

***Maxima*: esempi d'uso**

Milla Lacchini

CRIAD (Centro di Ricerca Informatica applicata alla Didattica)
Università di Bologna sede di Cesena

Agosto 2003

(La trattazione è per ora incompleta ed esamina solo una parte delle numerose funzioni disponibili con *Maxima*)

[Calcolo simbolico: i principali comandi per semplificazione, fattorizzazione e sviluppo di potenze](#)

[Valutazione di espressioni](#)

[Equazioni](#)

[Sistemi di equazioni](#)

[Funzioni](#)

[Limiti](#)

[Derivate](#)

[Studio di funzione reale ad una variabile reale](#)

[Rappresentazione grafica di curve parametriche](#)

[Zeri di una funzione](#)

Calcolo simbolico: i principali comandi

Il comando **RATSIMP** effettua la semplificazione di espressioni razionali, i cui argomenti possono essere anche funzioni non razionali:

(C1) `ratsimp((sin(x-1))^4/sin(x-1));`
(D1) $\text{SIN}^3(x - 1)$

(C2) `ratsimp((x^4-1)/(x+1));`
(D2) $x^3 - x^2 + x - 1$

Il comando **FACTOR** effettua la fattorizzazione di una espressione purchè i coefficienti siano razionali:

(C3) `factor(%);`
(D3) $(x - 1)(x^2 + 1)$

Il comando `factor(%)`, in cui il simbolo % indica il contenuto della riga precedente, è equivalente a `factor(D2)`

(C3) `factor(x^2-4/9);`
(D3) $\frac{(3x - 2)(3x + 2)}{9}$

Si osservi che in questo caso è stata effettuata la riduzione allo stesso denominatore, e poi la scomposizione del numeratore nell'insieme degli interi. Si veda anche il seguente esempio:

(C4) `9/4*x^2 - 3*x*y + y^2;`
(D4) $y^2 - 3xy + \frac{9x^2}{4}$

L'operatore di moltiplicazione * non deve essere sottinteso; *Maxima* visualizza l'operazione di moltiplicazione inserendo uno spazio vuoto fra i due operandi.

(C5) `factor(%);`
(D5) $\frac{(2y - 3x)^2}{4}$

Raccoglimento a fattore comune:

(C6) `2*a^3*b^2 + 6*a^2*b^3 - 8*a^3*b;`
(D6) $6a^2b^3 + 2a^3b^2 - 8a^3b$

(C7) `factor(%);`
(D7) $2a^2b(3b^2 + ab - 4a)$

Il comando **FACTOR** non fattorizza espressioni con coefficienti irrazionali o complessi: in tal caso viene riprodotta l'espressione immessa:

$$a^2 - 6a + 4 = (a - 3 - \sqrt{5})(a - 3 + \sqrt{5})$$

(C8) `a^2 - 6*a + 4;`
(D8) $a^2 - 6a + 4$
(C9) `factor(%);`

(D9) $a^2 - 6a + 4$

Il comando **GFACTOR** effettua la fattorizzazione nel campo dei numeri complessi:

(C10) $a^4 - 1$;

(D10) $a^4 - 1$

(C11) `gfactor(%);`

(D11) $(a - 1)(a + 1)(a - \%I)(a + \%I)$
 $\%I = \sqrt{-1}$

(C17) $a^2 - 2a + 2$;

(D17) $a^2 - 2a + 2$

(C18) `gfactor(%);`

(D18) $(a - \%I - 1)(a + \%I - 1)$

Il comando **EXPAND** effettua lo sviluppo di potenze:

(C19) `expand((b + 2*a)^5);`

(D19) $b^5 + 10ab^4 + 40a^2b^3 + 80a^3b^2 + 80a^4b + 32a^5$

(C20) $(a^{1/3} - b)^5$;

(D20) $(a^{1/3} - b)^5$

(C21) `expand(%);`

(D21) $-b^5 + 5a^{1/3}b^4 - 10a^{2/3}b^3 + 10a^2b^2 - 5a^{4/3}b + a^{5/3}$

Il comando **RADCAN** semplifica espressioni che contengono logaritmi, esponenziali e radicali:

(C22) `radcan((a-sqrt(5)-3)*(a-3+sqrt(5)));`

(D22) $a^2 - 6a + 4$

(C23) `radcan(exp(2*x+1)/exp(-x+2));`

(D23) $\%E3x - 1$

Valutazione di espressioni

Il comando **AT** consente di valutare una espressione letterale in corrispondenza a determinati valori assegnati alle variabili che compaiono in essa; ha due argomenti: l'espressione, e le variabili coinvolte nella sostituzione con i corrispondenti valori (le variabili sono indicate come lista e quindi racchiuse tra parentesi quadre):

(C1) $(a^3 + 2*b)^2$;
(D1) $(2 b + a^3)^2$

(C2) `at(esp,[a=2, b=1]);`
(D2) 100

Analogo risultato si ottiene col comando **EV**:

(C3) `ev(esp,[a=2,b=1]);`
(D3) 100

Questo comando risulta molto piu' potente.

Assegnamo alla variabile b il valore 1:

(C4) `b:1;`
(D4) 1

(C5) `at(esp,[a=2])`
(D5) $(2 b + 8)^2$

Il comando AT vede solo la variabile a che compare fra i suoi argomenti, ma non la variabile b.

(C6) `ev(esp,[a=2]);`
(D6) 100

Il comando EV opera invece in un ambiente che comprende sia la variabile locale a, sia la variabile globale b.

Per cancellare dalla memoria il valore attribuito alla variabile globale b, utilizziamo il comando **KILL** (il comando **KILL(all)** cancella il contenuto delle variabili in memoria e fa ripartire la sessione):

(C7) `kill(b);`
(D7) DONE

(C8) `ev(esp,a=2);`
(D8) $(2 b + 8)^2$

Gli esempi che seguono mostrano altre possibilità offerte dal comando EV:

(C9) `ev(log(esp),[a=2]);`
(D9) LOG(100)

(C10) `ev(log(esp),[a=2],numer);`
(D10) 4.605170185988092

(C11) `ev(sqrt(3));`
(D11) SQRT(3)

(C12) `ev(sqrt(3),float);`
(D12) 1.732050807568877

Per valutare un polinomio $p(x)$ di grado n , in corrispondenza ad un valore di x assegnato, è opportuno convertirlo nella rappresentazione di **Horner** in modo che il polinomio possa essere valutato con n addizioni ed n moltiplicazioni, riducendo così i tempi di calcolo e gli errori di arrotondamento/troncamento:

(C11) $px: -x^4 + 2x^3 + bx^2 - x + 2;$

(D11) $- X^4 + 2 X^3 + b X^2 - X + 2$

Rappresentiamo il polinomio nella forma richiesta rispetto alla variabile x mediante il comando **HORNER**:

(C12) `horner(px,x);`

(D12) $X (X ((2 - X) X + b) - 1) + 2$

(C13) `ev(%,x=1.03,b=2);`

(D13) 4.15174519

Equazioni

Il comando SOLVE risolve equazioni algebriche generali fino al quarto grado, equazioni algebriche particolari oppure equazioni contenenti funzioni logaritmiche, esponenziali, etc, che possono essere ricondotte a particolari equazioni algebriche.

Equazioni determinate:

(C1) `solve(2*(2-3*x)+3*(x+1)=10*x+8,x);`

(D1) $[x = -\frac{1}{13}]$

Se il primo argomento è una espressione, viene uguagliato automaticamente a 0:

(C2) `solve(x^2-6*x+4,x);`

(D2) $[x = 3 - \text{SQRT}(5), x = \text{SQRT}(5) + 3]$

Equazione impossibile:

(C3) `solve(x+5=2*(x-3)-x-1,x);`

(D3) $[\]$

Equazione indeterminata:

(C4) `solve(x-5=2*(x-3)-x+1,x);`

(D4) $[x = x]$

Equazioni con radici in campo complesso:

(C5) `solve(x^4-1,x);`

(D5) $[x = \%I, x = -1, x = -\%I, x = 1]$

Per ottenere solo le radici reali di polinomi o equazioni polinomiali, impostiamo a TRUE la variabile **REALONLY** e richiamiamo il comando **ALGSYS**:

(C6) `realonly:true;`

(D6) `TRUE`

(C7) `algsys([x^4-1],[x]);`

(D7) $[[x = 1], [x = -1]]$

Maxima risolve le equazioni generali di terzo grado applicando la formula di Cardano.

Ricordiamo che ogni equazione algebrica di terzo grado, mediante trasformazione a radici aumentate, si può sempre ridurre alla forma

$$x^3 + px + q = 0 \quad \text{con } p \text{ e } q \text{ numeri assegnati, reali o complessi.}$$

La formula di Cardano è espressa da

$$x = \left(-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right)^{1/3} + \left(-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}\right)^{1/3}$$

Posto $\Delta = \frac{q^2}{4} + \frac{p^3}{27}$, se $\Delta < 0$ l'equazione ha tre radici reali, sebbene all'interno dei radicali cubici compaiano dei numeri complessi coniugati; esprimendo tali numeri complessi coniugati in forma trigonometrica, si ottengono per le tre radici formule del tipo

$$x_1 = 2 \sqrt[3]{-p/3} \cos \omega/3, \quad x_2 = 2 \sqrt[3]{-p/3} \cos (\omega + 2\pi)/3, \quad x_3 = 2 \sqrt[3]{-p/3} \cos (\omega + 4\pi)/3$$

che ne evidenziano la realtà.

Consideriamo l'equazione

$$x^3 - 3x - 1 = 0$$

$$p = -3, \quad q = -1, \quad \Delta = -3/4$$

Per visualizzare le soluzioni in formato ridotto, impostiamo a FALSE la variabile **DISPLAY2D** (la visualizzazione in formato esteso si riattiva col comando `display2d:true`):

(C8) `display2d:false;`

(D8) FALSE

(C9) `solve(x^3-3*x-1,x);`

(D9) $[x = (-\sqrt{3}\%I/2-1/2)*(\sqrt{3}\%I/2+1/2)^{(1/3)} + (\sqrt{3}\%I/2-1/2)/(\sqrt{3}\%I/2+1/2)^{(1/3)},$

$x = (\sqrt{3}\%I/2-1/2)*(\sqrt{3}\%I/2+1/2)^{(1/3)} + (-\sqrt{3}\%I/2-1/2)/(\sqrt{3}\%I/2+1/2)^{(1/3)},$

$x = (\sqrt{3}\%I/2+1/2)^{(1/3)}+1/(\sqrt{3}\%I/2+1/2)^{(1/3}]$

Riscriviamo la terza radice in un formato più leggibile:

$$x = \left(\frac{i\sqrt{3}}{2} + \frac{1}{2} \right)^{1/3} + \frac{1}{\left(\frac{i\sqrt{3}}{2} + \frac{1}{2} \right)^{1/3}}$$

Le tre radici sono tutte reali (sebbene *Maxima* le esprima in una forma in cui compare l'unità immaginaria), come si può verificare dal grafico della funzione o utilizzando il comando **ALGSYS** (che determina una approssimazione delle radici):

(C10) `algsys([x^3-3*x-1],[x]);`

(D10) $[[x = 1.879385232208487], [x = -1.532088888888889], [x = -0.34729635646322]]$

E' interessante notare che, risolvendo la stessa equazione mediante *Derive*, le tre radici risultano espresse mediante funzioni goniometriche secondo le formule descritte in precedenza:

$[x = -2 \sin(\pi/18), x = 2 \cos(\pi/9), x = -2 \cos(2\pi/9)]$

(C11) `solve(x^3-5*x-2,x);`

(D11) $[x = 1 - \sqrt{2}, x = \sqrt{2} + 1, x = -2]$

Equazioni fratte:

(C12) `solve((2*x+1)/(x+1)=2/(x^2-1)+2,x);`

(D12) $[]$

(C13) `solve((x-1)/(x-2)-(x+2)/(x-1)=(x+1)/(x^2-3*x+2),x);`

(D13) $[x = \frac{4}{3}]$

Equazioni letterali

Gli esempi mostrano come la discussione in merito alla accettabilità delle radici venga lasciata all'utente: si ribadisce la necessità di un uso critico dello strumento informatico.

(C14) `solve(a*x-3=2*b*x,x);`

$$(D14) \quad [x = -\frac{3}{2b-a}]$$

(C15) solve(a*x-a,x);

$$(D15) \quad [x = 1]$$

Per calcolare in modo approssimato le radici di equazioni numeriche algebriche o trascendenti, si veda il paragrafo: **Zeri di una funzione**.

Maxima consente di risolvere particolari equazioni trascendenti, ma la verifica delle soluzioni è lasciata all'utente.

Equazioni esponenziali

Risolviamo l'equazione

$$3^{3x} - 3^{2x} - 3^{x+1} + 3 = 0$$

Impostiamo a TRUE la variabile **EXPTSUBST**, che abilita la sostituzione $3^x=t$:

(C1) exptsubst:true;

(D1) TRUE

(C2) esp:3^(3*x)-3^(2*x)-3^(x+1)+3;

$$(D2) \quad 3^{3x} - 3^{2x} - 3^{x+1} + 3$$

Esprimiamo in modo opportuno l'equazione mediante il comando **RADCAN**:

(C3) radcan(%);

$$(D3) \quad 3^{3x} - 3^{2x} - 3 \cdot 3^x + 3$$

Effettuiamo in D3 la sostituzione $3^x=t$:

(C4) subst(t,3^x,%);

$$(D4) \quad t^3 - t^2 - 3t + 3$$

Risolviamo D4 rispetto alla variabile t :

(C5) solve(%,t);

$$(D5) \quad [t = -\sqrt{3}, t = \sqrt{3}, t = 1]$$

(C6) solve(3^x=3^(1/2),x);

$$(D6) \quad [x = \frac{1}{2}]$$

(C7) solve(3^x=1,x);

$$(D7) \quad [x = 0]$$

La soluzione $t = -\sqrt{3}$ non è accettabile in campo reale, ma *Maxima* la considera in campo complesso:

(C8) solve(3^x = -3^(1/2),x);

$$(D8) \quad [x = \frac{\text{LOG}(-\sqrt{3})}{\text{LOG}(3)}]$$

Risolviamo l'equazione:

$$3 \cdot 2^{4x} - 5 \cdot 4^x = 2$$

(C9) `esp:3*2^(4*x)-5*4^x=2;`

(D9) $3 \cdot 2^{4x} - 5 \cdot 4^x = 2$

(C10) `radcan(%);`

(D10) $3 \cdot 2^{4x} - 5 \cdot 2^{2x} = 2$

Effettuiamo la sostituzione $2^{2x}=t$:

(C11) `subst(t,2^(2*x),%);`

(D11) $3 \cdot t^2 - 5 \cdot t = 2$

(C12) `solve(%,t);`

(D12)
$$\begin{matrix} 1 \\ [t = 2, t = - \frac{1}{3}] \\ 3 \end{matrix}$$

La seconda soluzione non è accettabile:

(C13) `solve(2^(2*x)=2,x);`

(D13)
$$\left[x = \frac{\text{LOG}(-\text{SQRT}(2))}{\text{LOG}(2)}, x = \frac{1}{2} \right]$$

Solo la soluzione $x=1/2$ è accettabile in campo reale.

Equazioni logaritmiche

Si consideri l'equazione

$$\log(x-1)+\log(x+1)=\log(4*x-5)$$

dove la funzione LOG indica il logaritmo naturale.

(C14) `esp:log(x-1)+log(x+1)=log(4*x-5);`

(D14) $\text{LOG}(x + 1) + \text{LOG}(x - 1) = \text{LOG}(4 x - 5)$

Riscriviamo l'equazione in forma opportuna utilizzando il comando **LOGCONTRACT** :

(C15) `logcontract(%);`

(D15) $\text{LOG}(x^2 - 1) = \text{LOG}(4 x - 5)$

(C16) `solve(x^2-1=4*x-5,x);`

(D16) $[x = 2]$

Con analogo procedimento si risolve ad esempio l'equazione

$$2 \log x - \log(x+6) = \log 5$$

L'equazione

$$3 \log^2 x - 5 \log x - 2 = 0$$

può essere risolta con *Maxima* mediante la sostituzione di variabile $\log x = t$, seguendo il procedimento descritto per le equazioni esponenziali.

Sistemi di equazioni

Il comando **SOLVE** consente anche di risolvere sistemi di equazioni.

(C1) `solve([2*x-y=3,x^2=4],[x,y]);`

(D1) `[[x = - 2, y = - 7], [x = 2, y = 1]]`

Dal punto di vista didattico può essere molto utile risolvere il sistema con tutti i passaggi che si compirebbero risolvendo il problema con carta e penna, per sostituzione.

(C2) `e1:2*x-y=3;`

(D2) $2x - y = 3$

(C3) `e2:x^2=4;`

(D3) $x^2 = 4$

Ricaviamo x dalla seconda equazione:

(C4) `solx:solve(e2,x);`

(D4) `[x = - 2, x = 2]`

Sostituiamo il primo valore di x nella prima equazione e risolviamo rispetto ad y :

(C5) `ev(e1,part(solx,1));`

(D5) $-y - 4 = 3$

(C6) `sol1y:solve(%,y);`

(D6) `[y = -7]`

Sostituiamo il secondo valore di x nella prima equazione e risolviamo rispetto ad y :

(C7) `ev(e1,part(solx,2));`

(D7) $4 - y = 3$

(C8) `sol2y:solve(%,y);`

(D8) `[y = 1]`

Il comando **PART** consente di estrarre elementi da una lista, una matrice o una espressione :

(C9) `part(solx,1);`

(D9) $x = - 2$

Funzioni

Una funzione viene definita mediante l'operatore :=

$$F(x) = e^x - \log \sin(\sqrt{2} x)$$

(C1) `f(x):= exp(x)-log(sin(sqrt(2)*x));`

(D1) `f(x) := EXP(x) - LOG(SIN(SQRT(2) x))`

In questo modo si può fare riferimento alla funzione senza dover ridigitare l'espressione analitica; volendo ad esempio determinare la derivata prima della funzione mediante il comando **DIFF**:

(C2) `diff(f(x),x);`

(D2) `%E - $\frac{x \text{ SQRT}(2) \text{ COS}(\text{SQRT}(2) x)}{\text{SIN}(\text{SQRT}(2) x)}$`

Consideriamo la funzione a tratti:

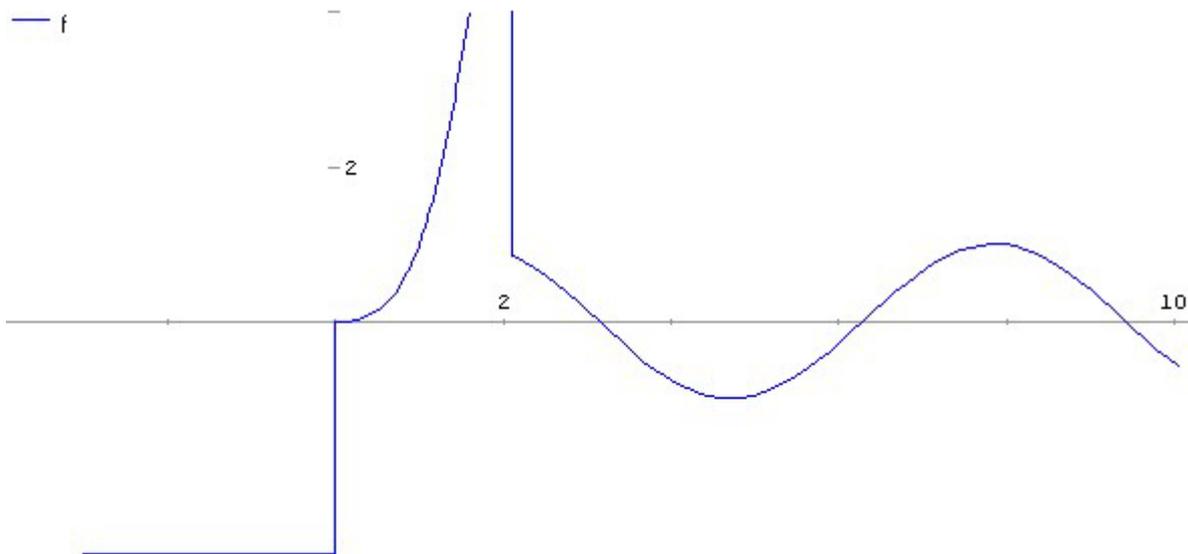
$$f(x) = \begin{cases} -3 & x \leq 0 \\ x^3 & 0 < x \leq 2 \\ \sin x & x > 2 \end{cases}$$

Viene definita usando il comando **BLOCK**, che consente di specificare le varie condizioni:

(C3) `f(x):=block([],if (x<=0) then return(-3) else (if (x>0 and x<=2) then return(x^3) else return(sin(x))));`

(D3) `f(x) := BLOCK([], IF x <= 0 THEN RETURN(- 3)
ELSE (IF x > 0 AND x <= 2 THEN RETURN(x) ELSE RETURN(SIN(x))))`

(C4) `plot2d(f,[x,-3,10],[y,-4,4]);`



Si osservi che *Maxima* traccia erroneamente due segmenti verticali di connessione dei singoli tratti.

(C5) `f(-5);`

(D5) `- 3`

(C6) `f(2);`

(D6)

8

Funzioni composte

Maxima consente di determinare l'espressione analitica e il grafico di una funzione composta

(C1) $f(x):=x-1$;

(D1) $f(x) := x - 1$

(C2) $g(x):=x^3$;

(D2) $g(x) := x^3$

(C3) $h1(x):=f(g(x))$;

(D3) $h1(x) := f(g(x))$

(C4) $h1(3)$;

(D4) 26

(C5) $f(g(x))$;

(D5) $x^3 - 1$

(C6) $h2(x):=g(f(x))$;

(D6) $h2(x) := g(f(x))$

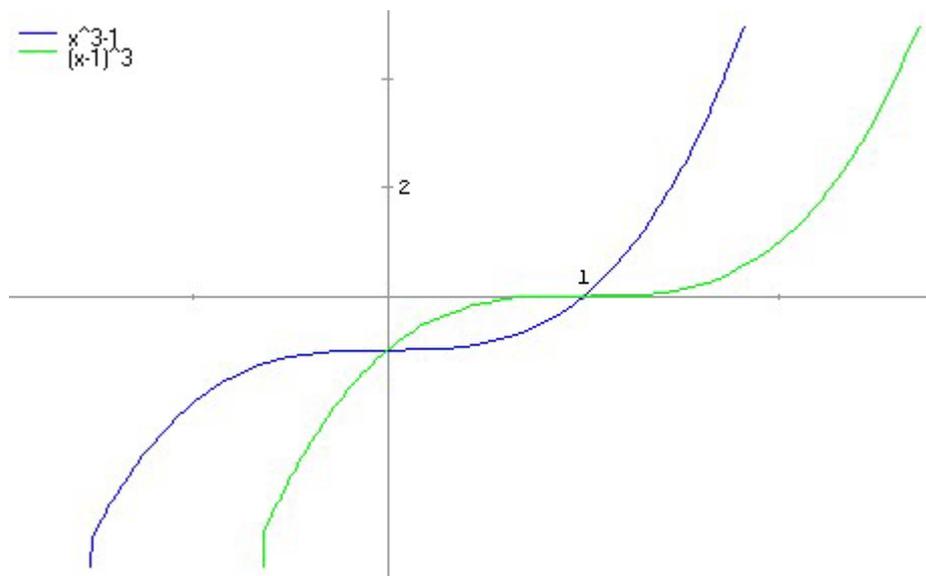
(C7) $h2(3)$;

(D7) 8

(C8) $g(f(x))$;

(D8) $(x - 1)^3$

(C9) $\text{plot2d}([h1(x),h2(x)],[x,-4,4],[y,-5,5])$;



Funzione inversa

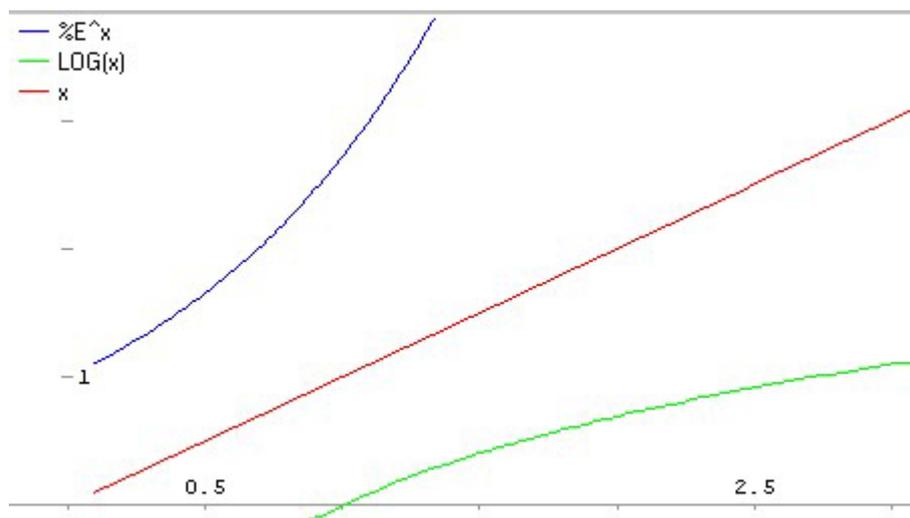
La funzione inversa si ottiene risolvendo l'equazione $y=f(x)$ rispetto alla variabile x , per mezzo del comando **SOLVE**:

(C1) `solve(y=exp(x),x);`

(D1) `[x = LOG(y)]`

Verifichiamo che il grafico della funzione e della sua inversa sono simmetrici rispetto alla retta $y=x$:

(C2) `plot2d([exp(x),log(x),x],[x,0.1,4],[y,-0.1,4]);`



Se si fa variare la x in un intervallo comprendente valori negativi (es. $[x,-4,4]$), compare un ramo di curva anomalo, le cui motivazioni sono descritte al paragrafo: **Studio completo di funzione reale**.

Maxima si limita a risolvere simbolicamente l'equazione $y=f(x)$:

(C3) `solve(y=sin(x),x);`

SOLVE is using arc-trig functions to get a solution.

Some solutions will be lost.

(D3) `[x = ASIN(y)]`

Occorre pertanto riflettere sulle effettive condizioni che consentono di determinare la funzione inversa (la funzione $y=\arcsin x$ è definita solo sull'intervallo $[-\pi/2, \pi/2]$, poiché in tale intervallo la funzione $y=\sin x$ è sempre crescente e assume tutti i valori compresi tra -1 e $+1$)

Un interessante spunto per la discussione in classe può essere costituito dal comportamento di *Maxima* in relazione alla funzione potenza con esponente naturale dispari:

(C4) `f(x):=x^3;`

(D4) `f(x) := x^3`

(C5) `solve(y=f(x),x);`

Maxima risolve l'equazione in campo complesso:

$$(D5) \left[x = \frac{(\text{SQRT}(3) \% I - 1) y^{1/3}}{2}, x = -\frac{(\text{SQRT}(3) \% I + 1) y^{1/3}}{2}, x = y^{1/3} \right]$$

La funzione

$$y=x^k \quad x>0, \quad k \text{ naturale dispari}$$

è continua e crescente quindi ammette funzione inversa

$$y=x^{1/k}$$

Il significato del simbolo $x^{1/k}$ viene definito attraverso le funzioni esponenziale e logaritmo:

$$x^{1/k} = \exp(\ln(x^{1/k})) = \exp((1/k)\ln x)$$

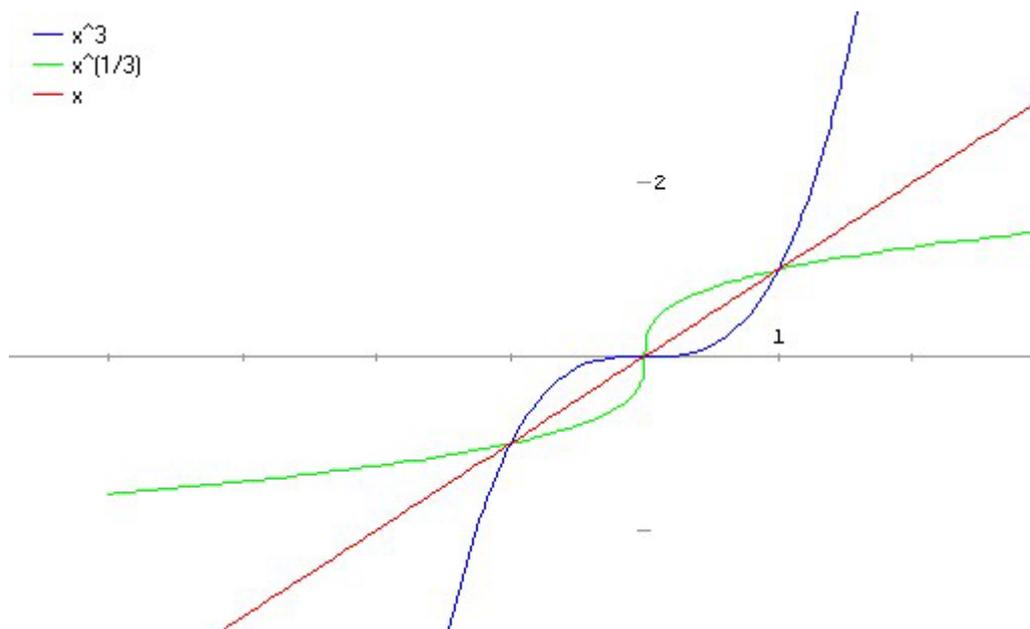
La funzione può essere prolungata con continuità nell'origine ponendo $0^{1/k}=1$

Pertanto la funzione $y=x^{1/3}$ si intende in generale definita per $x \geq 0$.

D'altra parte, poichè la funzione $y=x^k$ k naturale dispari, è continua e strettamente crescente su tutto \mathbb{R} e quindi dotata di funzione inversa, alcuni considerano tale funzione inversa $y=x^{1/k}$ definita per ogni x reale. Pertanto si pone $(-8)^{1/3} = -2$, essendo -2 l'unica soluzione reale dell'equazione $x^3=-8$; tuttavia, rappresentando l'esponente $1/3$ con una frazione equivalente, si ottiene una diversa definizione: $(-8)^{2/6}=2$!!!!!

Scopriamo quale è la scelta operata dagli autori di *Maxima*;

(C6) plot2d([f(x),x^(1/3),x],[x,-4,4],[y,-4,4]);



E' interessante confrontare il grafico con quelli prodotti da *Mathematica* e *Derive*.

Limiti di funzioni reali ad una variabile reale

Maxima utilizza il comando **LIMIT**.

Limite finito per una funzione in un punto:

$$\lim_{x \rightarrow 2} 4x+1 = 12$$

$x \rightarrow 2$

(C1) `limit(4*(x+1),x,2);`

(D1) 12

Nell'esempio seguente, si osservi che non esiste il valore della funzione per $x=1$ ma esiste il valore del limite :

(C2) `limit((x^2-1)/(x-1),x,1);`

(D2) 2

Limite infinito per x tendente ad un valore finito :

(C3) `limit(1/(x-1)^2,x,1);`

(D3) INF

Consideriamo la funzione così definita :

(C4) `f(x):=1/(x-1);`

(D4)
$$f(x) := \frac{1}{x-1}$$

Non esiste il limite della funzione per $x \rightarrow 1$, quindi *Maxima* risponde UNDEFINED:

(C5) `limit(f(x),x,1);`

(D5) UND

Esiste il limite destro (PLUS), uguale a $+\infty$ (INF).

(C6) `limit(f(x),x,1,plus);`

(D6) INF

Esiste limite sinistro (MINUS), uguale a $-\infty$ (MINF)

(C7) `limit(f(x),x,1,minus);`

(D7) MINF

Esiste il limite per $x \rightarrow \infty$ ed è uguale a 0:

(C8) `limit(f(x),x,inf);`

(D8) 0

Maxima non risolve disequazioni, quindi il valore calcolato per il limite può essere verificato non applicando la definizione bensì esaminando il grafico della funzione:

(C9) `Plot2d(f(x),[x,-0.5,2.5],[y,-10,10]);`

Limiti notevoli

(C10) `limit(sin(x)/x,x,0);`

(D10) 1

$$\lim_{x \rightarrow \infty} (1+1/x)^x = e$$

(C11) `limit((1+1/x)^x,x,inf);`

(D11) `%E`

Consideriamo il limite:

$$\lim_{x \rightarrow 0} x \sin \frac{1}{x} = 0$$

Per $x=0$ la funzione non è definita ma esiste il limite:

(C12) `limit(x*sin(1/x),x,0);`

(D12) `0`

Verifica del limite mediante grafico della funzione:

(C13) `plot2d(x*sin(1/x),[x,-0.2,0.2],[y,-0.9,0.9]);`

Forme indeterminate

$$\lim_{x \rightarrow \infty} (\sqrt{x^2+1} - \sqrt{x^2-3})$$

(C14) `limit(sqrt(x^2+1)-sqrt(x^2-3),x,inf);`

(D14) `0`

$$\lim_{x \rightarrow 0^+} (\log x - \log \sin 2x)$$

(C15) `limit(log(x)-log(sin(2*x)),x,0,plus);`

(D15) `- LOG(2)`

(C16) `limit(1/log(x)-1/(x-1),x,1);`

(D16) `1/2`

(C17) `limit((tan(x))/(1-cos(x)),x,0);`

(D17) `UND`

(C18) `limit((tan(x))/(1-cos(x)),x,0,minus);`

(D18) `MINF`

(C19) `limit((tan(x))/(1-cos(x)),x,0,plus);`

(D19) `INF`

Consideriamo ora $\lim_{x \rightarrow 1} \frac{\operatorname{arctg} x - \pi/4}{x-1}$

Si tratta di una forma indeterminata del tipo 0/0 ; applicando il teorema di de l'Hospital viene ricondotta a

$$\lim_{x \rightarrow 1} \frac{1}{1+x^2} = 1/2$$

Maxima applica invece la sostituzione di variabile $y = x-1$ e non riesce a risolvere il limite:

(C20) `f(x):= (atan(x)-%pi/4)/(x-1);`

(D20)
$$f(x) := \frac{\text{ATAN}(x) - \frac{\%PI}{4}}{x - 1}$$

(C21) `limit(f(x),x,1);`

(D21)
$$\lim_{x \rightarrow 0} \frac{\text{ATAN}(x + 1) - \frac{\%PI}{4}}{x}$$

Abilitiamo ora il ricorso allo sviluppo in serie di Taylor, impostando al valore TRUE la variabile **TLIMSWITCH** (per default con valore FALSE), e *Maxima* effettua correttamente il calcolo del limite:

(C22) `tlimswitch:true;`

(D22) TRUE

(C23) `limit(f(x),x,1);`

(D23)
$$\frac{1}{2}$$

Esaminiamo ora il seguente comportamento:

(C24) `tlimswitch:false;`

(D24) FALSE

(C25) `limsubst:false;`

(D25) FALSE

Nonostante la variabile **LIMSUBST** venga impostata a FALSE, per prevenire le sostituzioni nel tentativo di risolvere le forme indeterminate, *Maxima* effettua comunque la sostituzione $y=x-1$ già analizzata in precedenza e, non potendo applicare lo sviluppo in serie perchè TLIMSWITCH ha valore FALSE, non è in grado di calcolare il limite:

(C26) `limit(f(x),x,1);`

(D26)
$$\lim_{x \rightarrow 0} \frac{\text{ATAN}(x + 1) - \frac{\%PI}{4}}{x}$$

Applichiamo ora il teorema di De l'Hospital al calcolo del limite proposto, effettuando tutti i passaggi.

Derivata del numeratore:

(C27) `diff(atan(x)-%pi/4,x);`

(D27)
$$\frac{1}{x^2 + 1}$$

Derivata del denominatore:

(C28) `diff(x-1,x);`

(D28)
$$1$$

Limite, per $x \rightarrow 1$, del rapporto delle derivate:

(C29) `limit(D1/%x,1);`

(D29)
$$\frac{1}{2}$$

Vediamo ora un esempio in cui l'applicazione "pedestre" del teorema di De l'Hospital da parte del programma genera una situazione di errore:

$$\lim_{x \rightarrow 0} \frac{x - \arctg x}{\arcsen x - x}$$

Si tratta di una forma indeterminata del tipo 0/0. Dopo una prima applicazione del teorema, occorre con opportuni passaggi trasformare il limite ottenuto nel prodotto dei limiti

$$\lim_{x \rightarrow 0} \frac{\sqrt{1-x^2}}{1+x^2} \cdot \lim_{x \rightarrow 0} \frac{x^2}{1-\sqrt{1-x^2}}$$

Il primo limite tende ad 1; riapplicando il teorema al secondo ed effettuando il prodotto dei limiti, si giunge al valore 2 del limite iniziale.

Il programma invece applica de l'Hospital senza effettuare i passaggi indicati e ottiene pertanto un quoziente con grado del numeratore e del denominatore sempre piu' elevato, quindi segnala un messaggio di errore.

Se invece si imposta a TRUE la variabile TLIMSWITCH, il limite viene calcolato correttamente.

(C30) `f(x):=(x-atan(x))/(asin(x)-x);`

(D30)
$$f(x) := \frac{x - \text{ATAN}(x)}{\text{ASIN}(x) - x}$$

(C31) `tlimswitch:false;`

(D31)
$$\text{FALSE}$$

(C32) `limit(f(x),x,0);`

Quotient by a polynomial of higher degree

-- an error. Quitting. To debug this try `DEBUGMODE(TRUE);`

(C33) `tlimswitch:true;`

(D33)
$$\text{TRUE}$$

(C34) `limit(f(x),x,0);`

(D34)

2

Interpretazione geometrica del teorema di De l'Hospital

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x^2}$$

Si tratta di una forma indeterminata del tipo 0/0.

Il limite del rapporto delle derivate prime è ancora una forma indeterminata di questo tipo:

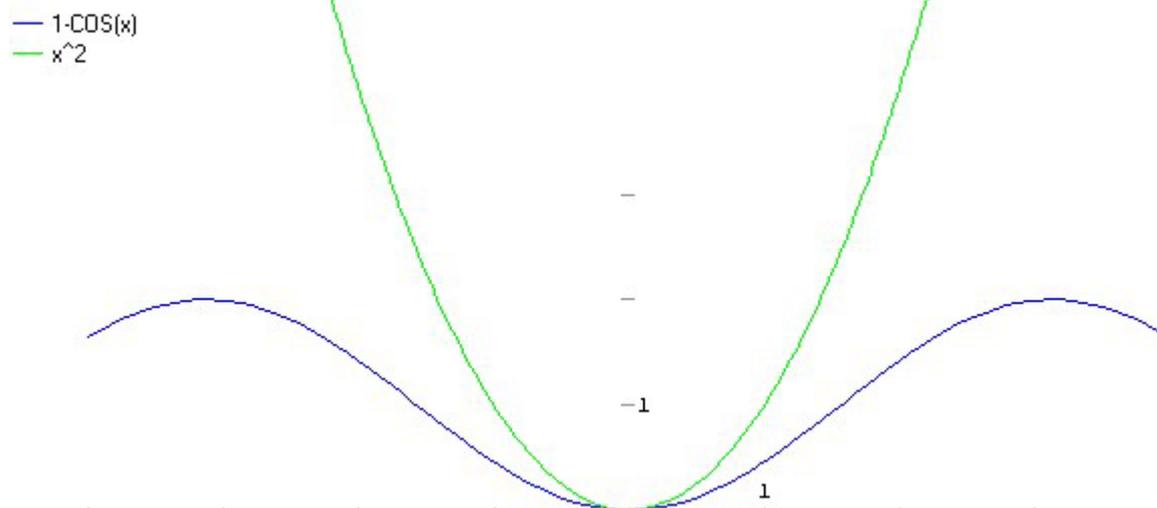
$$\lim_{x \rightarrow 0} \frac{\sin x}{2x}$$

(Osserviamo che la forma indeterminata potrebbe risolversi applicando il limite notevole

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1).$$

Ciò significa che le due curve $y = 1 - \cos x$ e $y = x^2$ hanno come tangente comune nell'origine l'asse delle ascisse; questo si può verificare disegnando in uno stesso grafico le due curve:

(C35) `plot2d([1-cos(x),x^2],[x,-4,4],[y,-1,5]);`



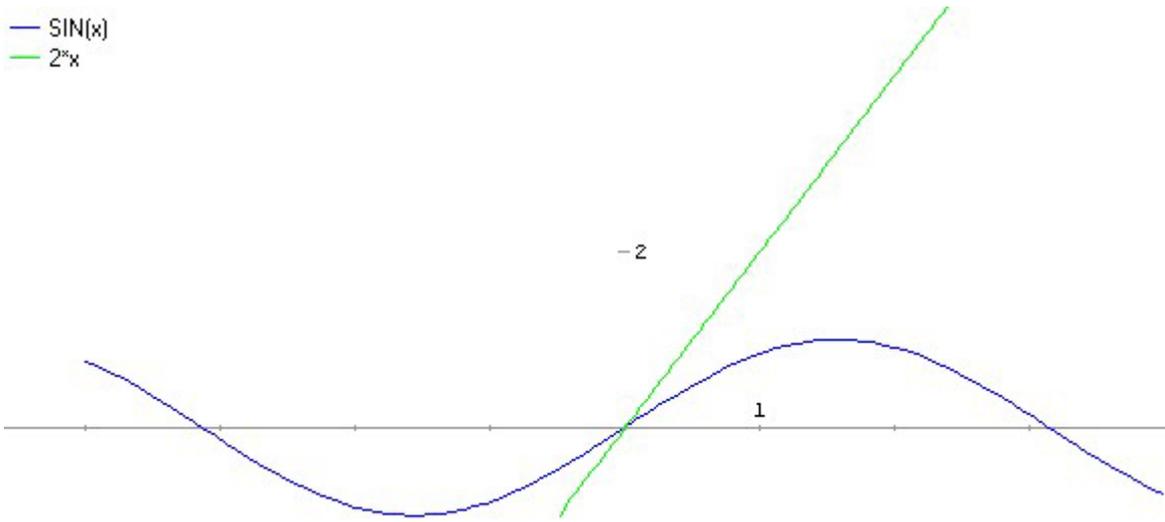
Ricorrendo alle derivate seconde, si ottiene :

$$\lim_{x \rightarrow 0} \frac{1 - \cos x}{x^2} = \lim_{x \rightarrow 0} \frac{\sin x}{2x} = \lim_{x \rightarrow 0} \frac{\cos x}{2} = \frac{1}{2}$$

Quindi le due curve $y = \sin x$ e $y = 2x$ hanno nell'origine tangenti diverse, con coefficienti angolari finiti, come si può osservare dal grafico:

(C36) `plot2d([sin(x),2*x],[x,-4,4],[y,-1,5]);`

— SIN(x)
— 2^x



Derivate

Maxima utilizza il comando **DIFF**, che ha come argomenti la funzione da derivare e la variabile rispetto alla quale derivare.

(C1) $f(x) := 2 * x^3 - 3 * x;$

(D1) $f(x) := 2 x^3 - 3 x$

Calcoliamo $f'(x)$ come limite del rapporto incrementale:

(C2) $f(x+h);$

(D2) $2 (x + h)^3 - 3 (x + h)$

(C3) $\text{limit}((f(x+h)-f(x))/h, h, 0);$

(D3) $6 x^2 - 3$

Calcoliamo ora la stessa derivata usando il comando **DIFF**:

(C4) $\text{diff}(f(x), x);$

(D4) $6 x^2 - 3$

(C5) $\text{diff}(\log(\sin(x)+x^3), x);$

(D5)
$$\frac{\text{COS}(x) + 3 x^2}{\text{SIN}(x) + x^3}$$

Significato geometrico del concetto di derivata

Data una funzione $y=f(x)$ e fissato un valore x_0 appartenente al suo dominio, $f'(x_0)$ è il coefficiente angolare della retta tangente al grafico della funzione nel punto $(x_0, f(x_0))$;

(C6) $f(x) := x^3 - 2 * x + 1;$

(D6) $f(x) := x^3 - 2 x + 1$

Consideriamo il punto di coordinate $(1, 0)$

(C7) $x_0: 1;$

(D7) 1

Definiamo il rapporto incrementale come funzione della variabile h :

(C8) $\text{rpinc}(h) := (f(x_0+h) - f(x_0))/h;$

(D8)
$$\text{rpinc}(h) := \frac{f(x_0 + h) - f(x_0)}{h}$$

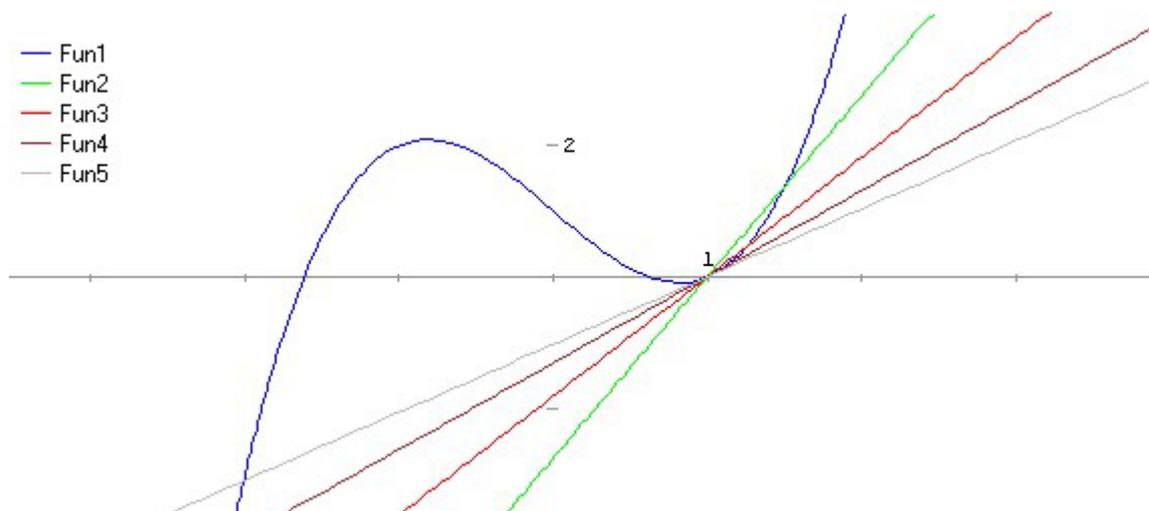
Scriviamo l'equazione della retta secante che congiunge i punti di coordinate $(x_0, f(x_0))$ e $(x_0+h, f(x_0+h))$:

(C9) $\text{rsec}(h, x) := \text{rpinc}(h) * (x - x_0) + f(x_0);$

(D9) $\text{rsec}(h, x) := \text{rpinc}(h) (x - x_0) + f(x_0)$

Quando h tende a 0, la retta secante tende alla retta tangente al grafico della funzione f nel punto $(1, 0)$:

(C10) `plot2d([f(x),rsec(0.5,x),rsec(0.25,x),rsec(0.1,x),rsec(0.01,x)],[x,-4,4],[y,-4,4]);`



Studio completo di funzione reale ad una variabile reale

$$F(x) = \frac{x^3 - 4x}{x^2 - 1} + 5 [\ln(x+1) - \ln(x-1)]$$

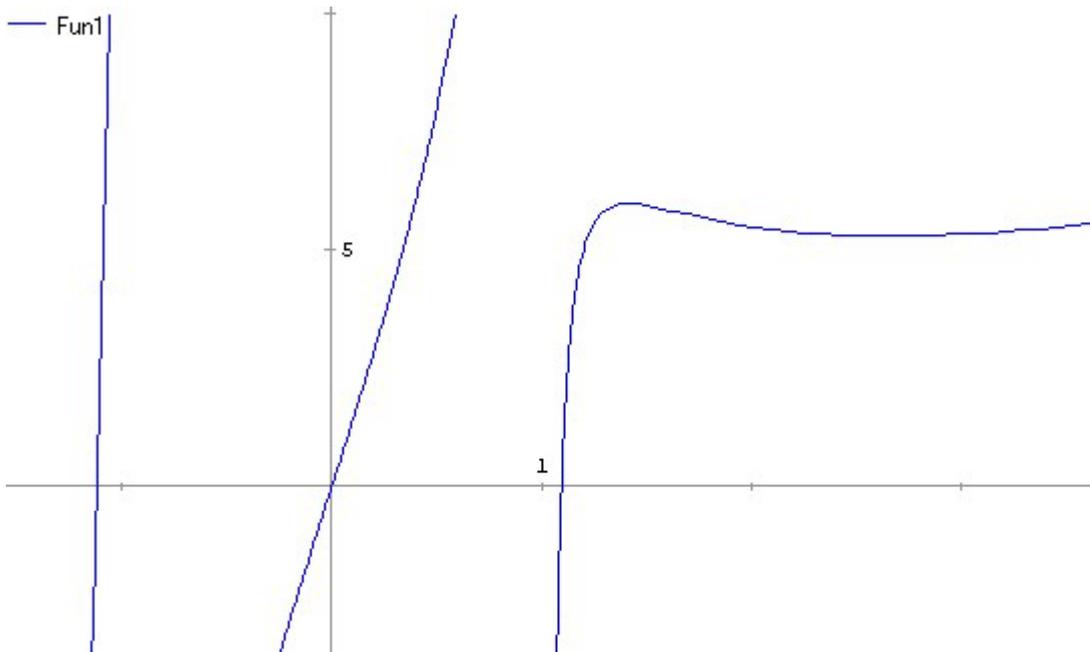
(Per lo studio della stessa funzione tramite *Mathematica* e *Derive*, v. “Matematica e software didattici”, IRSAE Emilia Romagna, Bologna 1999, pagg. 40-47)

(C1) `f(x):=(x^3-4*x)/(x^2-1)+5*(log(x+1)-log(x-1));`

$$(D1) \quad f(x) := \frac{x^3 - 4x}{x^2 - 1} + 5 (\text{LOG}(x + 1) - \text{LOG}(x - 1))$$

La funzione è definita per ogni $x > 1$. Per disegnare il grafico, *Maxima* richiede l'intervallo delle ascisse da considerare, che è opportuno coincida col dominio della funzione (inclusi gli eventuali punti di discontinuità), perché altrimenti il programma segnala degli errori o genera, come si può vedere nell'esempio che segue, grafici non corretti:

(C2) `plot2d(f(x),[x,-4,4],[y,-5,10]);`



Il ramo sinistro di curva esterno al dominio deriva dal fatto che *Maxima* (come *Mathematica*), determina il logaritmo di un numero in campo complesso; per $x < -1$, $\log(x+1)$ e $\log(x-1)$ hanno la stessa parte immaginaria quindi la loro differenza è un numero reale, come si può verificare cdo i due valori per $x = -1.01$:

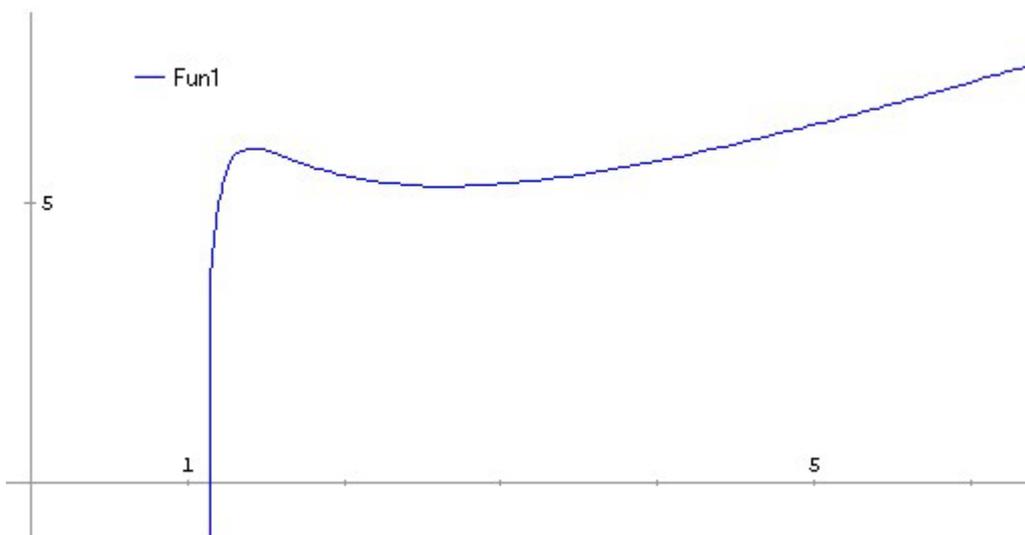
(C3) `ev([log(x+1),log(x-1)], x = -1.01);`

(D3) `[3.141592653589793 %I - 4.605170185988091,
3.141592653589793 %I + 0.69813472207098]`

Nell'intervallo $-1 < x < 1$ il logaritmo non è definito, e *Maxima* disegna erroneamente il grafico di $(x^3-4*x)/(x^2-1)$, corrispondente al ramo centrale del grafico.

Disegniamo ora il grafico corretto con riferimento al dominio:

(C4) `plot2d(f(x),[x,1.01,8],[y,-4,10]);`



Per determinare l'intersezione con l'asse x , siccome la funzione è trascendente e non riconducibile a casi particolari, utilizziamo il comando **INTERPOLATE** che implementa il metodo di bisezione, e

richiede pertanto un intervallo agli estremi del quale la funzione assuma valori di segno opposto; dal grafico deduciamo che la funzione possiede uno zero in prossimità di 1:

(C5) `interpolate(f(x)=0,x,1.01,2);`
 (D5) `1.094519583652444`

Calcolo del limite per $x \rightarrow 1^+$ (osserviamo che *Maxima* risolve direttamente la forma indeterminata $\infty - \infty$):

(C6) `limit(f(x),x,1,plus);`
 (D6) `MINF`

Esaminiamo ora il comportamento della funzione per $x \rightarrow \infty$:

(C7) `limit(f(x),x,inf);`
 (D7) `INF`

Ricerca asintoto obliquo:

(C8) `m:limit(f(x)/x,x,inf);`
 (D8) `1`

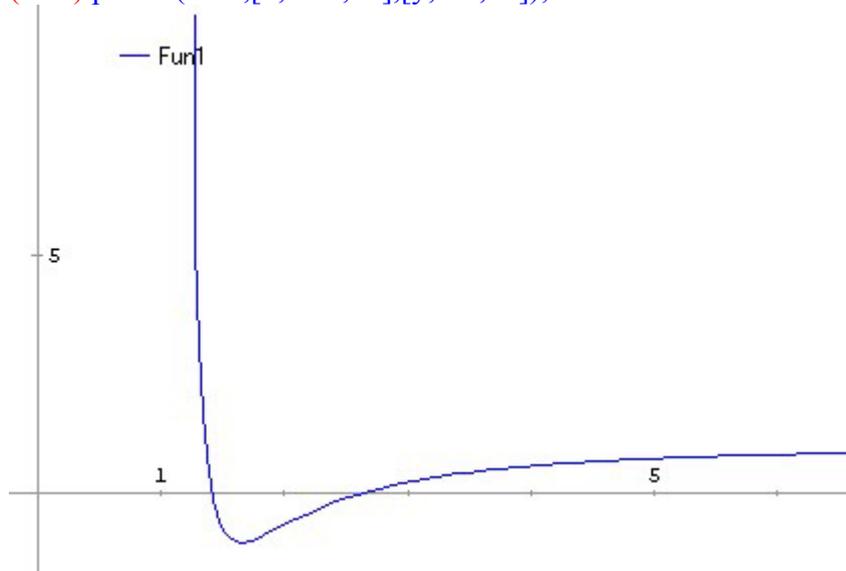
(C9) `q:limit(f(x)-m*x,x,inf);`
 (D9) `0`

La retta $y=x$ è asintoto obliquo per $x \rightarrow \infty$.

Studio della derivata prima:

(C10) `der1:ratsimp(diff(f(x),x));`
 (D10)
$$\frac{x^4 - 9x^2 + 14}{x^4 - 2x^2 + 1}$$

(C11) `plot2d(der1,[x,1.01,10],[y,-10,10]);`



Trattandosi di una funzione algebrica, per la ricerca degli zeri possiamo utilizzare ora il comando

SOLVE :

(C12) `solve(der1=0,x);`
 (D12) `[x = - SQRT(2), x = SQRT(2), x = - SQRT(7), x = SQRT(7)]`

Delle 4 soluzioni, solo due appartengono al dominio della funzione: $\sqrt{2}$ e $\sqrt{7}$; mediante il comando **EV** calcoliamo il valore delle ordinate corrispondenti:

(C13) `ev(f(sqrt(2)),float);`

(D13) 5.985308745449242

(C14) `ev(f(sqrt(7)),float);`

(D14) 5.299702961651823

Possiamo quindi concludere che $(\sqrt{2}, 5.9853)$ è punto di massimo relativo, mentre $(\sqrt{7}, 5.2997)$ è punto di minimo relativo.

Studio della derivata seconda per la ricerca degli eventuali punti di flesso:

(C15) `diff(der1,x);`

Non-variable 2nd argument to DIFF:

SQRT(2)

#0: f(x=SQRT(2))

-- an error. Quitting. To debug this try DEBUGMODE(TRUE);)

Maxima ci segnala una situazione di errore: la variabile x è legata al valore $\sqrt{2}$ calcolato in precedenza. Liberiamo la variabile x:

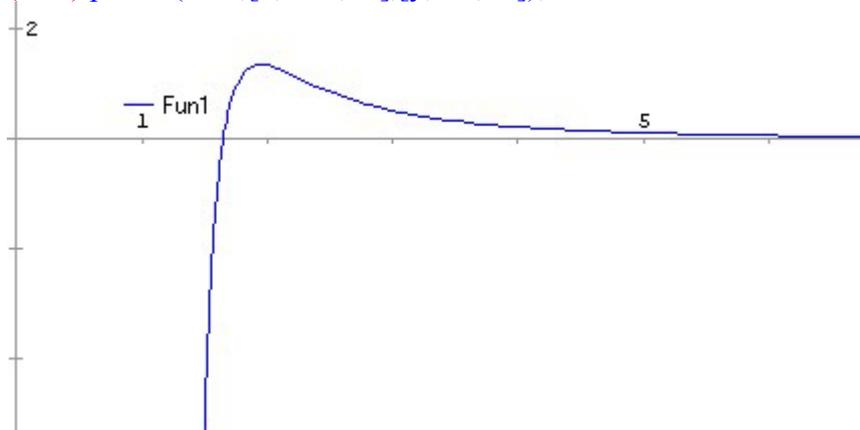
(C15) `kill(x);`

(D15) DONE

(C16) `der2:ratsimp(diff(der1,x));`

(D16)
$$\frac{14x^3 - 38x}{x^6 - 3x^4 + 3x^2 - 1}$$

(C17) `plot2d(der2,[x,1.01,10],[y,-10,10]);`



(C18) `solve(der2=0,x);`

(D18)
$$\left[x = -\frac{\text{SQRT}(19)}{\text{SQRT}(7)}, x = \frac{\text{SQRT}(19)}{\text{SQRT}(7)}, x = 0 \right]$$

$\sqrt{19/7}$ è l'ascissa del punto di flesso, la cui ordinata può essere calcolata col comando **EV** come mostrato in precedenza.

Equazione della tangente nel punto di flesso:

(C19) `a:sqrt(19/7);`

SQRT(19)

(D19)

$$\text{-----}$$

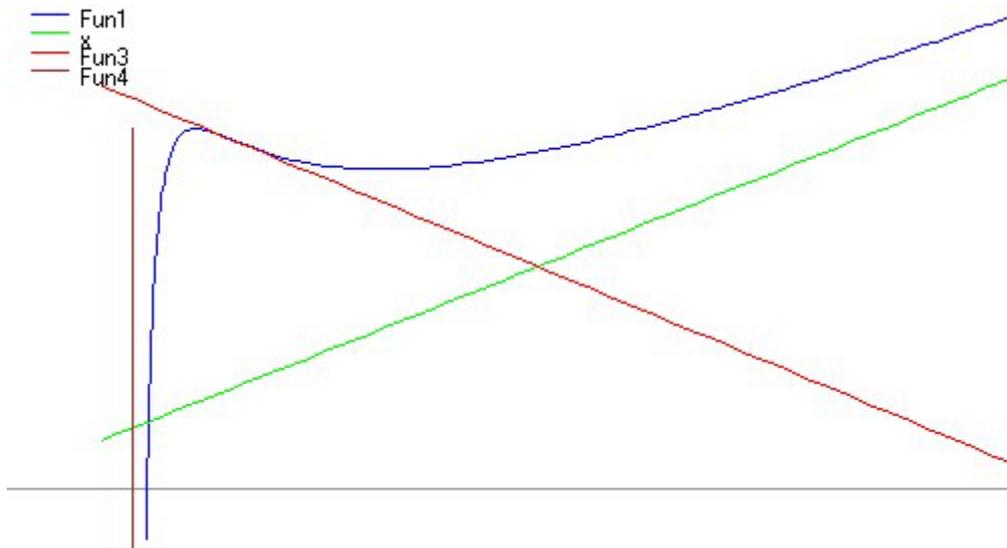
$$\text{SQRT}(7)$$

(C20) $\text{tflesso}(x) := \text{ev}(\text{der1}, x=a) * (x-a) + f(a);$

(D20) $\text{tflesso}(x) := \text{EV}(\text{der1}, x = a, \text{FLOAT}) (x - a) + f(a)$

Disegno del grafico della funzione, comprendente l'asintoto verticale, l'asintoto obliquo e la tangente nel punto di flesso:

(C21) $\text{plot2d}([f(x), x, \text{tflesso}(x), [\text{parametric}, 1, t, [t, -1, 6]]], [x, 0.8, 8], [y, -1, 8]);$



Per disegnare l'asintoto verticale di equazione $x=1$ si è utilizzata la forma parametrica.

Rappresentazione grafica di curve parametriche

Mediante il comando PLOT2D *Maxima* consente la rappresentazione grafica di curve espresse in forma parametrica, cioè da equazioni del tipo

$$x=f(t)$$

$$y=g(t) \quad t \text{ appartenente ad } [a,b].$$

Consideriamo la curva di Lissajous, rappresentata dalle equazioni parametriche

$$x= \cos ht$$

$$y= \sin kt \quad , \quad t \text{ reale}$$

Se h e k sono numeri interi, si tratta di una curva periodica con periodo 2π .

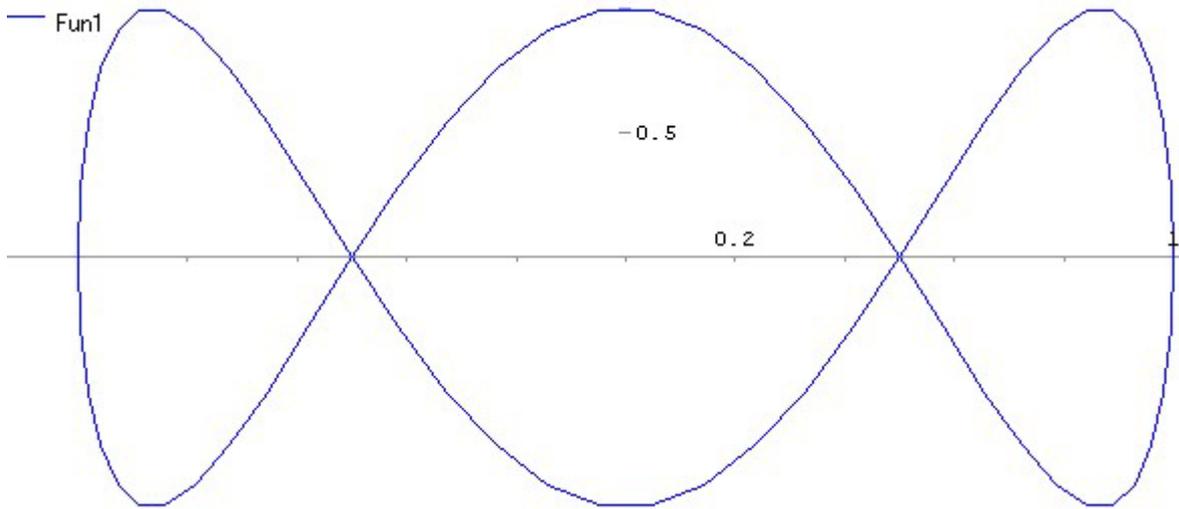
(C1) $h:1;$

(D1) 1

(C2) $k:3;$

(D2) 3

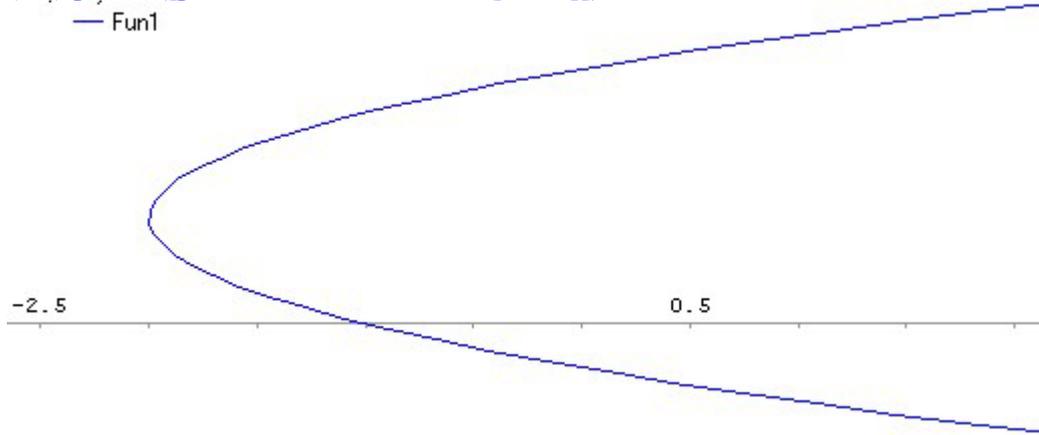
(C3) $\text{plot2d}([\text{parametric}, \cos(h*t), \sin(k*t), [t, 0, 3*\%pi]]);$



Possiamo usare questa forma del comando PLOT2D anche per rappresentare graficamente le curve di equazione $x=f(y)$, come ad esempio le rette $x=k$ (v. **Studio completo di funzione reale**) o le parabole con asse parallelo all'asse delle ascisse.

Grafico della parabola $x=y^2 - 2*y - 1$:

(C4) `plot2d([parametric, t^2-2*t-1, t, [t,-5,5]]);`



Zeri di una funzione

Maxima dispone di un linguaggio di programmazione mediante il quale è possibile descrivere algoritmi. Implementiamo come esempio il **metodo delle approssimazioni successive**, che discende dal teorema del punto fisso di seguito enunciato.

Sia $[a,b]$ un intervallo limitato e chiuso e sia $\Phi(x)$ una funzione definita su $[a,b]$ che verifica le seguenti condizioni:

$\Phi(x)$ è continua su $[a,b]$;

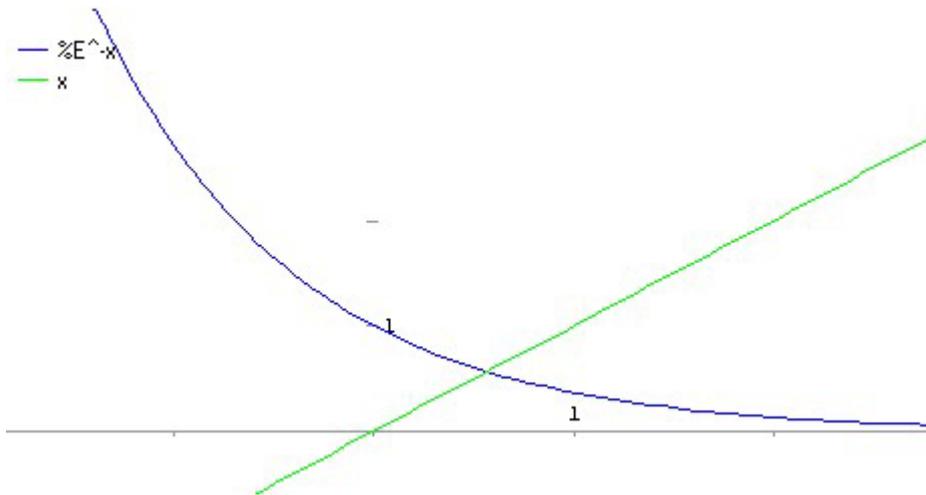
$\Phi(x)$ muta l'intervallo $[a,b]$ in se', cioè $\Phi(x)$ appartiene ad $[a,b]$ per ogni x di $[a,b]$;

$\Phi(x)$ è derivabile in $[a,b]$ e per una opportuna costante $\lambda < 1$ risulta $|\Phi'(x)| < \lambda$.

Allora per ogni scelta di x_0 in $[a,b]$ la successione definita da $x_n = \Phi(x_{n-1})$ converge all'unico punto fisso ζ della funzione $\Phi(x)$, cioè all'unica soluzione dell'equazione $\zeta = \Phi(\zeta)$.

Sia $\Phi(x) = e^{-x}$

(C1) `plot2d([exp(-x),x],[x,-2,4],[y,-1,4]);`



Su di un intervallo $[a,b]$ con $a > 0$ la funzione verifica le condizioni del teorema.

Il punto fisso della funzione è l'intersezione con la retta $y=x$; attivando il comando ZOOM e cliccando col mouse in prossimità del punto è possibile determinarne le coordinate.

Il sottoprogramma ha come parametri di ingresso: la funzione, la variabile da considerare (x), l'iterato iniziale x_0 , la precisione ϵ a meno della quale determinare l'approssimazione della soluzione, il numero massimo di iterazioni da eseguire, che assicura la terminazione del ciclo iterativo anche nel caso in cui il metodo non converga. Vengono utilizzate le variabili locali k , x_n , x_{prec} ; l'istruzione RETURN determina l'uscita dal sottoprogramma

```
(C2) apprsucc(funz,var, x0,eps,numiter):=block([xn,xprec,k],
      define(f(var),funz),k:0, xn:x0,
      do
      (k:k+1, xprec:xn, xn:f(xprec), if (abs(xprec-xn)<eps or k>numiter) then return(xn)))$
```

(C3) `h:exp(-x);`

(D2) `%E-x`

Utilizziamo il comando **SETCHECK** per visualizzare i valori di x_n calcolati ad ogni iterazione:

(C4) `setcheck:[xn]$`

(C5) `apprsucc(h,x,0.0,0.01,10);`

`xn set to 0.0`

```
xn set to 1.0
xn set to 0.36787944117144
xn set to 0.69220062755535
xn set to 0.50047350056364
xn set to 0.6062435350856
xn set to 0.54539578597503
xn set to 0.57961233550338
xn set to 0.56011546136109
xn set to 0.57114311508018
xn set to 0.56487934739105
```

(D5) 0.56487934739105

Si noti che il valore di xn ad ogni iterazione viene visualizzato due volte perchè compare in due assegnazioni.

(C6) kill(all) ;

Modifichiamo il sottoprogramma affinché stampi anche il numero di iterazioni eseguite:

```
(C1) apprsucc(funz,var,x0,eps,numiter):=block([xn,xprec,k],
      define(f(var),funz),k:0,xn:x0,
      do
      (k:k+1,xprec:xn,xn:f(xprec),if(abs(xprec-xn)<eps or k>numiter) then (print("xn=",xn,"
in ",k," iterazioni"),return(fine))))$
```

(C2) h:exp(-x);

(D2) %E^{-x}

(C3) setcheck:[xn]\$

(C4) apprsucc(h,x,0.0,0.001,100);

```
xn set to 0.0
xn set to 1.0
xn set to 0.36787944117144
xn set to 0.69220062755535
xn set to 0.50047350056364
xn set to 0.6062435350856
xn set to 0.54539578597503
xn set to 0.57961233550338
xn set to 0.56011546136109
xn set to 0.57114311508018
xn set to 0.56487934739105
xn= 0.56487934739105 in 14 iterazioni
```

(D4) fine