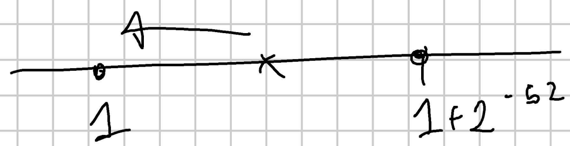


function y=pow(x,n)
% calcola x^n tramite n prodotti

function y=myexp(x,n)
% calcola e^x con la serie di Maclaurin troncata dopo $x^n/n!$

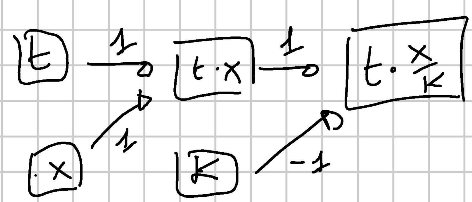
$$e^x = 1+x+\frac{x^2}{2}+\frac{x^3}{3!}+\dots+\frac{x^k}{k!}+\dots+\frac{x^n}{n!}$$

Idea per ridurre costo:

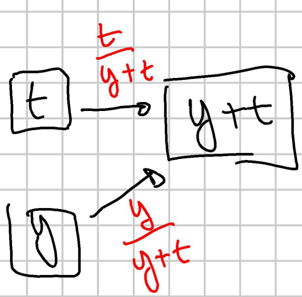
$$\frac{x^k}{k!} = \frac{x\cdot x\cdot x\cdots x}{1\cdot 2\cdot 3\cdots k} = \frac{x^{k-1}}{(k-1)!}\cdot\frac{x}{k}$$

Ciclo for semplice, risolve anche i problemi di overflow

Oss::



→ gli errori sul calcolo di t non crescono mai troppo, ogni termine è $|(\pm1)\cdot(\pm1)\cdot\ldots\cdot(\pm1)\cdot\varepsilon| \le |u|$



Instabilità quando i singoli termini sono molto più grandi della somma.

$$\text{myexp2}(10, 50) = \boxed{1 + 10 + \frac{10^2}{2} + \frac{10^3}{3!} + \dots + \frac{10^{50}}{50!}}$$

$\Downarrow$

somme di termini positivi

$\Rightarrow$ i coefficienti $\frac{y}{y+t}$, $\frac{t}{y+t}$ sono sempre ≤ 1.

Su $\text{myexp2}(-10, 50) = 1 + (-10) + \frac{(-10)^2}{2!} + \dots + \frac{(-10)^k}{k!} + \dots$

su un termine $t \approx 2 \cdot 10^3$, andandolo a scrivere in memoria ottengo $\tilde{t} = t(1+\varepsilon) = t + \underbrace{t\varepsilon}$

$$2 \cdot 10^3 \cdot 2 \cdot 10^{-16} \approx 10^{-13}$$

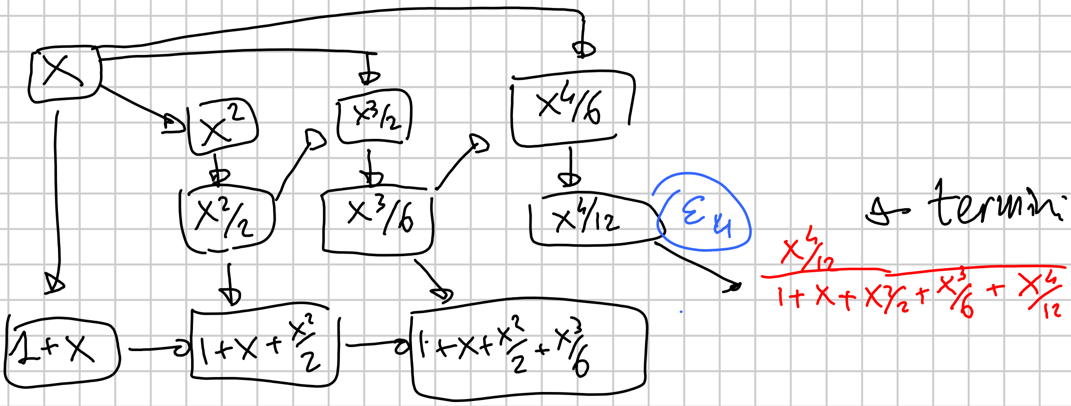
<u>Idea</u>: $e^{-30} = \frac{1}{e^{30}}$ e e^{30} si calcola senza perdita di precisione.

<u>Oss</u>: il numero di condizionamento dell'esponenziale $y = e^x$ è

$$K_{y,x} = \left| \frac{dy}{dx} \cdot \frac{x}{y} \right| = \left| \cancel{e^x} \frac{x}{\cancel{e^x}} \right| = |x|$$

L'errore inerente nel calcolo di e^x per $x = -30$ è

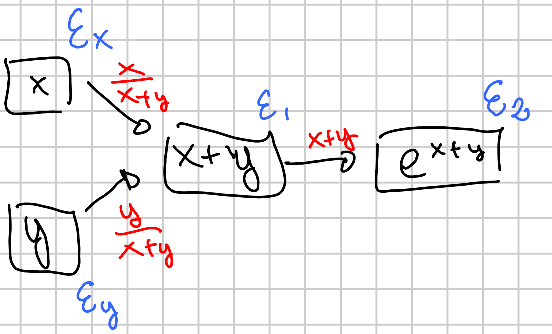
$e_{in} = x \cdot \varepsilon_x \qquad |e_{in}| \leq |x| \cdot u$



Es: Supponiamo di avere una funzione di libreria exp( ) che restituisce $e^x(1+\varepsilon)$ con un errore ε limitato da $C\cdot u$, dove C è una costante.

Vogliamo calcolare $f(x,y) = e^{x+y}$

con due possibili algoritmi: $\exp(x+y)$, $\exp(x)\exp(y)$



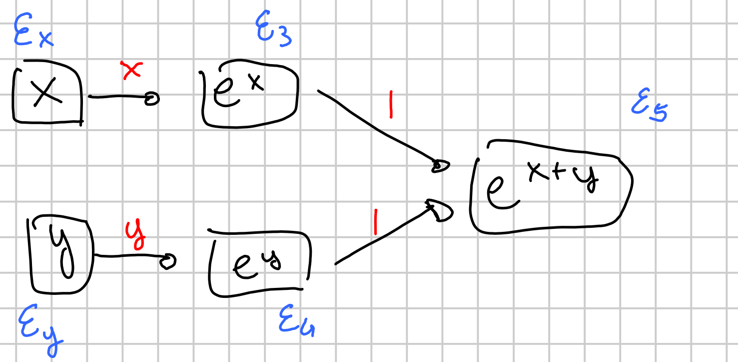
$|\varepsilon_x|, |\varepsilon_y|, |\varepsilon_1| \leq u$

$|\varepsilon_2| \leq Cu$

$$\varepsilon_{mec} = \varepsilon_x \cdot \frac{x}{x+y} \cdot \cancel{(x+y)} + \varepsilon_y \cdot \frac{y}{x+y}\cancel{(x+y)} + \varepsilon_1 (x+y) + \varepsilon_2$$

$$|\varepsilon_{mec}| \leq \underbrace{|x|\cdot u + |y|\cdot u}_{\varepsilon_{in} \leq |x|u+|y|u} + \underbrace{|x+y|u + u}_{\varepsilon_{alg} \leq |x+y|u+Cu}$$

$$|\varepsilon_{alg}| \leq (C+|x+y|)\cdot u$$



$|\varepsilon_x|, |\varepsilon_y|, |\varepsilon_5| \leq u$, $|\varepsilon_3|, |\varepsilon_4| \leq Cu$

$$\varepsilon_{mec} = \underbrace{\varepsilon_x \cdot x \cdot 1 + \varepsilon_y \cdot y \cdot 1}_{=\varepsilon_{in}\ \text{uguale all'altro}} + \underbrace{\varepsilon_3 \cdot 1 + \varepsilon_4 \cdot 1 + \varepsilon_5}_{=\varepsilon_{alg}\ \text{diverso}}$$

$$|\varepsilon_{mec}| \leq |x|\cdot u + |y|u + Cu + Cu + u$$

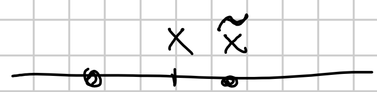
$$|\varepsilon_{alg}| \leq (2C+1)\cdot u$$

vince (sx) se $|x+y| \leq C+1$, (dx) altrimenti.

Entrambi gli algoritmi sono stabili? Sì: $|x+y|u$ è sempre minore o uguale a $|x|u+|y|u$ che è la maggiorazione dell'errore inerente.

tutor: quindi S alle ||·||S solo minore est (n f)

$$\varepsilon_x = \frac{\tilde{x} - x}{x}$$



$$\left|\frac{\tilde{x}-x}{x}\right| \leq u$$

$4+2^{-100}$

$u \approx 2^{-52}$