Hasky Plotter

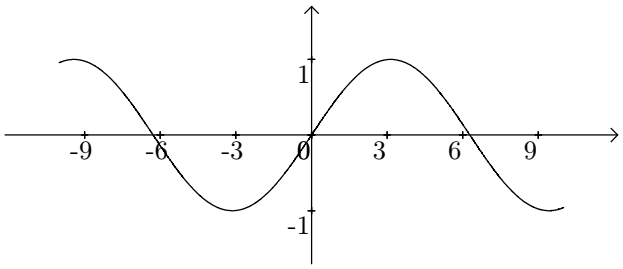
(a small function-plotter written in Haskell for $\mathrm{T_EX_{MACS}}$)

1] Cartesian single curve (function of variable x only)

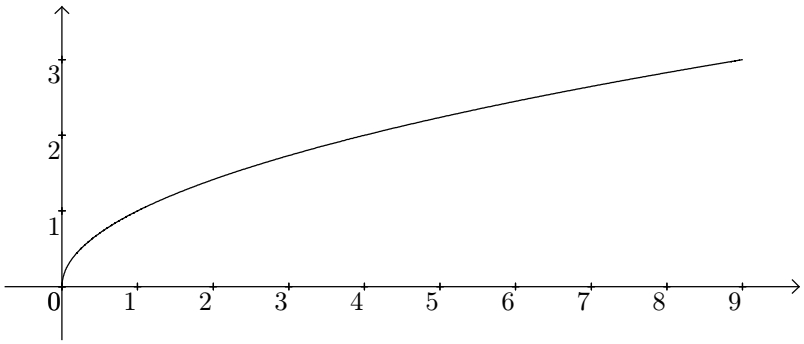
plot f(x) (x_1,x_2)

Examples:

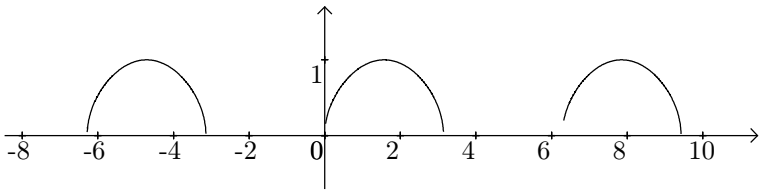
```
HP] plot sin(x) (-3,3)
```




```
HP] plot sqrt(x) (0,9)
```




```
HP] plot sqrt(sin(x)) (-7,10)
```

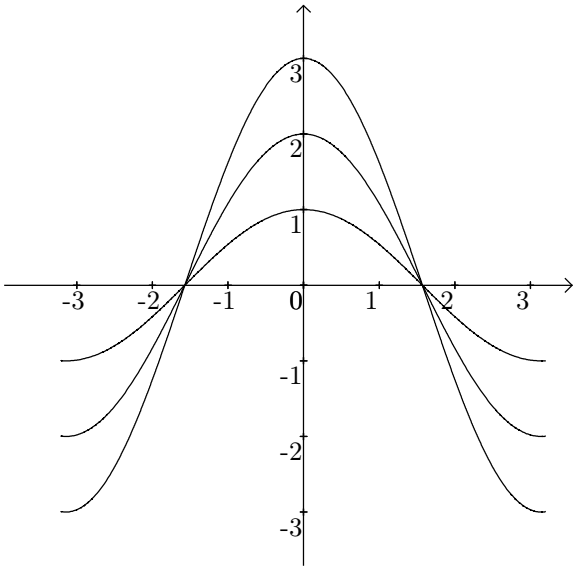


2] Set of n cartesian curves (functions of variable x only)

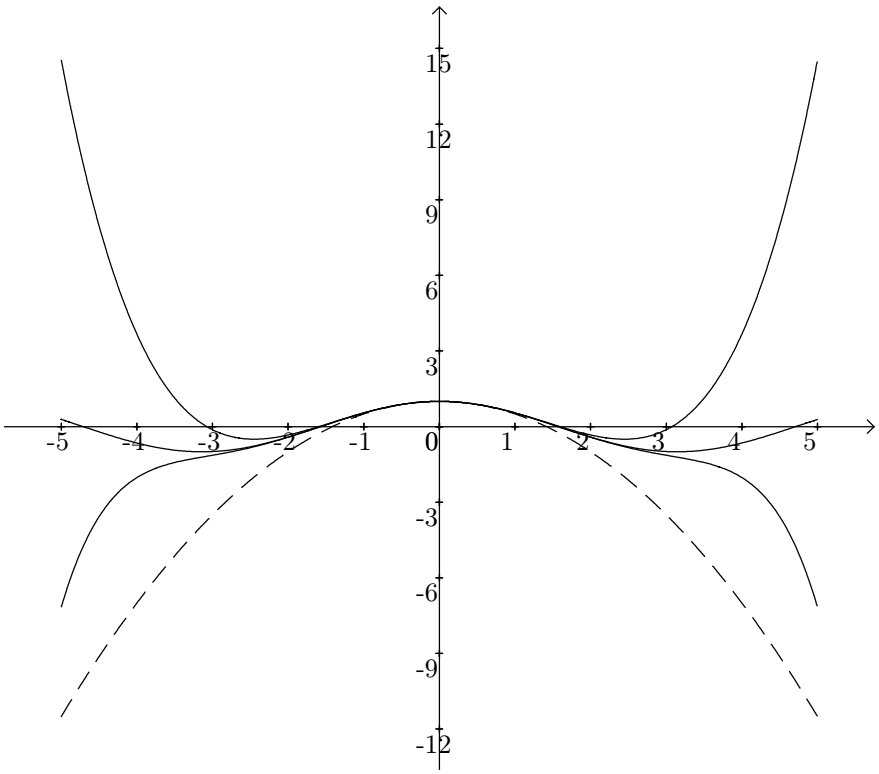
plots $f_1(x)|f_2(x)|...|f_n(x)$ (x_1,x_2)

Examples

```
HP] plots cos(x)|2*cos(x)|3*cos(x) (-3.2,3.2)
```




```
HP ] plots 1-x^2/2|1-x^2/2+x^4/24|1-x^2/2+x^4/24-x^6/720|cos(x) (-5,5)
```

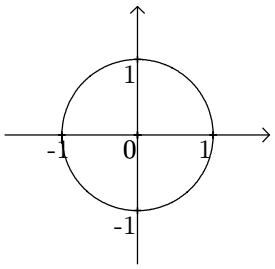


3] **Parapetric single curves** (x and y functions of parameter t only)

paramplot x(t),y(t) (t_1,t_2)

Example

```
HP] paramplot cos(t),sin(t) (-3.1,3.2)
```

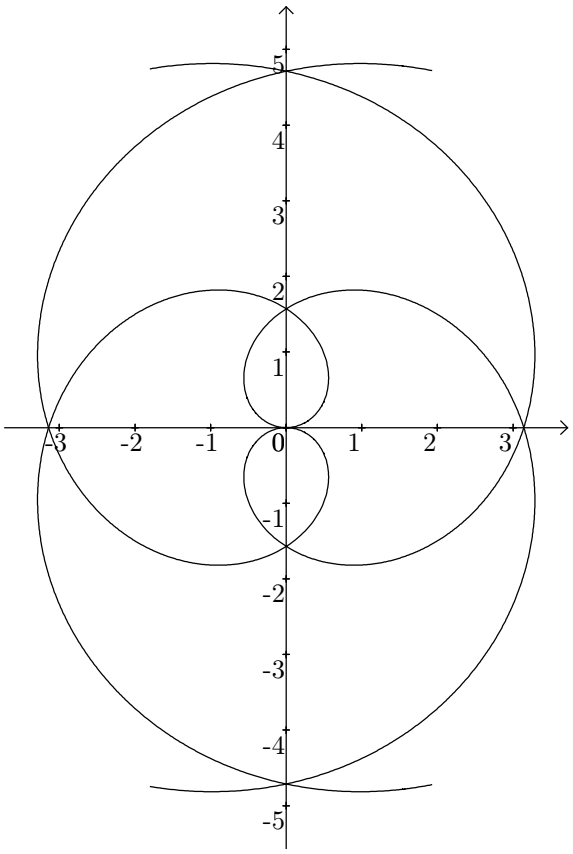


4] **Set of n parapetric curves** (x and y functions of parameter t only)

paramplots x_1(t),y_1(t)|x_2(t),y_2(t)|...|x_n(t),y_n(t) (t_1,t_2)

Example

```
HP] paramplots t*cos(t),t*sin(t)|t*cos(t),(-t)*sin(t) (-5,5)
```

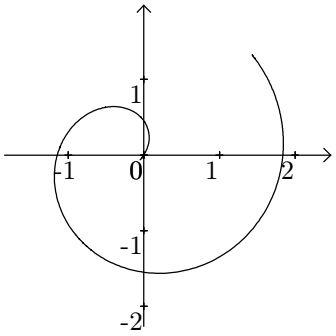


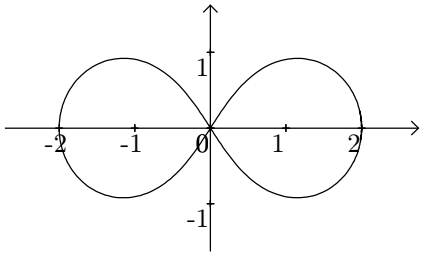
5] Polar (parapetric) single curves (r and θ as functions of t only)

polarplot r(t),θ(t) (t_1,t_2)

Exemples

```
HP] polarplot log(t),t (0,6.3)
HP] polarplot 2*cos(t),sin(t) (0,6.3)
```



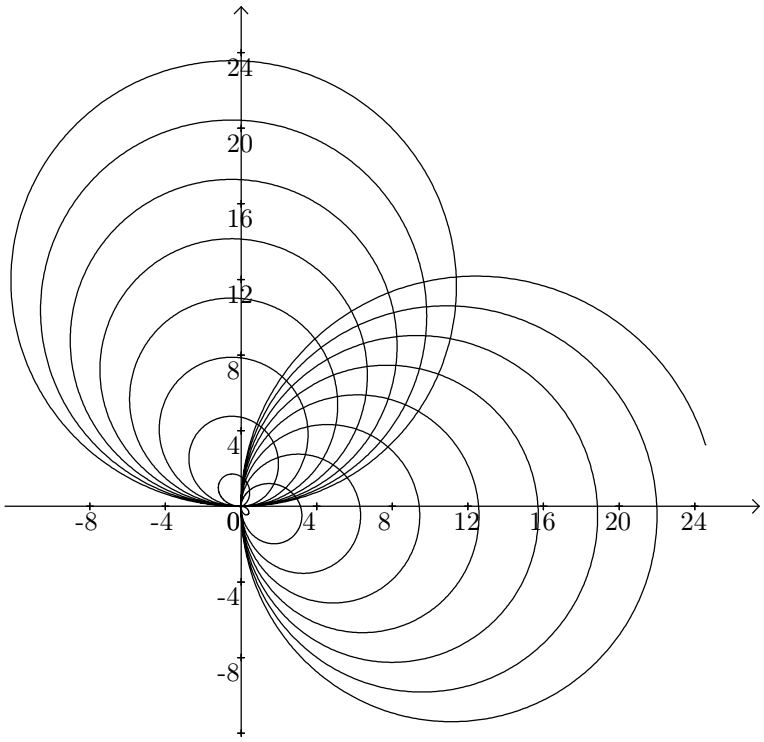


6] Set of n polar (parapetric) curves (r and θ as functions of t only)

polarplots r_1(t),y_1(t)|x_2(t),y_2(t)|...|x_n(t),y_n(t) (t_1,t_2)

Example

```
HP] polarplots t*sin(t),t|t*cos(t),-t (0,25)
```

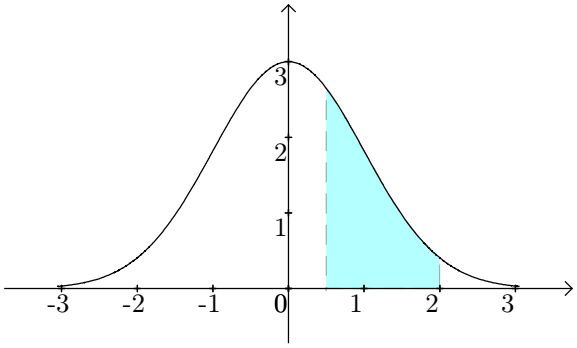


8] Surface between an curve, the $x-$axis and the verticals at $x=a$ and $x=b$.

integral f(x) (x_1,x_2) (a,b)

Example

```
HP] integral 3*exp(-x^2/2) (-3,3) (0.5,2)
```



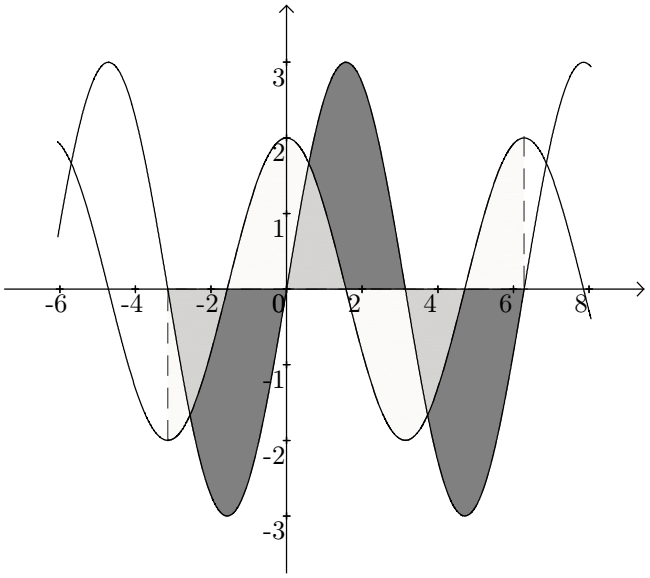
9] Sets of n cartesian curves and m surfaces (generalisation of plots and integral)

plotsMixt $f_1(x)|f_2(x)|...|f_n(x)$ $g_1(x)|g_2(x)|...|g_m(x)$ (x_1,x_2) (a,b)

as a result, hp will plot the curves of $\{f_i(x)\}_{i=1...n}$ and the surfaces of $\{g_i(x)\}_{i=1...m}$.

Example

```
HP] plotsMixt 3*sin(x)|2*cos(x) 3*sin(x)|2*cos(x) (-6,8) (-3.14,6.28)
```



Using Suffixes:

We can add the suffixes **C** and **InRange** to most of the functions introduced below.
C : For considering the curve as **c**losed (using *cspline* of TeXmacs in place of *spline*)
InRange : For specificying the interval of the images.

Notices:

i) The use of **C** doesn't test if the curve is actually closed. It will just force it to be.

ii) The use of **InRange** requires the parameters (ymin,ymax) to specify the range.
This parameter, a couple $(y_{\min}, y_{\max})$, has to be added at the very end of the command.

For example:
plot "sin(x)" "(-3,3)" would become : plot**InRange** "sin(x)" "(-3,3)" **"(-2,2)"**

iii) **C** and **InRange** can be used together, (only) by adding the composite suffixe **CInRange**.

The table below summerizes the possible combinations

instruction	+ suffixe		
	C	InRange	both *C* & InRange
plot	(∗)	plot*t*InRange	
plot**s**		plotsInRange	
integral		integral**s**InRange	
plot**s**Mixt		plot**s**MixtInRange	
paramplot	paramplot*C*	paramplotInRange	paramplot*C*InRange
paramplot**s**	paramplot**s***C*	paramplot**s**InRange	paramplot**s***C*InRange
polarplot	polarplot*C*	polarplotInRange	polarplot*C*InRange
polarplot**s**	polarplot**s***C*	polarplot**s**InRange	polarplot**s***C*InRange

(∗) C is not implemented for cartesian curves, as it rarely make sense to consider such a curve as closed. In order to show the surface between the curve and the $x-$axis, it is more convenient to use Integral (see [8])

Example with the suffixe **InRange**

Using plots with a set $\{f\}$ of functions for $x_1 < x < x_2$ will draw
the curves considering $\min(\{f\})_{\{x\}} \leqslant y \leqslant \max(\{f\})_{\{x\}}$.

That may be not very convenient if the functions of set $\{f\}$ have ranges of different orders.

We can improve the representation by specificying another range. (Figure 2 : $-5 \leqslant y \leqslant 5$)

```
HP] plots sin(x)|x|x^3-3*x (-3,3)
```

```
HP] plotsInRange sin(x)|x|x^3-3*x (-3,3) (-5,5)
```

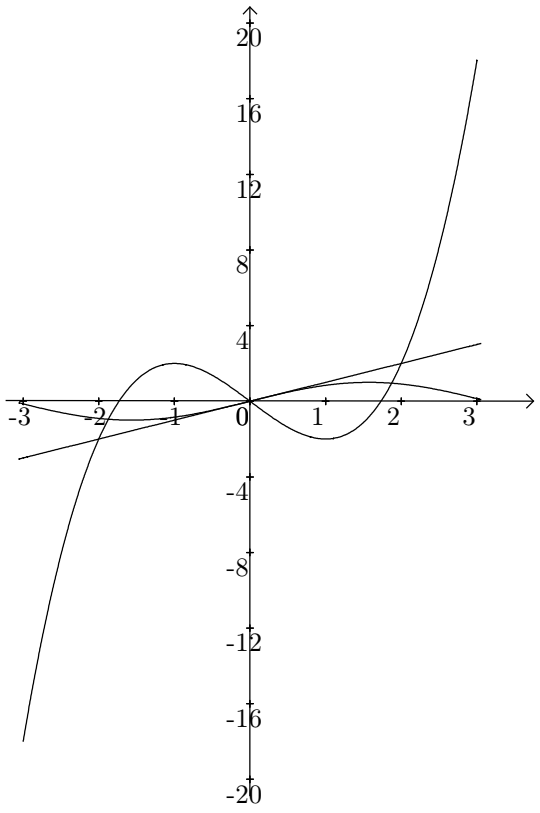


fig.1

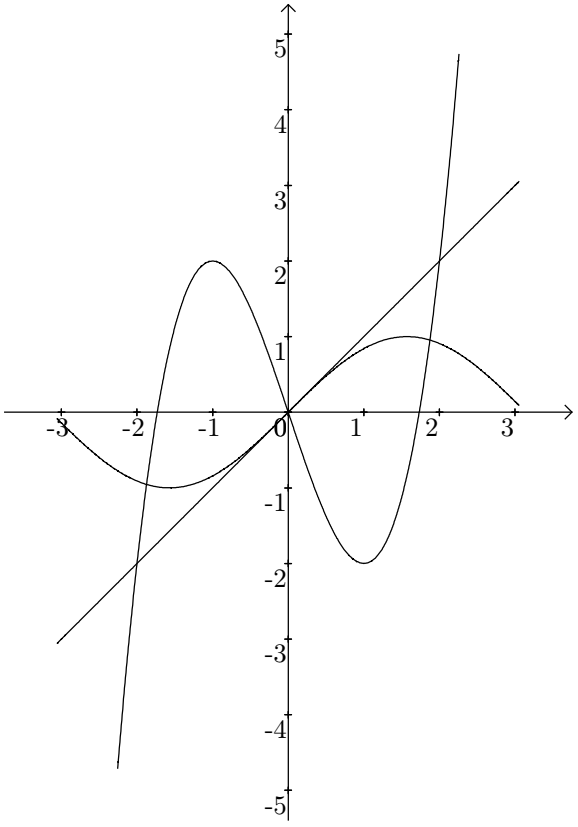
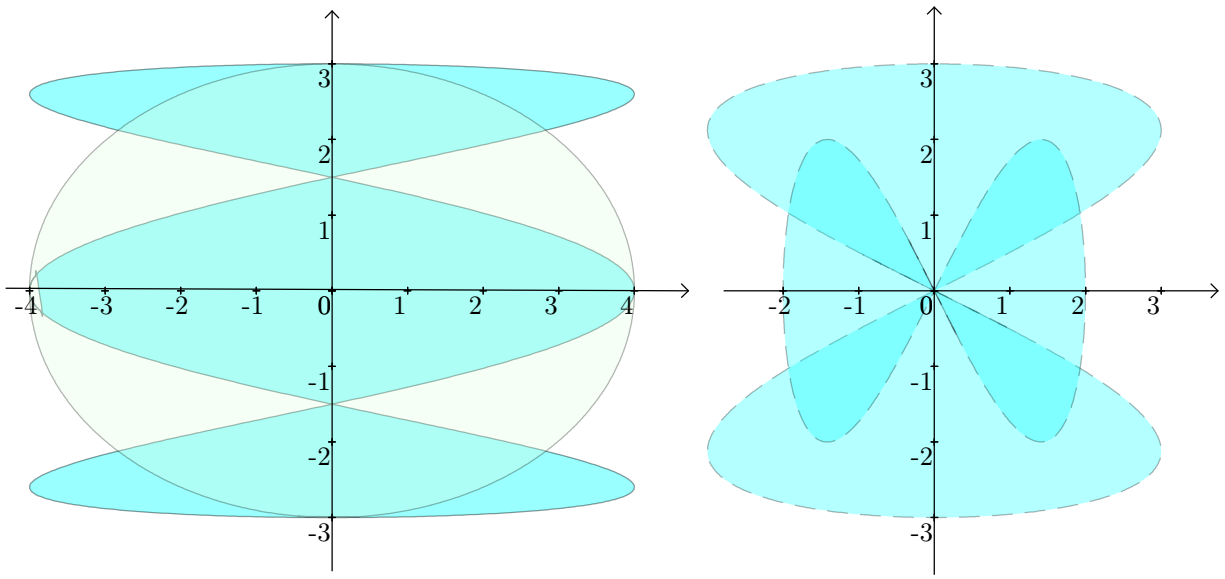


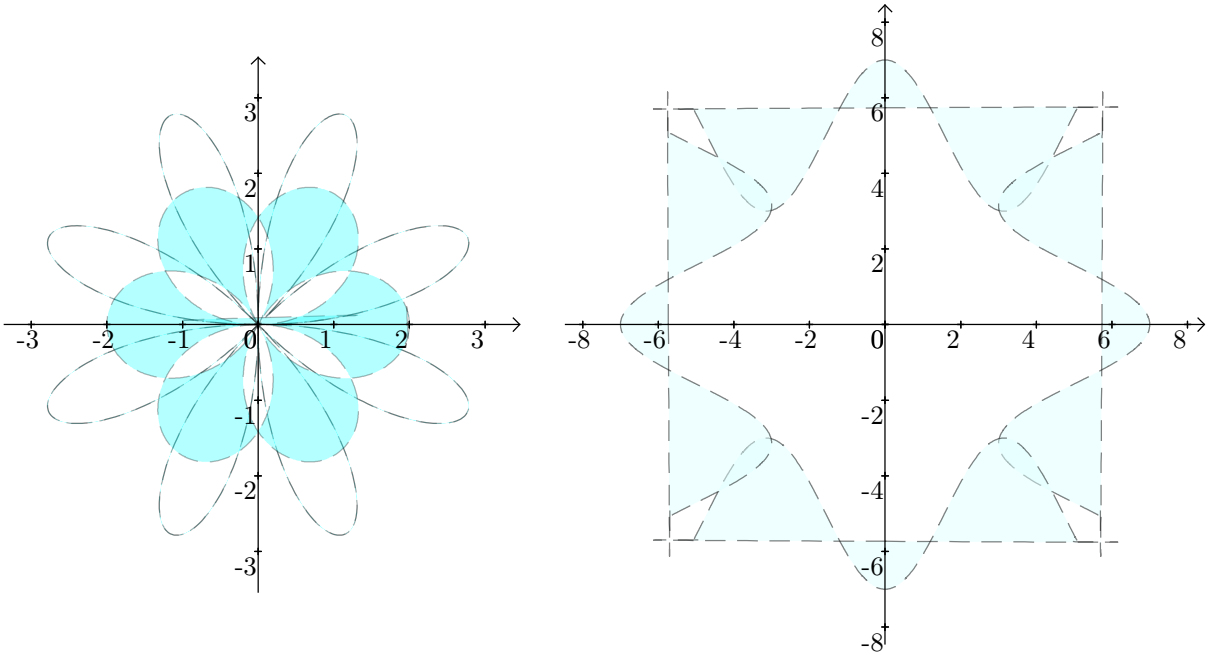
fig.2

Examples with the suffix C

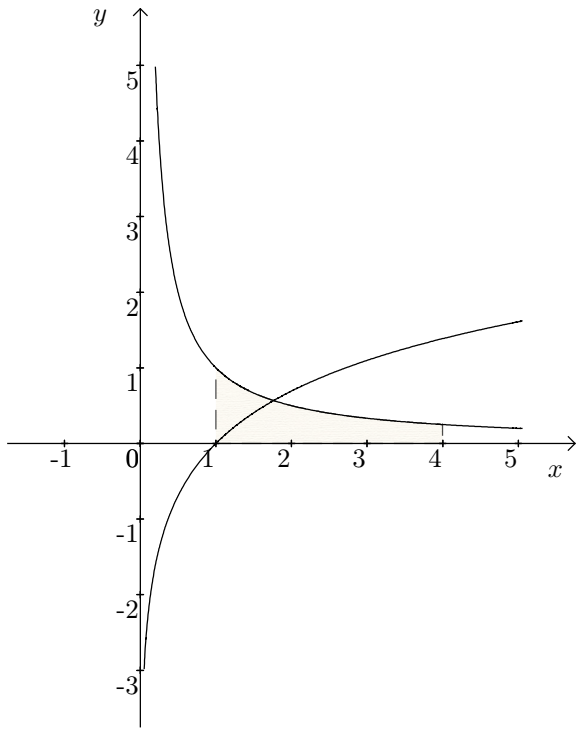
```
HP]  paramplotsC 4*cos(3*t),3*sin(t)|4*cos(t),3*sin(t)  (-3.14,3.14)
HP]  paramplotsC 3*sin(2*t),3*sin(t)|2*sin(t),2*sin(2*t) (-3.2,3.2)
```




```
HP]  polarplotsC 3*sin(6*t),t|2*cos(1.5*t),t (0,12.57)
```



```
HP]  plotsMixtInRange log(x)|1/x  1/x  (-1,5) (1,4) (-3,5)
```




```
HP]  plotsMixtInRange (x-2)^2/4+1|1|log(x+1)/log(3) (x-2)^2/4+1|1 (0,2) (0,2)
     (0,2)
```

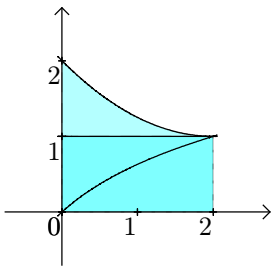


fig.1

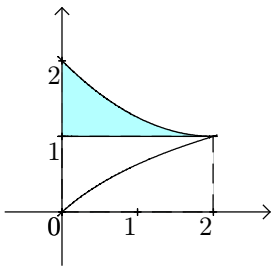


fig.2

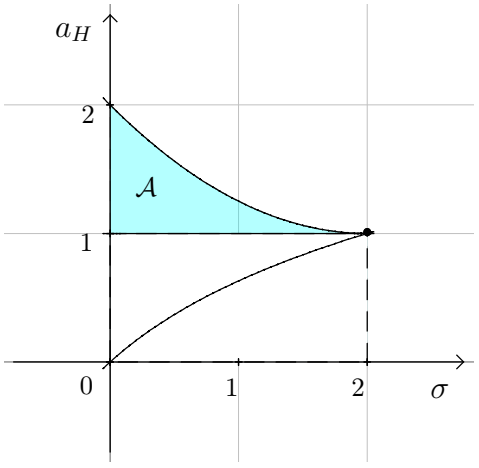


fig.3

fig.2 is produced from fig.1 by colouring one element of the surface in white, with opacity 100%.
fig.3 is produced from fig.2 by changing the unit (1.7cm) and adding some text.