sottrazione è analoga all'addizione, ma cambia il segno (si somma l'opposto di b)

$$\varepsilon_{a-b} \doteq \varepsilon_y + \varepsilon_a \frac{a}{a-b} - \varepsilon_b \frac{b}{a-b}$$

1.2.3.3 Moltiplicazione

$a, b \longmapsto y = ab$
$\tilde{y} = \tilde{a}\tilde{b}(1 + \varepsilon)$

$$\varepsilon_{ab} = \frac{\tilde{y} - y}{y} = \frac{\tilde{a}\tilde{b}(1+\varepsilon) - ab}{ab} \doteq \frac{ab(1 + \varepsilon_a + \varepsilon_b + \varepsilon) - ab}{ab} = \varepsilon_a + \varepsilon_b + \varepsilon$$

L'errore relativo nel fare una moltiplicazione è in prima approssimazione la somma degli errori sui fattori e dell'errore locale sulla moltiplicazione.

1.2.3.4 Divisione

$a, b \longmapsto y = \frac{a}{b}$
$\tilde{y} = \frac{\tilde{a}}{\tilde{b}}\left(1+\varepsilon\right) = \frac{\tilde{a}(1+\varepsilon_a)(1+\varepsilon)}{\tilde{b}(1+\varepsilon_b)} \doteq \frac{a(1+\varepsilon_a+\varepsilon-\varepsilon_b)}{b}$ (avendo moltiplicato e diviso $\frac{1}{1+\varepsilon_b}$ per $1-\varepsilon_b$)
$\varepsilon_{a/b} = \frac{\tilde{y}-y}{y} \doteq \frac{\frac{a}{b}(1+\varepsilon_a+\varepsilon-\varepsilon_b)-\frac{a}{b}}{\frac{a}{b}} = \varepsilon_a - \varepsilon_b + \varepsilon$
In conclusione l'unica cosa di cui preoccuparsi negli algoritmi contenenti le quattro operazioni elementari è se ci siano cancellazioni sommando o sottraendo.

1.2.3.5 Esempio 1

$$y = x^2 - 7x = f\left(x\right)$$

Algoritmo 1:

Algoritmo 1.1 Primo algoritmo dell'Esempio 1

$q = x^2$
$p = 7x$
$y_1 = q - p$

Algoritmo 2:

Algoritmo 1.2 Secondo algoritmo dell'Esempio 1

$d = x - 7$
$y_2 = xd$

$\varepsilon_{alg1} = \frac{\tilde{y}_1-y_1}{y_1} = \frac{(\tilde{q}-\tilde{p})(1+\varepsilon_1)-q+p}{q-p} = ... = \varepsilon_q \frac{x^2}{x^2-7x} - \varepsilon_p \frac{7x}{x^2-7x} + \varepsilon_1$
$\varepsilon_{alg2} = \frac{\tilde{y}_2-y_2}{y_2} = \frac{x\tilde{d}(1+\varepsilon_2)-xd}{xd} = ... = \varepsilon_d + \varepsilon_2$ (su x non metto un errore perchè eventualmente lo inserirò nell'errore inerente)
L'errore algoritmico nel secondo caso è basso, pero' l'errore inerente puo' dare errori alti!
$c = \frac{xf'(x)}{f(x)} = \frac{x(2x-7)}{x(x-7)} = \frac{2x-7}{x-7}$
Errore inerente: $\varepsilon_{in} = \frac{2x-7}{x-7}\varepsilon_x$
Bisogna studiare il coefficiente: finchè la x è lontana da 7 non ci sono grossi problemi, quando è vicina ivece possono nascere.
$x \approx 7 \rightarrow \varepsilon_{in} \approx \frac{7}{x-7}\varepsilon_x$
Definizione: Un algoritmo si dice *stabile* se l'errore algoritmico ha al massimo lo stesso ordine di grandezza dell'errore inerente (nell'esempio anche il primo algoritmo è stabile).
La cancellazione è intrinseca a questo problema, ma nel secondo algoritmo viene fatta subito.
Si puo' ricavare una regola empirica: le cancellazioni vanno evitate, ma se il problema è patologico vanno svolte il prima possibile per non amplificare altri errori.

1.2.3.6 Esempio 2

$$f\left(x\right) = 1 - \cos x$$

$x \approx 0$

$c = \frac{x \sin x}{1-\cos x} \approx 2\ \varepsilon_{in} \approx 2\varepsilon_x$
Le funzioni elementari (tipo seno e coseno) hanno un errore locale circa paragonabile alla precisione di macchina.
Algoritmo 1:

Algoritmo 1.3 Primo algoritmo dell'Esempio 2

$c = \cos x$
$y_1 = 1 - c$

$\varepsilon_{alg1} = \frac{\tilde{y}-y}{y} = \frac{(1-\tilde{c})(1+\varepsilon_1)-1+c}{1-c} = ... = \varepsilon_1 - \varepsilon_c \frac{\cos x}{1-\cos x}$
Con la cancellazione quando x è molto piccolo il risultato viene gonfiato. L'algoritmo è *instabile*.
Trasformiamo la funzione $f(x)$ moltiplicando e dividendo per $1 + \cos x$:

$$f(x) = \frac{\sin^2 x}{1 + \cos x}$$

Algoritmo 2:

Algoritmo 1.4 Secondo algoritmo dell'Esempio 2

$s = \sin x$
$c = \cos x$
$n = s^2$
$d = 1 + c$
$y_2 = \frac{n}{d}$

$\varepsilon_{alg2} = \varepsilon_d + \varepsilon_n$

1.2.3.7 Rappresentazione algoritmi tramite grafi

Si applica l'errore inerente a ciascun nodo in cui è contenuta la variabile, su ogni arco si mette come peso il coefficiente di amplificazione. Finita l'etichettatura si puo' calcolare l'errore algoritmico.
Legge generale:

$$\varepsilon_{alg} \doteq \left(\sum_{\varepsilon} \varepsilon \cdot \left(\prod \text{archi che seguono in un cammino} + \text{altri cammini}\right)\right)$$

Consideriamo l'esempio 1:

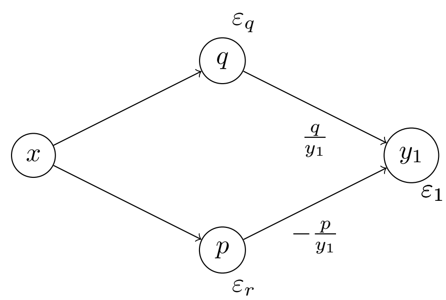

Figura 1.3: Grafo del primo algoritmo per l'esempio 1

$$\varepsilon_{alg1} = \varepsilon_q \frac{q}{y_1} + \varepsilon_p \left(-\frac{p}{y_1} \right) + \varepsilon_1$$

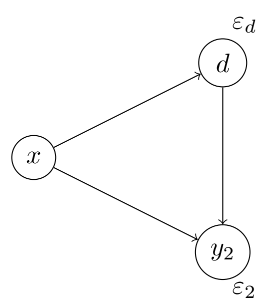

Figura 1.4: Grafo del secondo algoritmo per l'esempio 1

Consideriamo l'esempio 2:

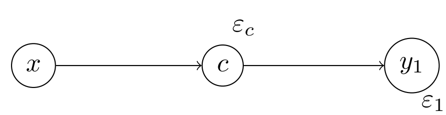

Figura 1.5: Grafo del primo algoritmo per l'esempio 2

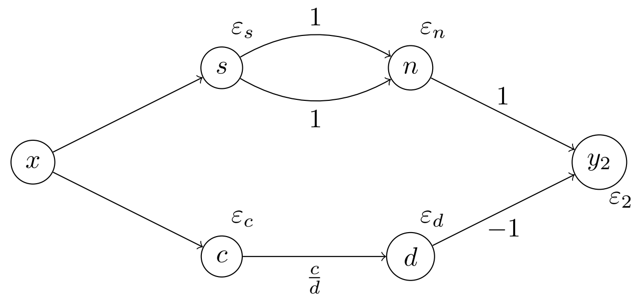

Figura 1.6: Grafo del secondo algoritmo per l'esempio 2

$$\varepsilon_{alg2} = \varepsilon_s (1 + 1) + \varepsilon_n + \varepsilon_2 + \varepsilon_c \frac{-c}{d} - \varepsilon_d$$

dove il coefficiente di ε_c è $\frac{-\cos x}{1+\cos x}$

1.3 Introduzione a Matlab

La precisione di macchina in Matlab è contenuta nella variabile *eps*, la risposta ad un comando invece in *ans*.

Algoritmo 1.5 Esercizi sugli errori

$s = 1 + 1e - 16$
R:$s = 1$ s è minore della precisione di macchina, non l'ha sentito
$s = 1 + 3e - 16$
R:$s = 1.000...0$ C'è qualcosa in fondo ma fuori dalla visualizzazione
$s - 1$
R:$2.2... \cdot e - 16$
$s2 = 1e5 + 1e - 11$
R:$1.00...0 \cdot e + 005$
$s2 - 1e5$
R:$1.45...0 \cdot 10 - 11$ L'ha sentito, questo è l'errore assoluto

Algoritmo 1.6 Disegno di grafici

$x = -2 : 0.01 : 2$ Crea e stampa a video un vettore di elementi tra -2 e 2 distanziati tra loro di 0.01.
$x = -2 : 0.01 : 2$; Aggiungendo il punto e virgola esegue senza stampare a video il risultato.
$y1 = x$; Funzione
$plot(x, y1,' : r')$ Disegna il grafico con una riga punteggiata (':') rossa ('r')
$y2 = 1.2 * x + 0.1$
$plot(x, y1,' : r', x, y2,' -b')$ Disegna un grafico come il precedente ed un'altro con una linea continua blu (se ne possono sovrapporre molti, tutti in questo formato)
hold on Istruzione che congela il grafico, permarrà anche se viene disegnato sopra altro
$plot(-0.5, -0.5,' o')$ Disegna un pallino
$y2p = 1.1 * x + 0.1$; Scriviamo la versione perturbata del secondo grafico
$plot(x, y2p,' -g')$ In verde stavolta
$hold\ off$ Disattiva la sovrapposizione dei grafici
clear Pulisce tutte le variabili

Algoritmo 1.7 Esercizio sulle cancellazioni

$x = 3 + eps$
$y = x + 3$
R:$y = 6$ Calcolo y con una diversa espressione...
$y2 = (x\hat{}2 - 9) / (x - 3)$Restituisce un errore: divisione per 0.
$x = 3 + 2 * eps$
Ricalcolo y e $y2$, nel primo caso non sente la differenza, nel secondo restituisce 8, colpa delle due cancellazioni presenti nell'algoritmo.
$x = 2.2$
$y = (x - 2)\hat{}2$
$y2 = x\hat{}2 + 4 - 4x$ (fa la cancellazione dopo)
$err = (y2 - y) / y$
R: $\sim -2.3 \cdot 10^{-14}$