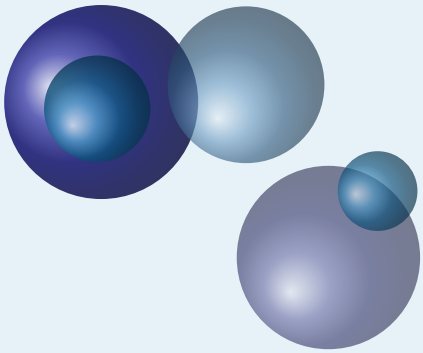
tkz-fn 1.02

AlterMundus



Alain Matthes

November 26, 2026 Documentation V.1.02

http://altermundus.fr

tkz-fn

AlterMundus

Alain Matthes

tkz-fn.sty (v1.02) is a package for drawing graphs of functions in 2D as simply as possible, with Ti*k*Z and Lua. It is the successor of `tkz-fct`: `gnuplot`, `fp.sty` and `tkz-base` are no longer needed, every computation is done by Lua and the document is compiled with `LuaLaTeX`. The drawings are the kind used in high school mathematics; the presentation (French or American conventions, "box" axes as with `pgfplots`) is chosen with options.

☞ I would like to thank `Till Tantau` for the wonderful tool `TikZ`, and `Michel Bovani` for the `fourier` fonts.

☞ I would also like to thank `David Arnold`, `Pablo González Luengo`, `Wolfgang Büchel`, `John Kitzmiller` and `Gaétan Marris` for their help with `tkz-fct`; the examples of this documentation come from it.

☞ You will find many examples on my website: altermundus.fr

Please send your comments and bug reports to: Alain Matthes.

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version.

Contents

1 Overview 6
1.1 Why a new package? . . . 6
1.2 Two coordinate spaces . . . 6

2 Installation and compilation 7
2.1 Files . . . 7
2.2 Loading . . . 7
2.3 Compilation . . . 7
2.4 Writing functions . . . 7

3 The coordinate system 8
3.1 Initialization: \tkzFnInit . . . 8
3.2 Grid: \tkzFnGrid . . . 8
3.3 Axes: \tkzFnAxes, \tkzFnAxisX, \tkzFnAxisY . . . 9
3.4 Clipping: \tkzFnClip . . . 9
3.5 Unit vectors: \tkzFnRep . . . 9
3.6 Text: \tkzFnText . . . 10
3.7 Arrows at both ends: arrows . . . 10
3.8 "Box" axes: box . . . 12

4 Drawing a function 15
4.1 Defining a function: \tkzFnDef . . . 15
4.2 Drawing a function: \tkzFnDraw . . . 15
4.3 Option: samples . . . 15
4.4 Options: xstep, ystep . . . 16
4.5 Changing xstep and ystep . . . 16
4.6 ystep and constant functions . . . 17
4.7 Linear functions . . . 18
4.8 Subgrid . . . 18
4.9 Text and points . . . 19

5 Points 20
5.1 A point on a curve: \tkzFnDefPoint . . . 20
5.2 Example . . . 21
5.3 Option draw . . . 21
5.4 Points without the curve: \tkzFnMarkPoints . . . 22
5.5 Points without worrying about coordinates . . . 23
5.6 Points with two functions . . . 24
5.7 Coordinates of a point: \tkzFnShowCoord and \tkzFnPrintCoord . . . 24

6 Naming the curves 26
6.1 Adding a label . . . 26

7 Tangent lines 27
7.1 Drawing a tangent line: \tkzFnTangent . . . 27
7.2 Tangent line when xstep and ystep are not 1 . . . 27
7.3 The options kl, kr and draw . . . 28
7.4 Tangent line with two functions . . . 28
7.5 A few tangent lines . . . 30
7.6 One-sided tangent lines . . . 30
7.7 One-sided tangent lines: Lorenz curve . . . 31
7.8 A family of tangent lines . . . 32

7.9 Tangent lines without the curve . . . 33
7.10 Point of tangency given by an expression . . . 35

8 Zeros and extrema 36
8.1 Defining the zeros: \tkzFnZeros . . . 36
8.2 Tangent lines at the zeros and at the extrema . . . 36
8.3 Discontinuous function . . . 37
8.4 Limits . . . 38

9 Areas 39
9.1 Drawing an area: \tkzFnArea . . . 39
9.2 Birth of the natural logarithm . . . 39
9.3 A simple area . . . 40
9.4 Area and hatching . . . 42
9.5 Area between two curves, in color . . . 43
9.6 Area between two curves, with hatching . . . 43
9.7 Area between two curves: option where . . . 44
9.8 Area between two curves: Lorenz curves . . . 45
9.9 Mixing styles . . . 46
9.10 Level curves . . . 47

10 Riemann sums 48
10.1 A Riemann sum . . . 48
10.2 Methods inf and sup . . . 49
10.3 Method mid . . . 50
10.4 Values of the sums . . . 50

11 Special lines 51
11.1 Vertical lines: \tkzFnVLines . . . 51
11.2 A vertical line . . . 51
11.3 Vertical lines . . . 51
11.4 Vertical lines at multiples of pi . . . 52
11.5 Horizontal lines: \tkzFnHLines . . . 52
11.6 Horizontal asymptote . . . 52
11.7 Horizontal lines . . . 54
11.8 Horizontal and vertical asymptotes . . . 55

12 Parametric curves 56
12.1 Parametric curve, example 1 . . . 56
12.2 Parametric curve, example 2 . . . 57
12.3 Parametric curve, example 3 . . . 58
12.4 Parametric curve, example 4 . . . 59
12.5 Parametric curve, example 5 . . . 60
12.6 Parametric curve, example 6 . . . 61
12.7 Parametric curve, example 7 . . . 62

13 Polar curves 63
13.1 Polar equation, example 1 . . . 63
13.2 Polar equation, example 2 . . . 63
13.3 Polar equation: heart . . . 65
13.4 Polar equation, example 4 . . . 65
13.5 Polar equation: cannabis curve . . . 66
13.6 Scarabaeus curve . . . 67

14 Implicit curves 68
14.1 Conics . . . 68
14.2 Lemniscate of Bernoulli and folium of Descartes . . . 68
14.3 An elliptic curve . . . 69
14.4 Level curves . . . 70
14.5 Discontinuous function . . . 70
14.6 Window, units and smoothing . . . 71
14.7 Limits of the method . . . 71

15 Symbols 72

16 Writing the results: American or French format 73
16.1 Choosing the format . . . 73
16.2 Writing a result: \tkzFnPrint . . . 73
16.3 Example . . . 74
16.4 Tick labels in French format . . . 74

17 With tkz-elements 75
17.1 In Lua . . . 75

18 A few examples 76
18.1 TikZ and tkz-fn . . . 76
18.2 Lorenz curves . . . 77
18.3 Exponential curve . . . 78
18.4 Logarithmic axis . . . 78
18.5 A bit of everything . . . 80
18.6 Interpolation . . . 81
18.6.1 The code . . . 81
18.6.2 The figure . . . 81
18.7 Van der Waals curves . . . 83
18.7.1 Variation table . . . 83
18.7.2 First curve, b=1 . . . 84
18.7.3 Second curve, b=1/3 . . . 85
18.7.4 Third curve, b=32/27 . . . 86
18.8 Critical values . . . 87
18.8.1 Van der Waals curves . . . 87
18.8.2 Van der Waals curves (continued) . . . 88

19 Examples with the packages alterqcm and tkz-tab 89
19.0.1 First graph . . . 90
19.0.2 Second graph . . . 91

20 From tkz-fct to tkz-fn 92
20.1 Correspondence of the macros . . . 92
20.2 What changes . . . 92

21 List of the macros 93
21.1 Styles . . . 94

Index 95

1 Overview

1.1 Why a new package?

tkz-fct delegated the computation of the points of the curves to **gnuplot**, and the computation of values and slopes to **fp.sty**. You had to install **gnuplot**, allow **shell-escape**, write each function with two syntaxes and often rewrite polynomials in Horner form to avoid the overflows of **fp**.

tkz-fn replaces all this with **Lua**, which is part of **LuaLaTeX**:

- the points of the curves, the values, the derivatives and the areas are computed by Lua in double precision;
- a function is defined only once, with the syntax of Lua (`x^2`, `sin(x)`, `exp(x)`);
- no **gnuplot**, no **shell-escape**, no auxiliary file;
- **tkz-fn** depends only on Ti*k*Z; it uses neither **tkz-base** nor **tkz-euclide**;
- numerical results are written in the American or in the French way;
- the axes can be drawn as in school textbooks, with arrows at one or both ends, or as a "box" as with **pgfplots**;
- a bridge makes the points defined by **tkz-fn** available to **tkz-elements**.

All the macros begin with **\tkzFn**, so there is no conflict with **tkz-fct**, **tkz-base** or **tkz-euclide**.

1.2 Two coordinate spaces

A coordinate system is defined with **\tkzFnInit**, using the options **xmin**, **xmax**, **xstep**, **ymin**, **ymax** and **ystep**. **xstep** is the value represented by one centimeter on the x-axis, **ystep** the value represented by one centimeter on the y-axis.

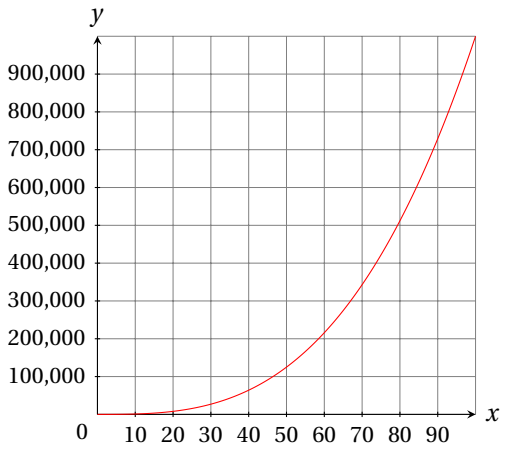
- Every computation is done in the units of the coordinate system: $f(x)$, the slope of a tangent line, an area, a Riemann sum, the coordinates of a point.
- The drawing is done in centimeters. The conversion happens at the last moment:

$$X = \frac{x}{\text{xstep}} \qquad Y = \frac{y}{\text{ystep}}$$

So you no longer write \x instead of x when **xstep** is not 1. For example, let us draw the graph of the function f defined by:

$$0 \leq x \leq 100 \text{ and } f(x) = x^3$$

The values of $f(x)$ lie between 0 and 1,000,000. With **xstep=10** and **ystep=100000**, the axes are about 10 cm long (without scaling).




```
\begin{tikzpicture}[scale=.5]
  \tkzFnInit[xmax=100,xstep=10,
             ymin=0,ymax=1000000,
             ystep=100000,xmin=0]
  \tkzFnGrid
  \tkzFnAxes[decimals=0]
  \tkzFnDraw[color=red,
             domain=0:100]{x^3}
\end{tikzpicture}
```

2 Installation and compilation

2.1 Files

The package consists of two files:

- **tkz-fn.sty**: the LaTeX interface (macros and options);
- **tkz_fn.lua**: the Lua code (computations and generated Ti*k*Z code).

Both files must be placed in a folder known to TeX, for example **texmf/tex/latex/tkz-fn**, or in the folder of the document.

2.2 Loading

```
\usepackage{tkz-fn}          % format according to the language of babel
\usepackage[us]{tkz-fn}      % American format of the results
\usepackage[fr]{tkz-fn}      % French format of the results
```

Ti*k*Z and its library **plotmarks** are loaded automatically. Some Ti*k*Z options used in the examples need other libraries: `\usetikzlibrary{arrows,patterns}`.

2.3 Compilation

The document must be compiled with **LuaLaTeX**. The **shell-escape** option is not needed and no auxiliary file is created.

2.4 Writing functions

Expressions are Lua expressions. The variable is `x` for a function, `t` for a parametric or a polar curve.

Lua	old syntax (gnuplot)	description
`x^3, x^0.5`	`x**3, x**0.5`	power
`1/3`	`1./3`	quotient (always decimal)
`ln(x)` or `log(x)`	`log(x)`	natural logarithm
`log10(x)`	`log10(x)`	common logarithm
`exp(x), e`	`exp(x), exp(1)`	exponential, number e
`sqrt(x), abs(x)`	`sqrt(x), abs(x)`	square root, absolute value
`sin, cos, tan`	same	radians
`asin, acos, atan`	same	inverse functions
`sinh, cosh, tanh`	same	hyperbolic functions
`pi`	`pi`	π
`floor, ceil, sign, mod(a,b)`	`floor, ceil, sgn, %`	
`cbrt(x), root(x,n)`		cube root, n-th root

☞ A function defined with **\tkzFnDef** can be used in another expression: `2*f(x)+1`. A global Lua variable can be used as a parameter: `\directlua{a = 0.5}` then `a*x^2`.

☞ The functions `diff(f,x)`, `integral(f,a,b)` and `riemann(f,a,b,n,"left")` give the derivative at a point, the integral (Simpson's rule) and a Riemann sum. They are used with **\tkzFnPrint**.

An argument between parentheses must not contain an unprotected closing parenthesis: write `(f,{1/exp(1)})`, not `(f,1/exp(1))`.

3 The coordinate system

These macros replace those of **tkz-base** (\tkzInit, \tkzGrid, \tkzAxeXY, \tkzDrawX, \tkzClip, \tkzRep, \tkzText).

3.1 Initialization: \tkzFnInit

\tkzFnInit[⟨options⟩]

options	default	definition
xmin	-5	smallest x
xmax	5	largest x
xstep	1	value represented by 1 cm on the x-axis
ymin	-5	smallest y
ymax	5	largest y
ystep	1	value represented by 1 cm on the y-axis
xorigin	0	x placed at $X = 0$ (a year, for example)
yorigin	0	y placed at $Y = 0$
enlarge	0 (0.1 without value)	enlarges the window by a fraction of its size
enlarge x, enlarge y	enlarge	enlargement in one direction only

*Values can be expressions: xmax=2*pi.*

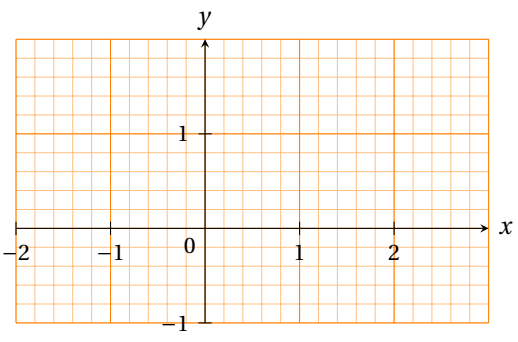
3.2 Grid: \tkzFnGrid

\tkzFnGrid[⟨options⟩](⟨x_1,y_1⟩)(⟨x_2,y_2⟩)

The region is optional; by default the grid covers the window.

options	default	definition
dx	xstep	distance between two vertical lines
dy	ystep	distance between two horizontal lines
sub	false	adds a subgrid
subdx	dx/5	spacing of the subgrid
subdy	dy/5	spacing of the subgrid

Other options are TikZ options; the style is **tkzfn grid**.




```
\begin{tikzpicture}[scale=1.25]
 \tkzFnInit[xmin=-2,xmax=3,
            ymin=-1,ymax=2]
 \tkzFnGrid[sub,color=orange]
 \tkzFnAxes
\end{tikzpicture}
```

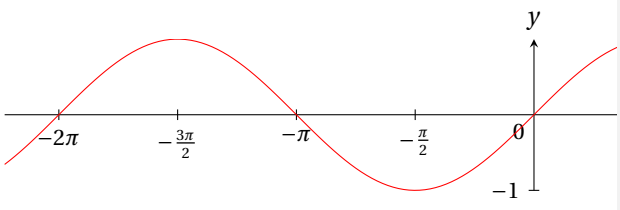
3.3 Axes: \tkzFnAxes, \tkzFnAxisX, \tkzFnAxisY

\tkzFnAxes[⟨options⟩]

Draws both axes, the tick marks and their labels. The axes cross at $(0,0)$ *if this point is in the window, at* $(x_{\min}, y_{\min})$ *otherwise. Values are written in the American or the French way (section 16).*

options	default	definition
dx, dy	xstep, ystep	distance between two tick marks
xlabel, ylabel	x, y	names of the axes
ticks	true	tick marks
labels	true	tick labels
line	true	axis lines
origin	0 or O	label of the origin (0 American, O French)
trig	0	if $n>0$, ticks of the x-axis at multiples of π/n
decimals	4	maximum number of decimal places
arrows	end	arrows: end, both or none (section 3.7)
box	false	``box'' axes (section 3.8)
style	school	school or box
tick length	3pt	length of the tick marks (box)

\tkzFnAxisX and \tkzFnAxisY draw a single axis; their options are the same with **label**, **label style** *and* **step** *instead of* **xlabel**, **ylabel** *and* **dx**. *With* **ticks=false**, *you get the equivalent of* **\tkzDrawX**.




```
\begin{tikzpicture}
  \tkzFnInit[xmin=-7,xmax=7,
             ymin=-1,ymax=1]
  \tkzFnAxisY
  \tkzFnAxisX[trig=2]
  \tkzFnDraw[red]{sin(x)}
\end{tikzpicture}
```

3.4 Clipping: \tkzFnClip

\tkzFnClip[⟨margin in pt⟩]

Restricts the rest of the drawing to the window (enlarged by a margin). As with **\clip**, *use it preferably inside a* **scope** *environment. Curves and tangent lines are clipped to the window anyway (option* **clip**).

3.5 Unit vectors: \tkzFnRep

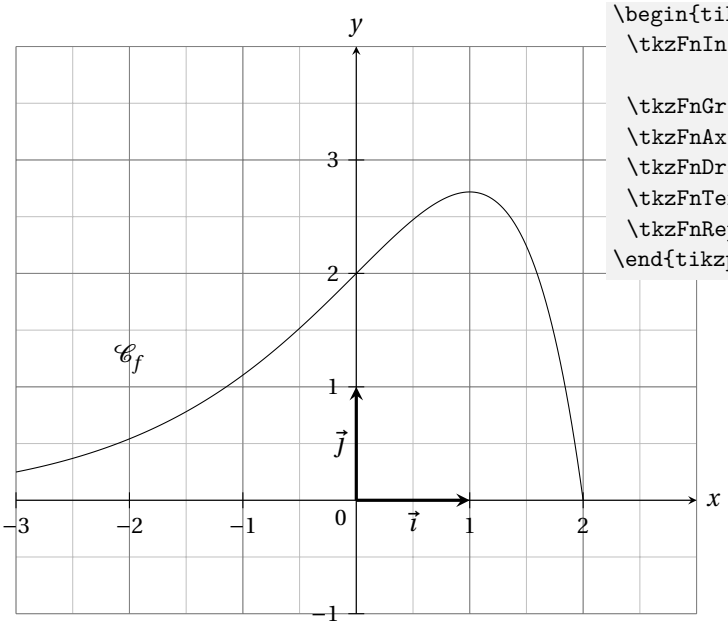
\tkzFnRep[⟨TikZ options⟩]

Draws the vectors $\vec{i}$ *and* $\vec{j}$; *the style is* **tkzfn rep**.

3.6 Text: \tkzFnText

\tkzFnText[⟨TikZ options⟩](⟨x,y⟩){⟨text⟩}

Places a text at the point with coordinates (x,y) in the coordinate system. You can also give the name of a point: ***\tkzFnText****[above](A){text}.*



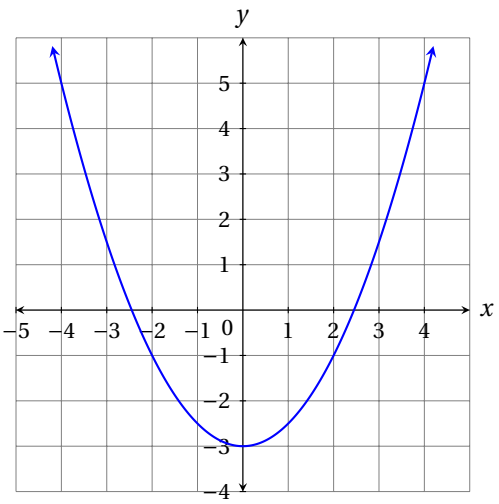

```
\begin{tikzpicture}[scale=1.5]
 \tkzFnInit[xmin=-3,xmax=3,
            ymin=-1,ymax=4]
 \tkzFnGrid[sub,subdx=.5,subdy=.5]
 \tkzFnAxes
 \tkzFnDraw[domain=-3:2]{(2-x)*exp(x)}
 \tkzFnText(-2,1.25){$\mathcal{C}_{f}$}
 \tkzFnRep
\end{tikzpicture}
```

3.7 Arrows at both ends: arrows

In the United States, textbooks and graphing calculators often put an arrow at each end of the axes. The option **arrows** of **\tkzFnAxes**, **\tkzFnAxisX** and **\tkzFnAxisY** chooses the arrows:

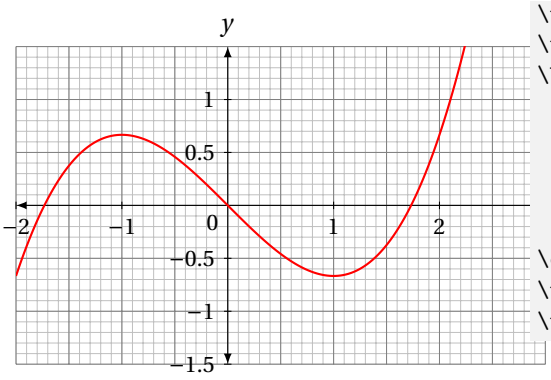
- end (default): one arrow at the positive end;
- both: an arrow at each end;
- none: no arrow.

\tkzFnSetup{arrows=both} chooses the arrows for all the following coordinate systems. The shape of the arrow tips is given by the style **tkzfn axis**.



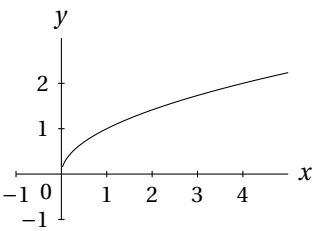

```
\begin{tikzpicture}[scale=.6]
  \tkzFnInit[xmin=-5,xmax=5,
             ymin=-4,ymax=6]
  \tkzFnGrid
  \tkzFnAxes[arrows=both]
  \tkzFnDraw[blue,thick,<->,>=stealth,
             clip=false,domain=-4.2:4.2]{x^2/2-3}
\end{tikzpicture}
```

For a whole document, with other arrow tips:




```
\tkzFnSetup{arrows=both}
\tikzset{tkzfn axis/.style={>=latex}}
\begin{tikzpicture}[scale=.7]
  \tkzFnInit[xmin=-2,xmax=3,xstep=.5,
             ymin=-1.5,ymax=1.5,ystep=.5]
  \tkzFnGrid[sub]
  \tkzFnAxes[dx=1]
  \tkzFnDraw[red,thick]{x^3/3-x}
\end{tikzpicture}
\tkzFnSetup{arrows=end}
\tikzset{tkzfn axis/.style={->,>=stealth}}
```

Without arrows:



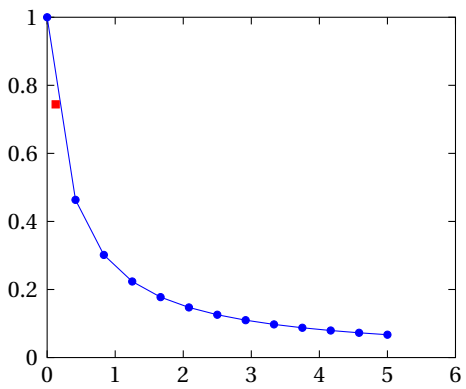

```
\begin{tikzpicture}[scale=.6]
  \tkzFnInit[xmin=-1,xmax=5,
             ymin=-1,ymax=3]
  \tkzFnAxes[arrows=none]
  \tkzFnDraw{sqrt(x)}
\end{tikzpicture}
```

3.8 ``Box'' axes: box

This presentation is usual in scientific publications and with **pgfplots**, and less common in high school textbooks. With **pgfplots** it is obtained with the **axis** environment; with **tkz-fn** an option is enough:

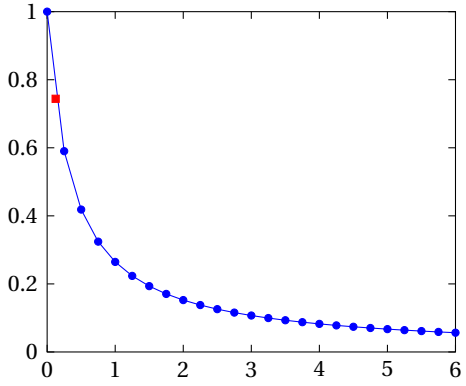
- `\tkzFnAxes[box]`: a frame surrounds the window, the tick marks point inward on the four sides, every value (including those at the edges) is written outside, there is no arrow and no origin label;
- the names of the axes are written only when they are given: `xlabel` below the frame, `ylabel` on the left, vertically;
- `\tkzFnSetup{axes=box}` chooses this presentation for all the following coordinate systems; `\tkzFnSetup{axes=school}` comes back to the usual one;
- the option **mark** of **\tkzFnDraw** puts a mark at each computed point (**samples** sets their number);
- the option **enlarge** of **\tkzFnInit** moves the curve away from the frame; the default domain is still $[x_{\min}, x_{\max}]$.

The style of the frame is **tkzfn box**. The following examples reproduce a figure made with **pgfplots**.



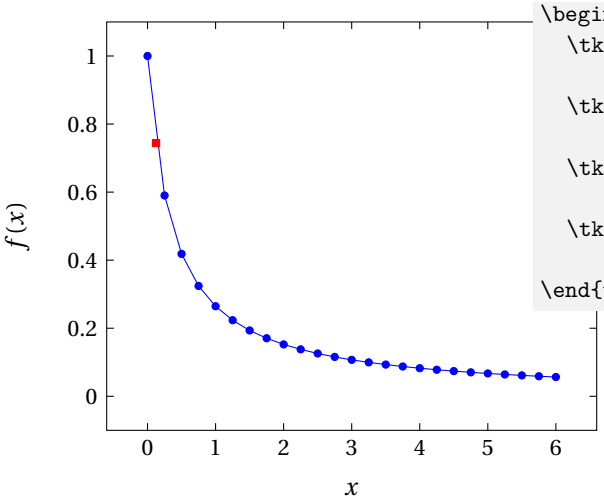

```
\tkzFnDef{f}{1/((1e9*x/11.778)*3.273e-8
              -3.273e-8+1)}
\begin{tikzpicture}[scale=.9]
  \tkzFnInit[xmin=0,xmax=6,
             ymin=0,ymax=1,ystep=.2]
  \tkzFnAxes[box]
  \tkzFnDraw[blue,mark=*,mark size=1.5pt,
             samples=13,domain=0:5](f)
  \tkzFnDrawData[red,mark=square*,
                 mark size=1.5pt]{.125/.744}
\end{tikzpicture}
```

The default domain is $[x_{\min}, x_{\max}]$:



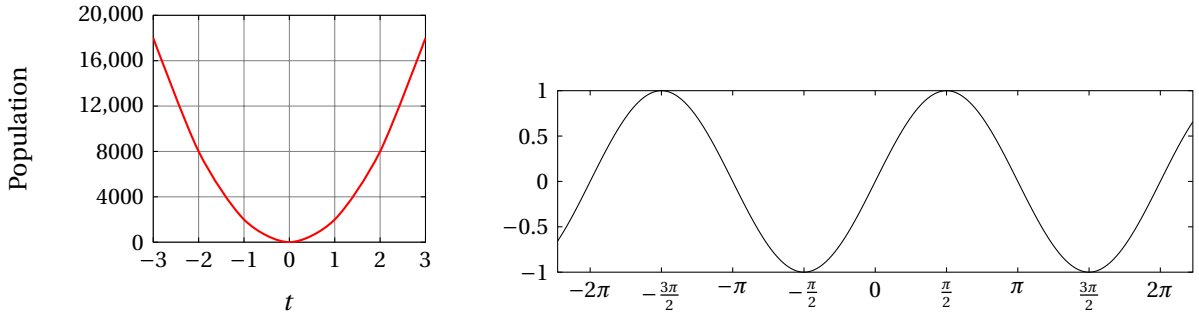

```
\begin{tikzpicture}[scale=.9]
  \tkzFnInit[xmin=0,xmax=6,
             ymin=0,ymax=1,ystep=.2]
  \tkzFnAxes[box]
  \tkzFnDraw[blue,mark=*,mark size=1.5pt,
             samples=25](f)
  \tkzFnDrawData[red,mark=square*,
                 mark size=1.5pt]{.125/.744}
\end{tikzpicture}
```

With **enlarge** (10 % by default) and the names of the axes:




```
\begin{tikzpicture}[scale=.9]
  \tkzFnInit[xmin=0,xmax=6,ymin=0,ymax=1,
             ystep=.2,enlarge]
  \tkzFnAxes[box,xlabel=$x$,
             ylabel=$f(x)$]
  \tkzFnDraw[blue,mark=*,mark size=1.5pt,
             samples=25](f)
  \tkzFnDrawData[red,mark=square*,
                 mark size=1.5pt]{.125/.744}
\end{tikzpicture}
```

With \tkzFnSetup{axes=box}, the grid and the ticks at multiples of π are used as usual; the option **smooth** is passed to Ti*k*Z.




```
\tkzFnSetup{axes=box}
\begin{tikzpicture}[scale=.6]
  \tkzFnInit[xmin=-3,xmax=3,ymin=0,ymax=20000,ystep=4000]
  \tkzFnGrid
  \tkzFnAxes[xlabel=$t$,ylabel=Population]
  \tkzFnDraw[red,thick,smooth,samples=7]{2000*x^2}
\end{tikzpicture}\qquad
\begin{tikzpicture}[scale=.6]
  \tkzFnInit[xmin=-7,xmax=7,ymin=-1,ymax=1,ystep=.5]
  \tkzFnAxes[trig=2]
  \tkzFnDraw{sin(x)}
\end{tikzpicture}
\tkzFnSetup{axes=school}
```

4 Drawing a function

Lua computes the points needed to draw the curve. Their number is set by the option **samples**. Ti*k*Z then draws the line through these points.

4.1 Defining a function: \tkzFnDef

\tkzFnDef{⟨name⟩}{⟨expression⟩}

*Defines a function of the variable x. The name is a Lua name (letters, digits, _). The function is then referred to by its name in every macro: **\tkzFnDraw**(f), **\tkzFnTangent**(f,1), **\tkzFnArea**(f), etc. In Lua, it is stored in the table FN: FN.f(2).*

arguments	example	explanation
name	f	name of the function
expression	x^3-2*x	Lua expression in x

4.2 Drawing a function: \tkzFnDraw

\tkzFnDraw[⟨options⟩](⟨name⟩) or [⟨options⟩]{⟨expression⟩}

*The function is given by its name, or directly by its expression; the option **name** then gives it a name.*

options	default	definition
domain	xmin:xmax	domain of the drawing
samples	200	number of computed points
name		name given to the function
clip	true	the curve is clipped to the window
mark=…		a mark at each computed point (**plotmarks**)
smooth		curve smoothed by TikZ

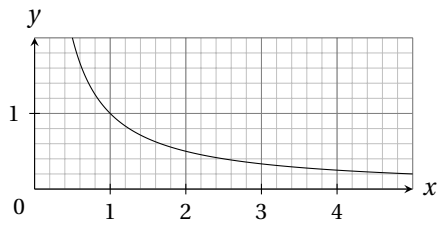
*Other options are TikZ options; the style is **tkzfn curve**.*

☞ The coordinate system must be initialized with **\tkzFnInit** before **\tkzFnDraw** is used.

☞ The curve is interrupted where the function is not defined (ln(x) for $x \leq 0$) and where it jumps from one side of the window to the other (vertical asymptote).

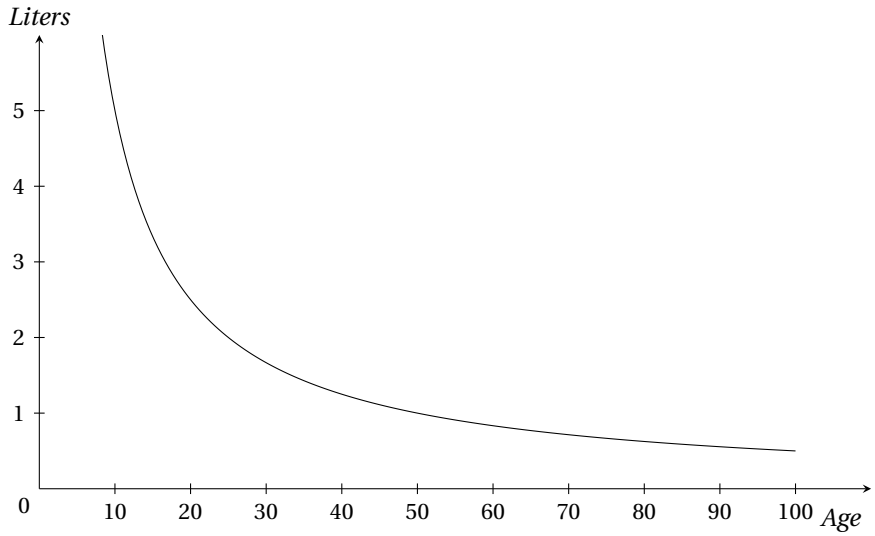
4.3 Option: samples

Two points are enough to draw a line: **samples=2**. Here 400 points are requested; the default value is 200.




```
\begin{tikzpicture}[scale=1]
    \tkzFnInit[xmin=0,xmax=5,
               ymin=0,ymax=2]
    \tkzFnGrid[sub]
    \tkzFnAxes
    \tkzFnDraw[samples=400,
               domain=.5:5]{1/x}
\end{tikzpicture}
```

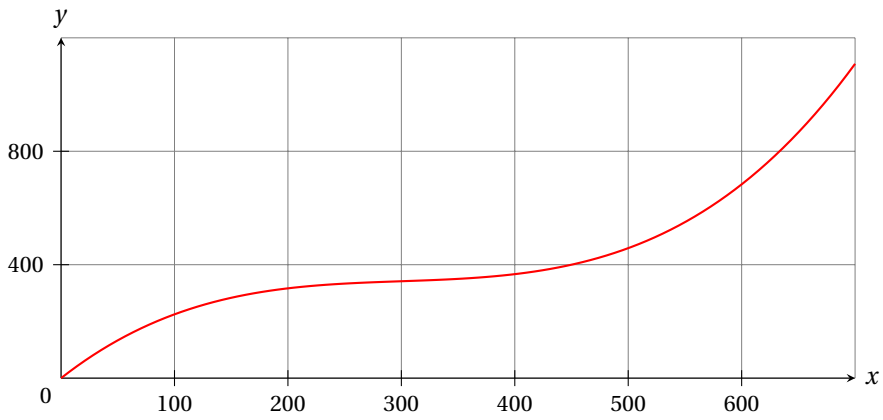
4.4 Options: xstep, ystep




```
\begin{tikzpicture}
\tkzFnInit[xmin=0,xmax=110,xstep=10,
           ymin=0,ymax=6,ystep=1]
\tkzFnAxisX[label={\textit{Age}},
            label style={below=12pt,left}]
\tkzFnAxisY[label={\textit{Liters}}]
\tkzFnDraw[domain=0.1:100]{50/x}
\end{tikzpicture}
```

4.5 Changing xstep and ystep

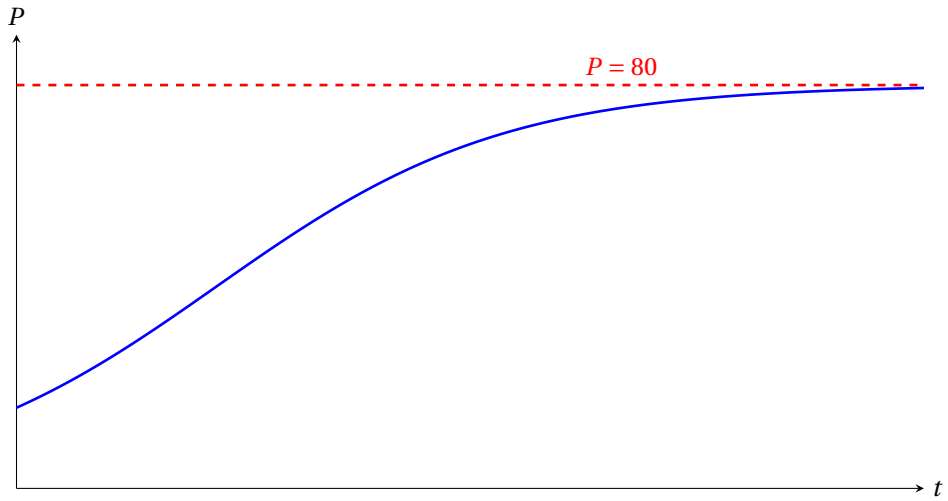
This time the domain goes from 0 to 700 and the values of the function from 0 to 1200. With **tkz-fct**, you had to write `\x` instead of x and `1./90000` instead of `1/90000`; this is no longer necessary.




```
\begin{tikzpicture}[scale=1.5]
 \tkzFnInit[xmin=0,xmax=700,xstep=100,
            ymin=0,ymax=1200,ystep=400]
 \tkzFnGrid(0,0)(700,1200)  \tkzFnAxes
 \tkzFnDraw[color=red,samples=100,line width=0.8pt,domain=0:700]
       {(1/90000)*x^3-(1/100)*x^2+(113/36)*x}
\end{tikzpicture}
```

4.6 ystep and constant functions

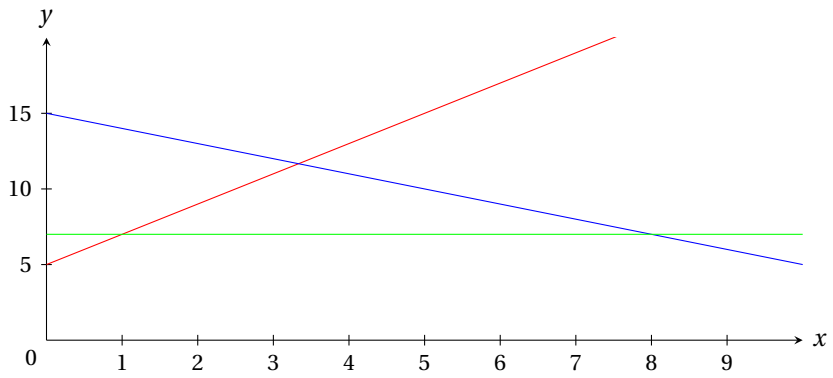
With **gnuplot**, `80/6` was an integer division and you had to write `80.`; with Lua, computations are always done with decimal numbers.




```
 \begin{tikzpicture}[scale=0.4]
 \tkzFnInit[xmin=0,xmax=30,ymin=0,ymax=90,ystep=6]
 \tkzFnAxisX[ticks=false,label=$t$]
 \tkzFnAxisY[ticks=false,label=$P$]
 \tkzFnDraw[line width=1pt,color=red,dashed,domain=0:30]{80}
 \tkzFnDraw[line width=1pt,color=blue,domain=0:30]{80/(1+4*exp(-0.21*x))}
 \tkzFnText[above,color=red](20,80){$P=80$}
\end{tikzpicture}
```

4.7 Linear functions

Two points are enough to draw a line: `samples=2`.

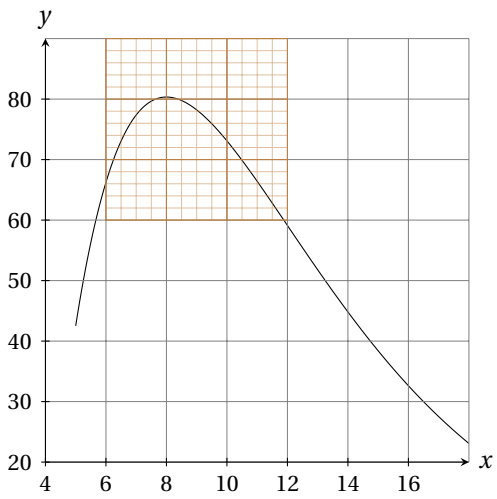



```
 \begin{tikzpicture}[]
 \tkzFnInit[xmin=0,xmax=10,ymin=0,ymax=20,ystep=5]
 \tkzFnAxes
 \tkzFnDraw[color=red,domain=0:10,samples=2]{2*x+5}
 \tkzFnDraw[color=blue,domain=0:10,samples=2]{-x+15}
 \tkzFnDraw[color=green,domain=0:10,samples=2]{7}
\end{tikzpicture}
```

4.8 Subgrid

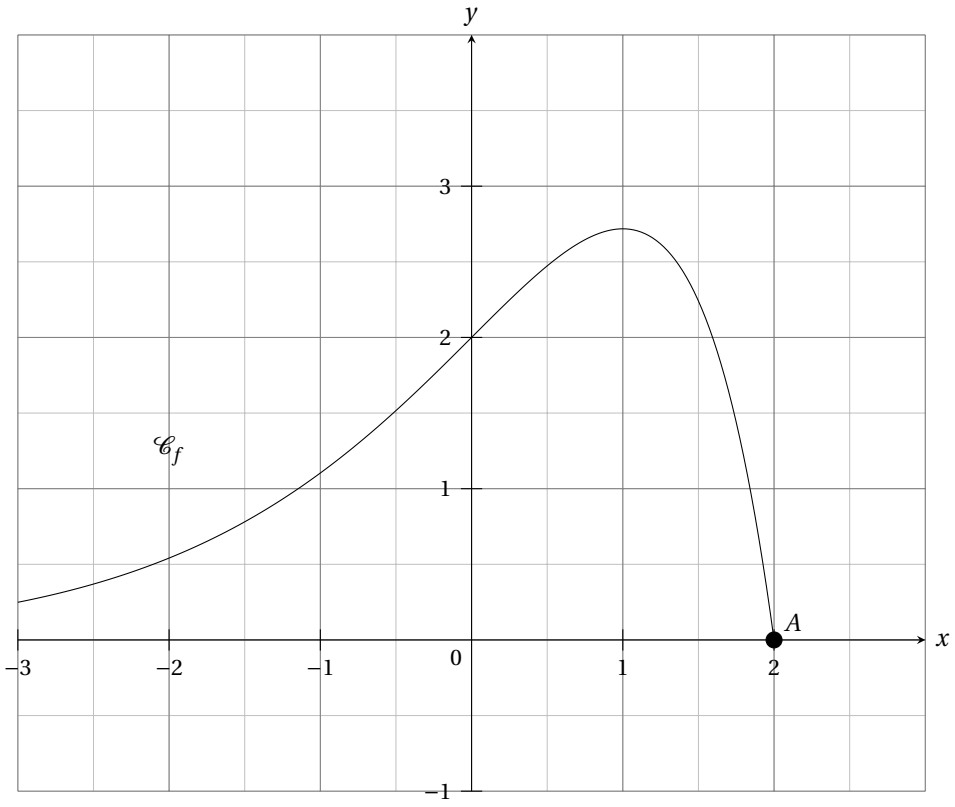
$y = (x-4)\mathrm{e}^{-0.25x+5}$

Another grid can be drawn on a part of the window.




```
\begin{tikzpicture}[scale=.8]
 \tkzFnInit[xmin=4,xmax=18,xstep=2,
            ymin=20,ymax=90,ystep=10]
 \tkzFnDraw[domain=5:18]
       {(x-4)*exp(-0.25*x+5)}
 \tkzFnGrid(4,20)(18,90)
 \tkzFnAxes
 \tkzFnGrid[sub,
            subdx=0.5,
            subdy=2,
            color=brown](6,60)(12,90)
\end{tikzpicture}
```

4.9 Text and points




```
\begin{tikzpicture}[scale=2]
 \tkzFnInit[xmin=-3,xmax=3,ymin=-1,ymax=4]
 \tkzFnGrid[sub,subdx=.5,subdy=.5]
 \tkzFnAxes
 \tkzFnDraw[domain=-3:2]{(2-x)*exp(x)}
 \tkzFnText(-2,1.25){$\mathcal{C}_{f}$}
 \tkzFnDefPointXY[draw](2,0){A}
 \tkzFnLabelPoints(A)
\end{tikzpicture}
```

5 Points

5.1 A point on a curve: \tkzFnDefPoint

\tkzFnDefPoint[⟨options⟩](⟨name of the function,x⟩){⟨name of the point⟩}

Computes the value of the function at x and defines the point $(x, f(x))$. The point is a TikZ coordinate: it is then used by its name. If **tkz-elements** *is loaded, the point is also stored in the table* **z** *(section 17).*

arguments	example	explanation
function,x	(f,0.5)	point of the graph of f with $x = 0.5$
name of the point	A	name of the coordinate

options	default	explanation
draw	false	the point is drawn
size	1.5pt	radius of the point

Other options are TikZ options used to draw the point.

\tkzFnDefPointXY[⟨options⟩](⟨x,y⟩){⟨name of the point⟩}

Defines a point by its coordinates in the coordinate system (equivalent of **\tkzDefPoint***).*

\tkzFnDrawPoints[⟨options⟩](⟨A,B,…⟩)

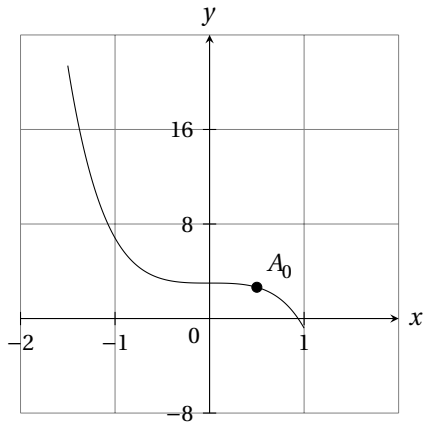
Draws points; the option **size** *gives the radius, other options are TikZ options. The style is* **tkzfn point***; it replaces* **\tkzSetUpPoint***:*

```
\tikzset{tkzfn point/.style={draw=black,fill=yellow}}
```

\tkzFnLabelPoints[⟨options⟩](⟨A,B,…⟩)

Writes the names of the points; **\tkzFnLabelPoint***[options](A){text} writes any text. Default position:* **above right***.*

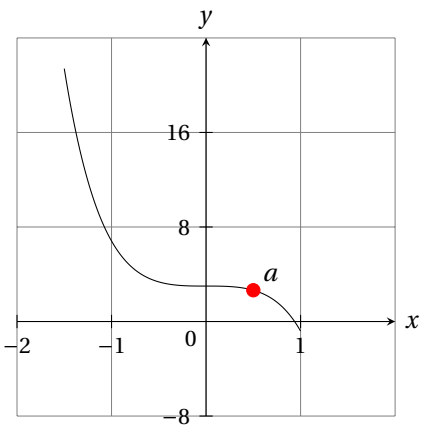
5.2 Example




```
\begin{tikzpicture}[scale=1.25]
  \tkzFnInit[xmin=-2,xmax=2,xstep=1,
             ymin=-8,ymax=24,ystep=8]
  \tkzFnGrid  \tkzFnAxes
  \tkzFnDraw[name=f,domain=-1.5:1]{3-1.3125*x^5-2.5*x^3}
  \tkzFnDefPoint(f,.5){A}  \tkzFnDrawPoints(A)
  \tkzFnLabelPoint[above right](A){$A_0$}
\end{tikzpicture}
```

5.3 Option draw

The point is defined and drawn at once.



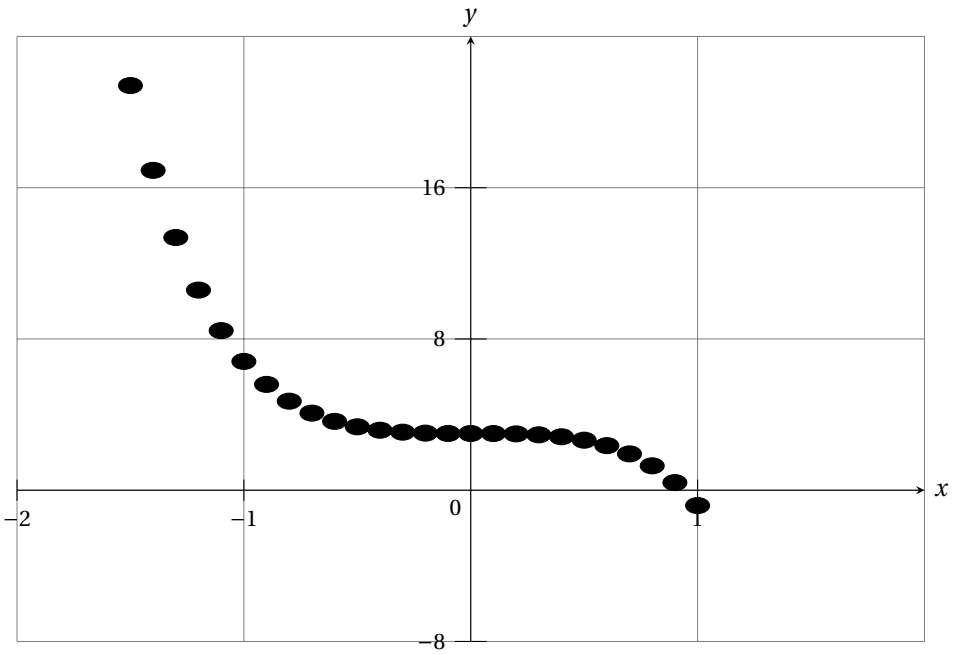

```
\begin{tikzpicture}[scale=1.25]
  \tkzFnInit[xmin=-2,xmax=2,xstep=1,
             ymin=-8,ymax=24,ystep=8]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[name=f,domain=-1.5:1]
            {3-1.3125*x^5-2.5*x^3}
  \tkzFnDefPoint[draw,red,size=2pt](f,.5){A}
  \tkzFnLabelPoint[above right](A){$a$}
\end{tikzpicture}
```

5.4 Points without the curve: \tkzFnMarkPoints

With **tkz-fct**, you had to define the macro **\tkzFctLast** with the syntax of **fp**. Now it is enough to define the function without drawing it.

\tkzFnMarkPoints[⟨options⟩](⟨name of the function⟩){⟨list of *x* values⟩}

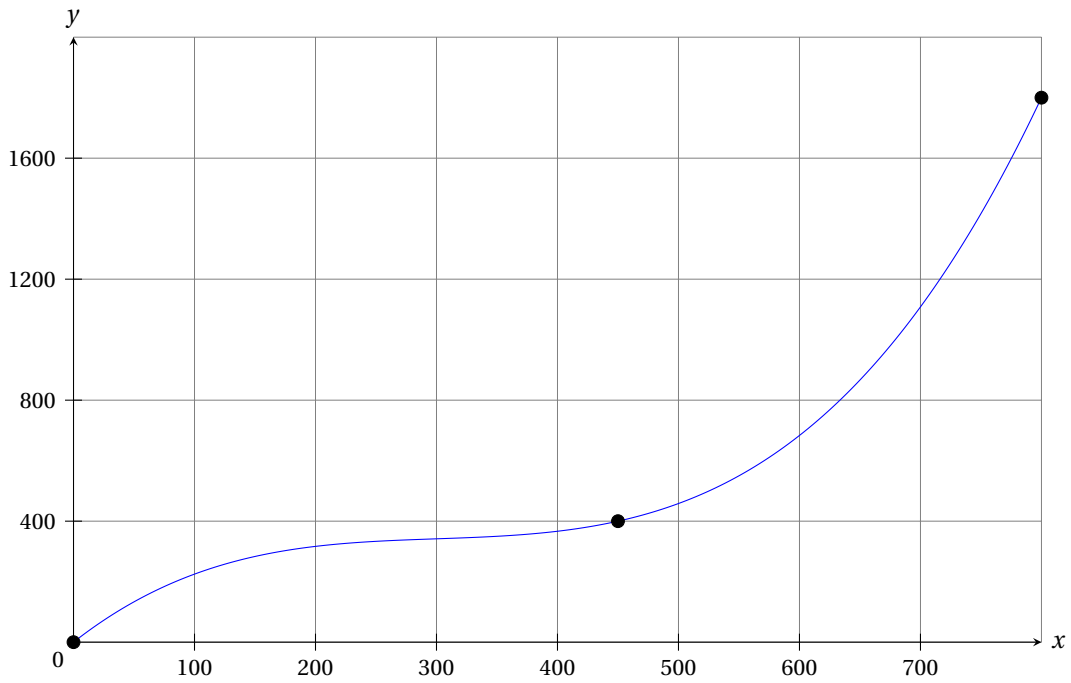
Draws the points of the graph for the given values of x; the list is written as with **\foreach**: `-1.5,-1.4,...,1`. *The points have no name.*




```
\begin{tikzpicture}[xscale=3,yscale=2]
  \tkzFnInit[xmin=-2,xmax=2,xstep=1,
             ymin=-8,ymax=24,ystep=8]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDef{f}{3-1.3125*x^5-2.5*x^3}
  \tkzFnMarkPoints(f){-1.5,-1.4,...,1}
\end{tikzpicture}
```

5.5 Points without worrying about coordinates

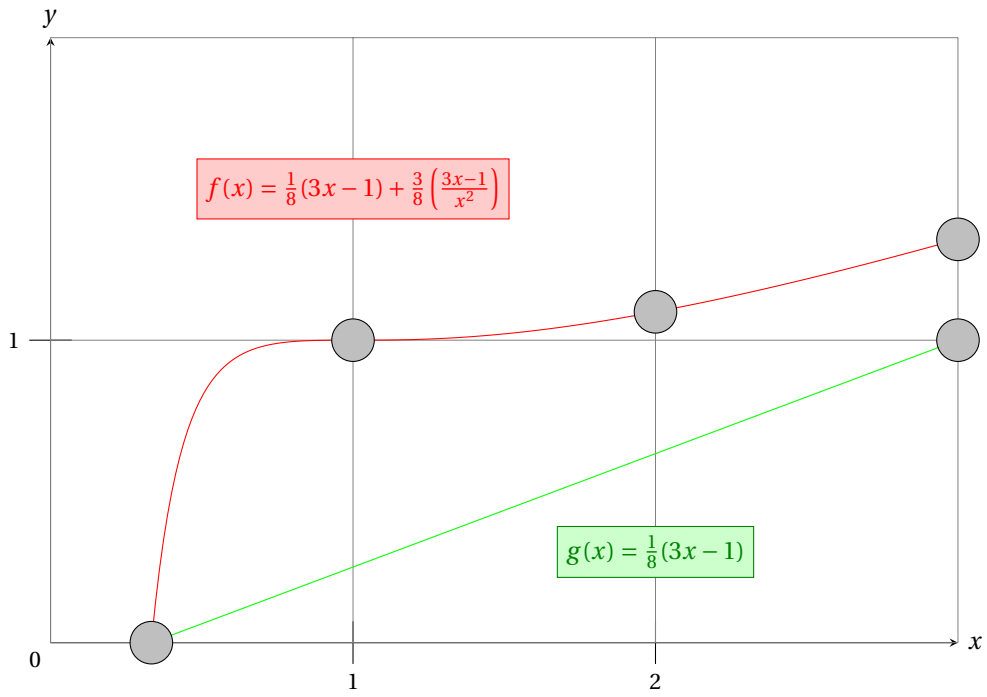
The domain goes from 0 to 800 and the values of the function from 0 to 2000. The values of x are given in the units of the coordinate system.




```
\begin{tikzpicture}[scale=1.6]
  \tkzFnInit[xmin  = 0,  xmax  = 800,
             ymin  = 0,  ymax  = 2000,
             xstep = 100,ystep = 400]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[name   = f,
             color  = blue,
             domain = 0:800]
        {(1/90000)*x^3-(1/100)*x^2+(113/36)*x}
  \tkzFnMarkPoints(f){0,450,800}
\end{tikzpicture}
```

5.6 Points with two functions

Each function has a name: there is no longer a `with` option.




```
\begin{tikzpicture}[scale=4]
  \tkzFnInit[xmin=0,xmax=3,ymin=0,ymax=2]
  \tkzFnAxes
  \tkzFnGrid(0,0)(3,2)
  \tkzFnDraw[name=f,color=red,domain=1/3:3]
            {0.125*(3*x-1)+0.375*(3*x-1)/(x*x)}
  \tkzFnDraw[name=g,color=green,domain=1/3:3]{0.125*(3*x-1)}
  \tikzset{tkzfn point/.style={draw=black,fill=lightgray}}
  \tkzFnMarkPoints[size=2pt](f){1,2,3}
  \tkzFnMarkPoints[size=2pt](g){1/3,3}
  \tkzFnText[draw,color=red,fill=red!20](1,1.5)
          {$f(x)=\frac{1}{8}(3x-1)+\frac{3}{8}
           \left(\frac{3x-1}{x^2}\right)$}
  \tkzFnText[draw,color=green!50!black,fill=green!20]
                (2,0.3){$g(x)=\frac{1}{8}(3x-1)$}
\end{tikzpicture}
```

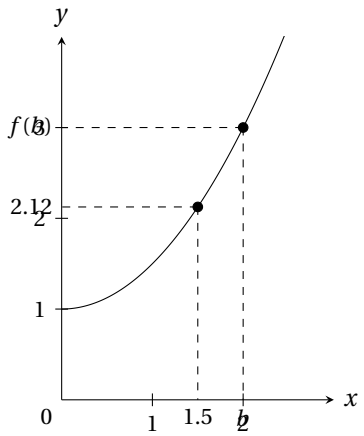
5.7 Coordinates of a point: \tkzFnShowCoord and \tkzFnPrintCoord

`\tkzFnShowCoord[⟨options⟩](⟨A⟩)`

Draws the segments from the point to the axes and writes its coordinates (American or French format). The options **`xlabel`** *and* **`ylabel`** *replace the computed values; with* **`xlabel=false`**, *nothing is written. The style is* **`tkzfn show coord`**.

`\tkzFnPrintCoord[⟨decimals⟩](⟨A⟩)`

Writes the coordinates of the point in the text.



$A(1.5, 2.12)$

```
\begin{tikzpicture}[scale=1.2]
  \tkzFnInit[xmin=0,xmax=3,ymin=0,ymax=4]
  \tkzFnAxes
  \tkzFnDraw[name=f,domain=0:2.5]{x^2/2+1}
  \tkzFnDefPoint[draw](f,1.5){A}
  \tkzFnShowCoord(A)
  \tkzFnDefPoint[draw](f,2){B}
  \tkzFnShowCoord[xlabel=$b$,
                  ylabel=$f(b)$](B)
\end{tikzpicture}

$A\tkzFnPrintCoord(A)$
```

6 Naming the curves

It is useful to give a name to the curves. Let us take as an example the function f defined by:

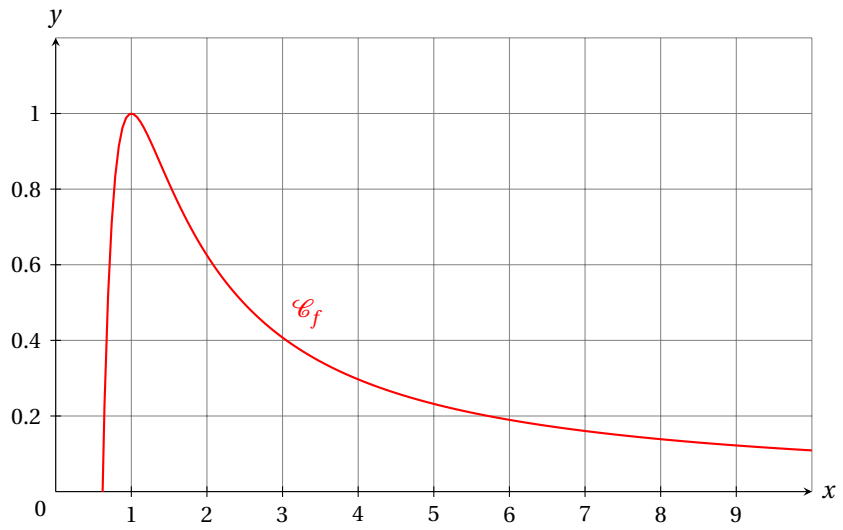
$$x > 0 \text{ and } f(x) = \frac{x^2 + x - 1}{x^3}$$

A title is easily written with the macro **\tkzFnText**. The coordinates are given in the units of the axes. To place a text along the curve, the simplest way is to define a point of the curve and to use its name.

```
1  \tkzFnDefPoint(f,3){P}
2  \tkzFnText[above right](P){${\mathcal{C}}_f$}
```

The first line defines the point P of the graph with $x = 3$. The TikZ options refine the result.

6.1 Adding a label



Graph of f

```
\begin{tikzpicture}
  \tkzFnInit[xmin=0,xmax=10,
             ymin=0,ymax=1.2,ystep=0.2]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[name=f,thick,color=red,domain=0.55:10]{(x*x+x-1)/x^3}
  \tkzFnText(3,-0.3){\textbf{Graph of} $\mathbf{f}$}
  \tkzFnDefPoint(f,3){P}
  \tkzFnText[above right,text=red](P){${\mathcal{C}}_f$}
\end{tikzpicture}
```

7 Tangent lines

The slope is computed by Lua in the units of the coordinate system, so a tangent line is correct even when **xstep** and **ystep** are different. Each tangent line is made of two half-lines: on the left (l) and on the right (r). Their lengths are given by **kl** and **kr**, in units of the x-axis: the right half goes from the point of tangency $(a, f(a))$ to the point $(a + k_r, f(a) + k_r f'_+(a))$. With $kl = 0$ or $kr = 0$, the corresponding half is not drawn.

If the function is not differentiable at a, the two halves have different slopes; they are computed separately.

7.1 Drawing a tangent line: \tkzFnTangent

`\tkzFnTangent[⟨options⟩](⟨name of the function,a⟩)`

arguments	example	explanation
function,a	(f,0)	tangent line at the point with $x = 0$
function	(f)	with the option **at**, see next section

options	default	definition
draw	false	the point of tangency is drawn
kl	1	length of the left half (units of the x-axis)
kr	1	length of the right half
clip	true	the line is clipped to the window
at		zeros or extrema: tangent lines at all these points

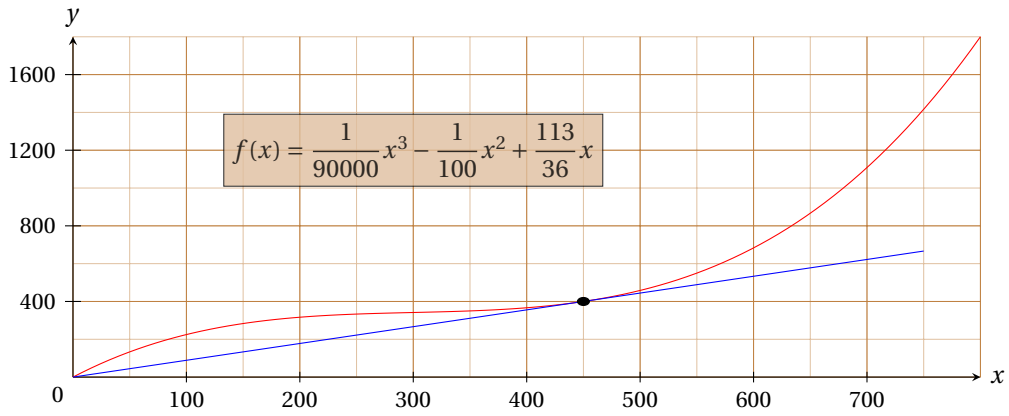
Other options are TikZ options; the style is **tkzfn tangent**, *empty by default. To get the arrows of* **tkz-fct**:

```
\tikzset{tkzfn tangent/.style={->,>=latex}}
```

☞ The function does not need to be drawn; it only has to be defined.

7.2 Tangent line when xstep and ystep are not 1

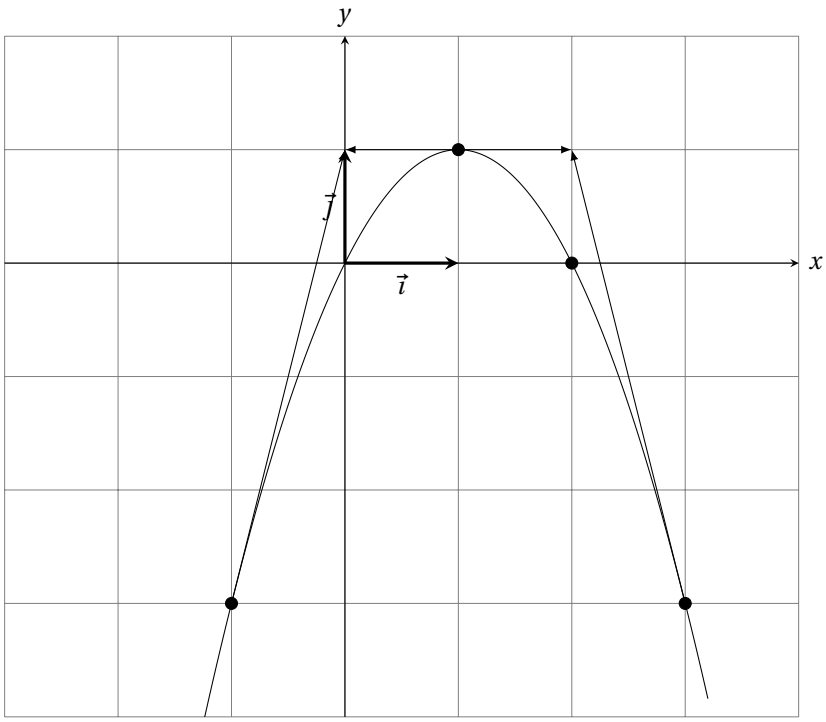
No computation is needed: just use the values of the tick marks.



```
\begin{tikzpicture}[xscale=1.5]
 \tkzFnInit[xmin=0,xmax=800,xstep=100,
            ymin=0,ymax=1800,ystep=400]
 \tkzFnGrid[color=brown,sub,subdx=50,subdy=200](0,0)(800,1800)
 \tkzFnAxes
 \tkzFnDraw[name=f,color=red,samples=100,domain=0:800]
    {(1/900000)*x^3-(1/100)*x^2+(113/36)*x}
 \tkzFnTangent[draw,color=blue,kr=300,kl=450](f,450)
 \tkzFnText[draw,color=black,fill=brown!50,opacity=0.8](300,1200)
 {$f(x)=\dfrac{1}{90000}x^3 -\dfrac{1}{{100}}x^2 +\dfrac{113}{36}x$}
\end{tikzpicture}
```

The slope at 450 is 0.8889; it is obtained with `\tkzFnPrint[4]{diff(f,450)}`.

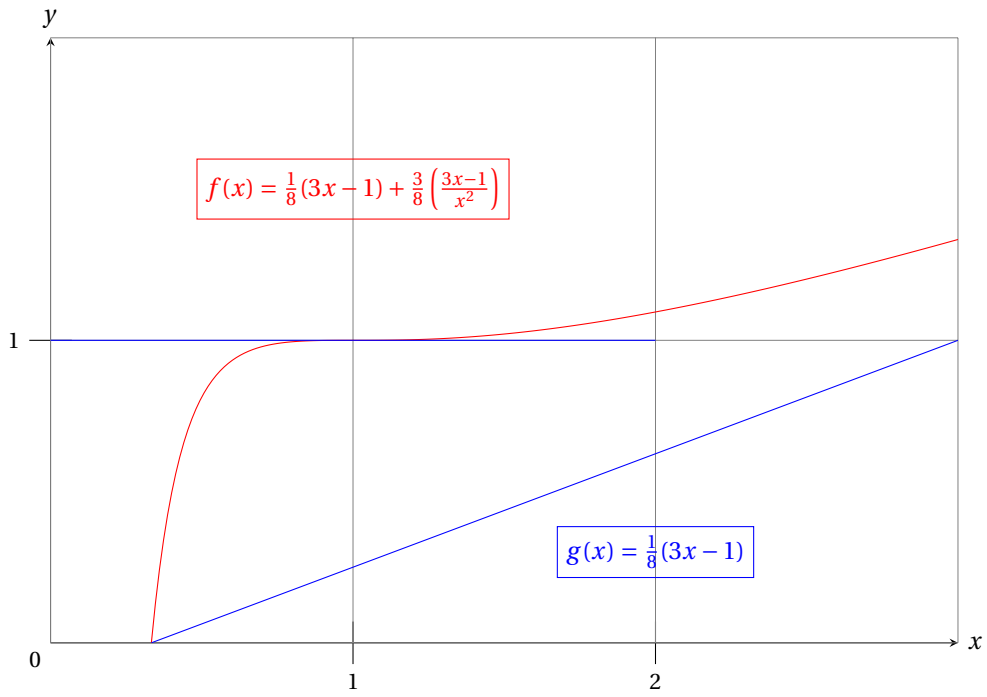
7.3 The options kl, kr and draw




```
\begin{tikzpicture}[scale=1.5]
  \tikzset{tkzfn tangent/.style={->,>=latex}}
  \tkzFnInit[xmin=-3,xmax=4,ymin=-4,ymax=2]
  \tkzFnGrid  \tkzFnAxes[ticks=false]
  \tkzFnDraw[name=f,domain=-2.15:3.2]{-x*x+2*x}
  \tkzFnDefPoint[draw](f,2){A}
  \tkzFnTangent[kl=0,draw](f,-1)
  \tkzFnTangent[draw](f,1)
  \tkzFnTangent[kr=0,draw](f,3)
  \tkzFnRep
\end{tikzpicture}
```

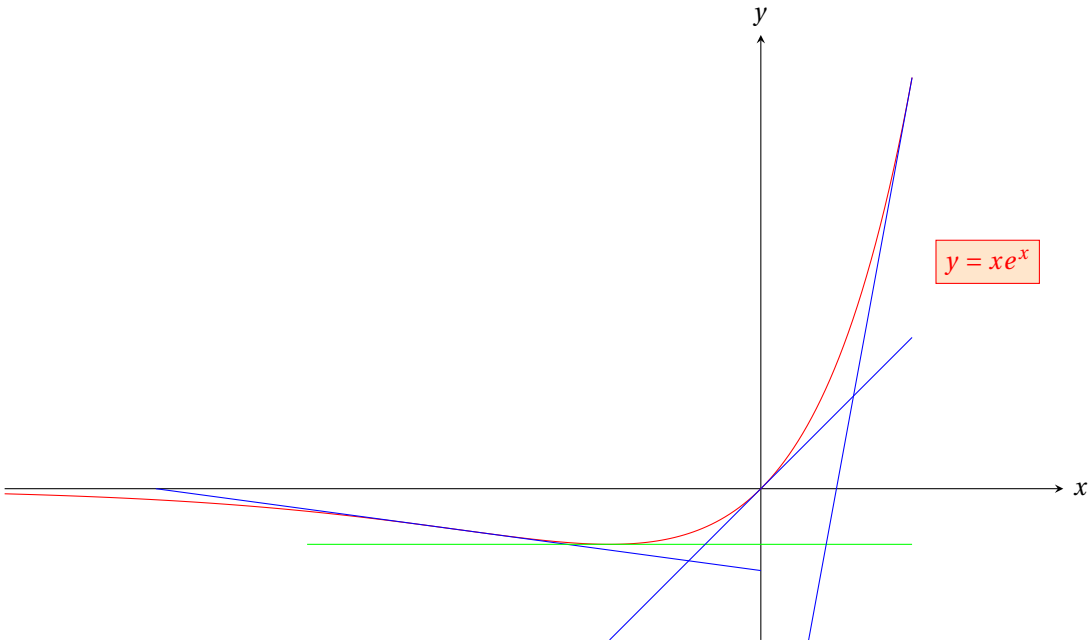
7.4 Tangent line with two functions

The option **with** of **tkz-fct** no longer exists: the function is given by its name.




```
\begin{tikzpicture}[scale=4]
 \tkzFnInit[xmin=0,xmax=3,ymin=0,ymax=2]
 \tkzFnAxes
 \tkzFnGrid(0,0)(3,2)
 \tkzFnDraw[name=f,color=red,domain=1/3:3]
           {0.125*(3*x-1)+0.375*(3*x-1)/(x*x)}
 \tkzFnDraw[name=g,color=blue,domain=1/3:3]
           {0.125*(3*x-1)}
 \tkzFnTangent[color=blue](f,1)
 \tkzFnText[draw,color=red](1,1.5)
   {$f(x)=\frac{1}{8}(3x-1)+\frac{3}{8}\left(\frac{3x-1}{x^2}\right)$}
 \tkzFnText[draw,color=blue](2,0.3)
   {$g(x)=\frac{1}{8}(3x-1)$}
\end{tikzpicture}
```

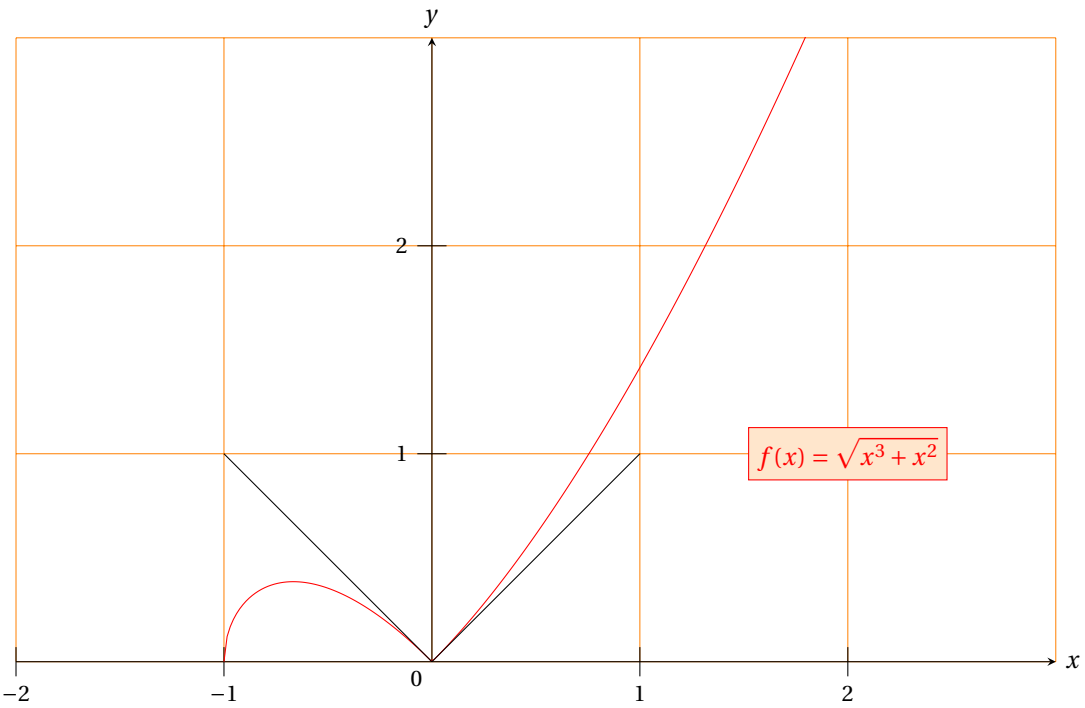
7.5 A few tangent lines




```
\begin{tikzpicture}[scale=2]
  \tkzFnInit[xmin=-5,xmax=2,ymin=-1,ymax=3]
  \tkzFnAxes[ticks=false]
  \tkzFnText[draw,color=red,fill=orange!20](1.5,1.5){$y = xe^x$}
  \tkzFnDraw[name=f,color=red,domain=-5:1]{x*exp(x)}
  \tkzFnTangent[color=blue,kr=2,kl=2](f,-2)
  \tkzFnTangent[color=green,kr=2,kl=2](f,-1)
  \tkzFnTangent[color=blue](f,0)
  \tkzFnTangent[color=blue,kr=0](f,1)
\end{tikzpicture}
```

7.6 One-sided tangent lines

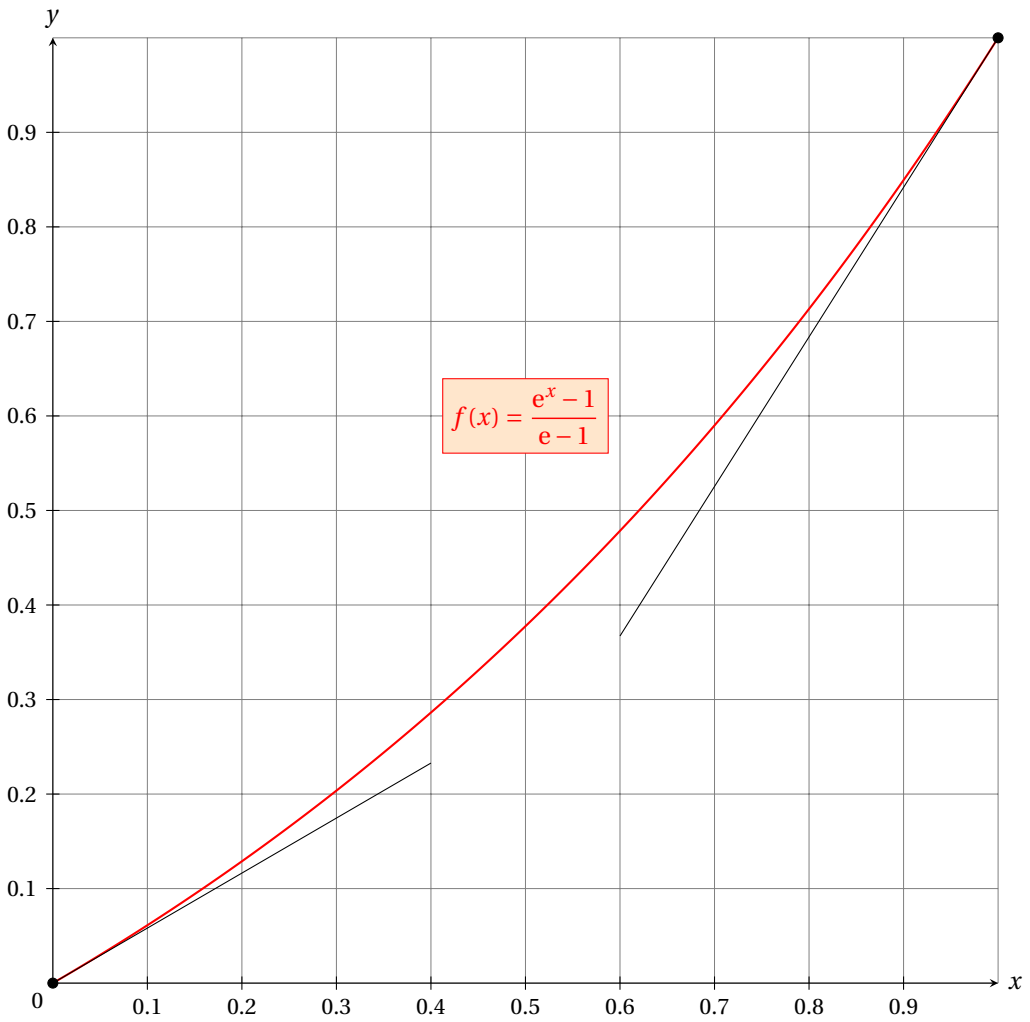
At 0, the function f defined by $f(x) = \sqrt{x^3 + x^2}$ is not differentiable: the two halves are obtained automatically.




```
\begin{tikzpicture}[scale=2.75]
    \tkzFnInit[xmin=-2,xmax=3,ymin=0,ymax=3]
    \tkzFnGrid[color=orange](-2,0)(3,3)
    \tkzFnAxes
    \tkzFnDraw[name=f,color=red,domain=-1:2]{sqrt(x^3+x^2)}
    \tkzFnTangent(f,0)
    \tkzFnText[draw,color=red,fill=orange!20](2,1){$f(x)=\sqrt{x^3+x^2}$}
\end{tikzpicture}
```

7.7 One-sided tangent lines: Lorenz curve

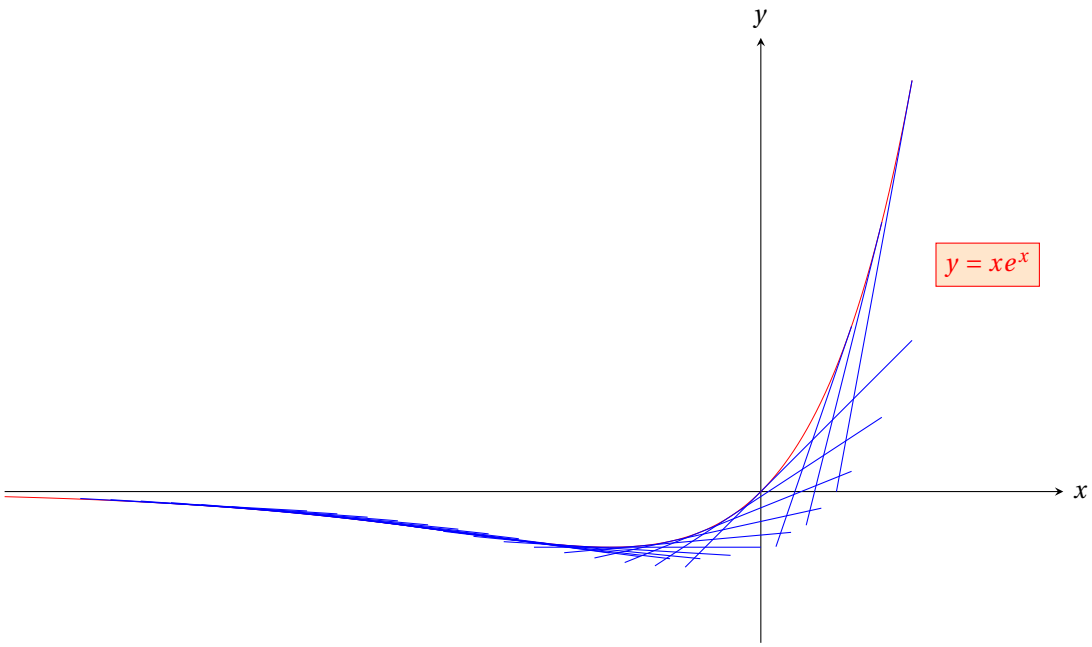
Here only the halves between 0 and 1 are wanted: give the value 0 to **kl** in one case and to **kr** in the other.




```
\begin{tikzpicture}[scale=1.25]
  \tkzFnInit[xmin=0,xmax=1,ymin=0,ymax=1,xstep=0.1,ystep=0.1]
  \tkzFnGrid(0,0)(1,1)
  \tkzFnAxes
  \tkzFnDraw[name=f,color=red,thick,domain=0:1]{(exp(x)-1)/(exp(1)-1)}
  \tkzFnTangent[draw,kl=0,  kr=0.4](f,0)
  \tkzFnTangent[draw,kl=0.4,kr=0  ](f,1)
  \tkzFnText[draw,color=red,fill=orange!20](0.5,0.6)
          {$f(x)=\dfrac{\text{e}^x-1}{\text{e}-1}$}
\end{tikzpicture}
```

7.8 A family of tangent lines

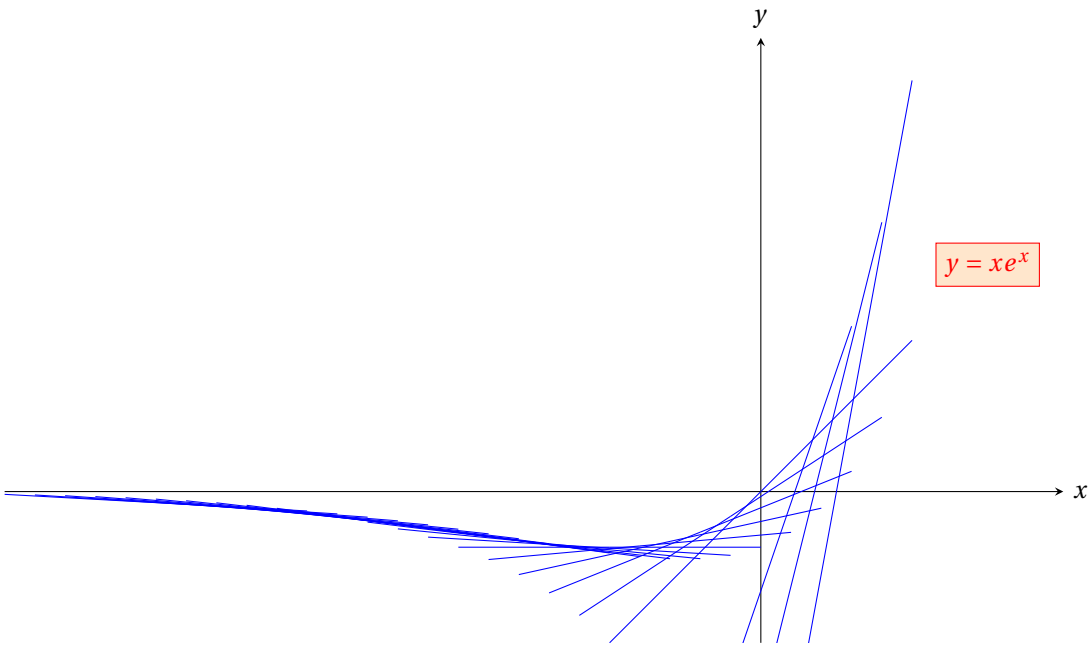
Tangent lines can be drawn in a `\foreach` loop.




```
\begin{tikzpicture}[scale=2]
  \tkzFnInit[xmin=-5,xmax=2,ymin=-1,ymax=3]
  \tkzFnAxes[ticks=false]
  \tkzFnText[draw,color=red,fill=orange!20](1.5,1.5){$y = xe^x$}
  \tkzFnDraw[name=f,line width=0.01pt,color=red,domain=-5:1]{x*exp(x)}
  \foreach \v in {-4,-3.8,...,0}{
    \tkzFnTangent[color=blue,line width=.4pt,kr=1,kl=0.5](f,\v)}
  \foreach \v in {0.6,0.8,1}{
    \tkzFnTangent[color=blue,line width=.4pt,kr=0,kl=0.5](f,\v)}
\end{tikzpicture}
```

7.9 Tangent lines without the curve

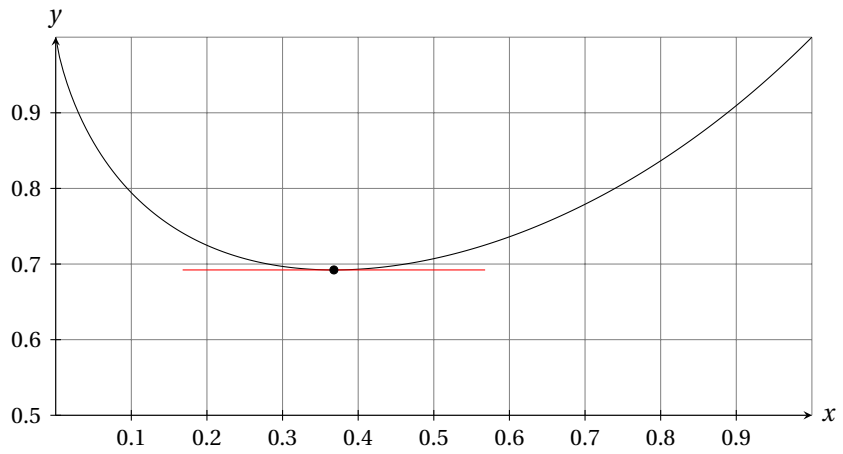
The function is defined with **\tkzFnDef** but not drawn.




```
\begin{tikzpicture}[scale=2]
  \tkzFnInit[xmin=-5,xmax=2,ymin=-1,ymax=3]
  \tkzFnAxes[ticks=false]
  \tkzFnText[draw,color=red,fill=orange!20](1.5,1.5){$y = xe^x$}
  \tkzFnDef{f}{x*exp(x)}
  \foreach \v in {-4,-3.8,...,0}{
    \tkzFnTangent[color=blue,line width=.4pt,kl=1](f,\v)}
  \foreach \v in {0.6,0.8,1}{
    \tkzFnTangent[color=blue,line width=.4pt,kr=0,kl=.75](f,\v)}
\end{tikzpicture}
```

7.10 Point of tangency given by an expression

The x-coordinate of the point of tangency can be a Lua expression; if it contains parentheses, put it between braces.




```
\begin{tikzpicture}[scale=1]
  \tkzFnInit[xmin=0,xmax=1,xstep=0.1,
             ymin=0.5,ymax=1,ystep=0.1]
  \tkzFnGrid \tkzFnAxes
  \tkzFnDraw[name=f,domain=0.00001:1]{x^x}
  \tkzFnTangent[draw,color=red,kr=0.2,kl=0.2](f,{1/exp(1)})
\end{tikzpicture}
```

8 Zeros and extrema

Lua finds all the zeros of a function on an interval: the interval is cut into **samples** steps, each change of sign is refined by bisection, and the zeros without change of sign (double root, the curve touches the axis) are found too. A change of sign without a zero, on both sides of an asymptote, is not taken for a zero. The extrema are the zeros of the derivative where it changes sign.

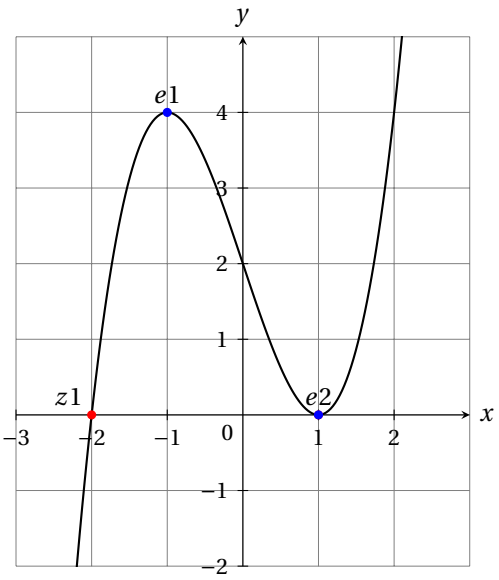
8.1 Defining the zeros: \tkzFnZeros

`\tkzFnZeros[⟨options⟩](⟨name of the function⟩)`

Defines the points z1, z2, ... of the curve, in increasing order of x. Their number is in **\tkzFnNbZeros**. *In a* `tikzpicture` *environment, the points are also TikZ nodes; outside, they are only stored, for* **\tkzFnPrintCoord**.

options	default	definition
at	zeros	zeros or extrema
domain	xmin:xmax	interval of the search
samples	400	number of steps of the scan
name	z	prefix of the names of the points
draw	false	the points are drawn
size	1.5pt	radius of the points

Other options are TikZ options, for drawing the points.




```
\begin{tikzpicture}
  \tkzFnInit[xmin=-3,xmax=3,ymin=-2,ymax=5]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[thick,name=g]{(x-1)^2*(x+2)}
  \tkzFnZeros[draw,red](g)
  \tkzFnLabelPoints[above left](z1)
  \tkzFnZeros[at=extrema,name=e,draw,blue](g)
  \tkzFnLabelPoints[above](e1,e2)
\end{tikzpicture}
```

$g(x) = (x-1)^2(x+2)$ has 2 zeros: $z_1 = (-2, 0)$ and the double root $z_2 = (1, 0)$, which is also the local minimum e_2. The local maximum is $e_1 = (-1, 4)$.

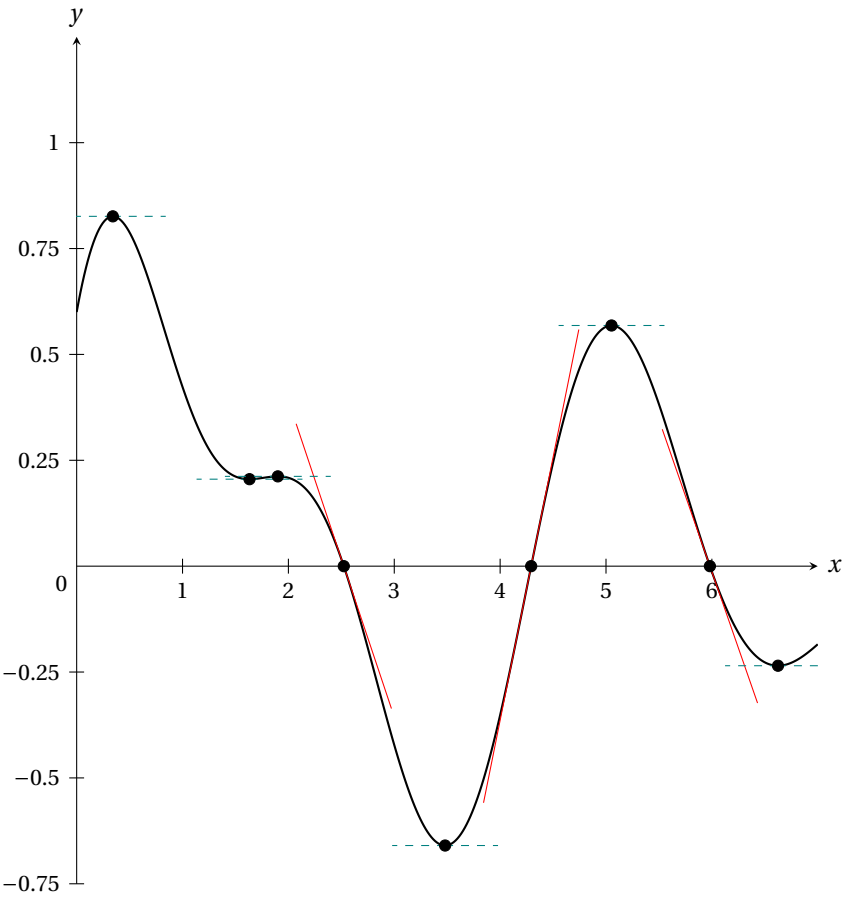
The text above is written with `\tkzFnZeros(g)\tkzFnNbZeros{} zeros: $z_1=\tkzFnPrintCoord(z1)$`, after the picture.

8.2 Tangent lines at the zeros and at the extrema

With the option **at**, **\tkzFnTangent** needs no abscissa: it draws the tangent line at each zero (`at=zeros`) or the horizontal tangent line at each extremum (`at=extrema`). The options **domain** and **samples** have the same

meaning as for \tkzFnZeros; the other options are those of \tkzFnTangent.

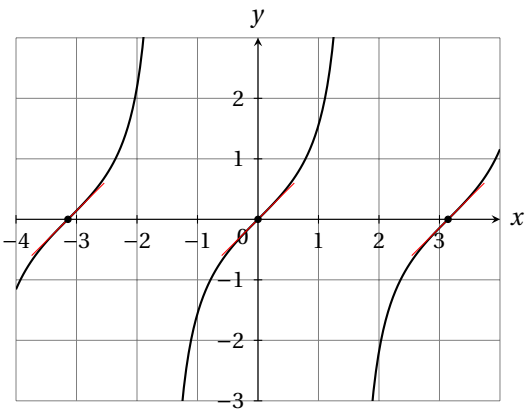
$f(x) = e^{-x/5}\big(\sin(1.5x) + 0.6\cos(2.5x)\big)$: none of these points can be computed by hand. The two extrema close to $x \approx 1.7$ are well separated.




```
\begin{tikzpicture}[scale=1.4]
  \tkzFnInit[xmin=0,xmax=7,ymin=-0.75,ymax=1.25,ystep=0.25]
  \tkzFnAxes
  \tkzFnDraw[name=f,samples=400,thick]
     {exp(-x/5)*(sin(1.5*x)+0.6*cos(2.5*x))}
  \tkzFnTangent[at=zeros,draw,red,kl=.45,kr=.45](f)
  \tkzFnTangent[at=extrema,draw,teal,dashed,kl=.5,kr=.5](f)
\end{tikzpicture}
```

8.3 Discontinuous function

tan changes sign on both sides of each asymptote without vanishing: these changes of sign are discarded.




```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-4,xmax=4,ymin=-3,ymax=3]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[thick,name=t]{tan(x)}
  \tkzFnTangent[at=zeros,draw,red,
                kl=.6,kr=.6](t)
\end{tikzpicture}
```

8.4 Limits

– Two zeros closer than one step of the scan may be missed: increase **samples** or reduce **domain**.

– The bounds of the interval are never extrema: only the points where the derivative vanishes and changes sign are kept. For $x \mapsto x^3$, there is no extremum at 0.

9 Areas

The aim is to show the region between the graph of a function, the x-axis and the lines $x = a$ and $x = b$, or the region between two graphs.

9.1 Drawing an area: \tkzFnArea

\tkzFnArea[⟨options⟩](⟨f⟩) or [⟨options⟩](⟨f,g⟩)

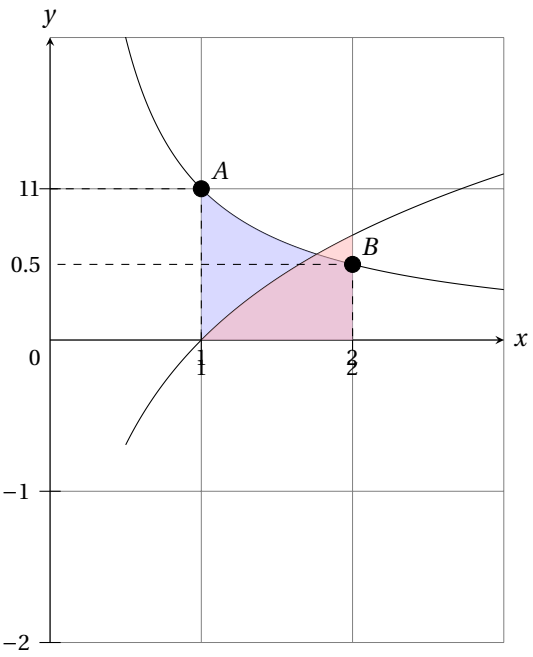
With one argument, the region lies between the graph of f and the x-axis. With two arguments, it lies between the graphs of f and g; g can be a number: (f,12).

options	default	definition
domain	xmin:xmax	interval $[a,b]$
where	both	above: only where $f \geq g$; below: where $f \leq g$
samples	200	number of points
clip	true	the region is clipped to the window

*Other options are TikZ options (**fill**, **pattern**, **opacity**...); the style is **tkzfn area**.*

☞ The functions do not need to be drawn; the drawing order (before or after the area) decides what is in the foreground.

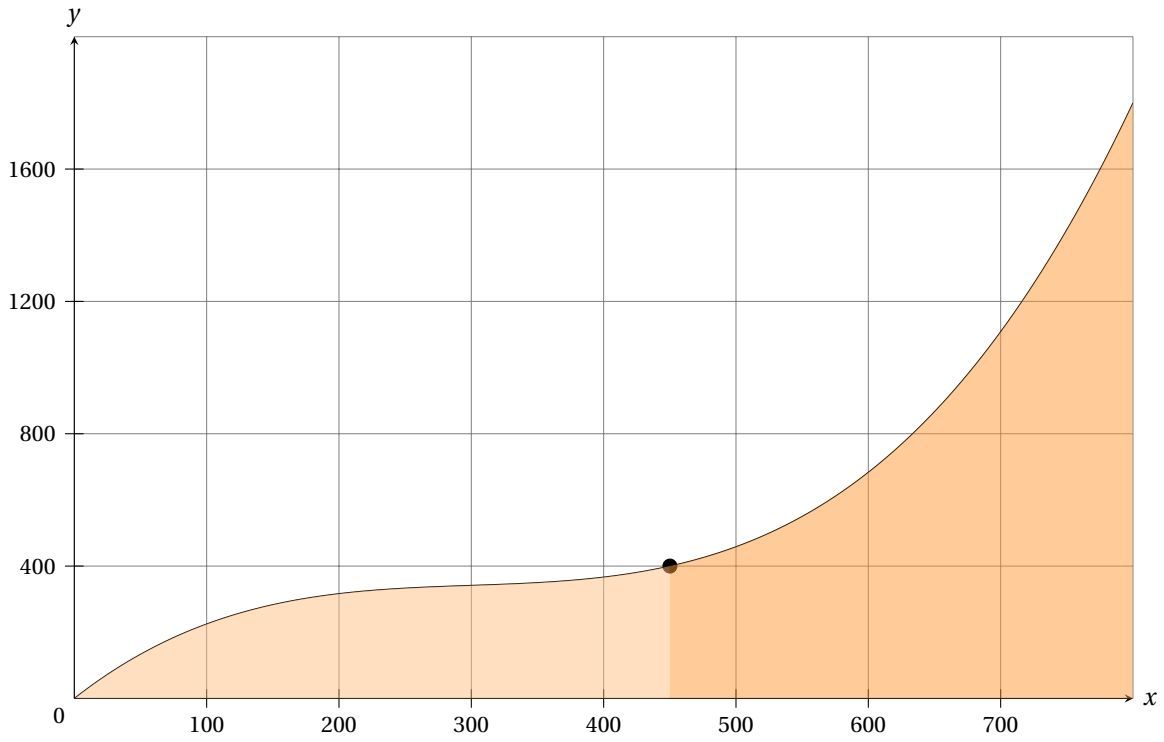
9.2 Birth of the natural logarithm



```
\begin{tikzpicture}[scale=2]
 \tkzFnInit[xmin=0,xmax=3,xstep=1,
            ymin=-2,ymax=2,ystep=1]
 \tkzFnGrid
 \tkzFnAxes
 \tkzFnDraw[name=f,domain=0.4:3]{1/x}
 \tkzFnDefPoint(f,1){A}
 \tkzFnDefPoint(f,2){B}
 \tkzFnLabelPoints[above right](A,B)
 \tkzFnArea[fill=blue!30,domain=1:2](f)
 \tkzFnDraw[name=g,domain=0.5:3]{ln(x)}
 \tkzFnArea[fill=red!30,domain=1:2](g)
 \tkzFnShowCoord(A)
 \tkzFnShowCoord(B)
 \tkzFnDrawPoints(A,B)
\end{tikzpicture}
```

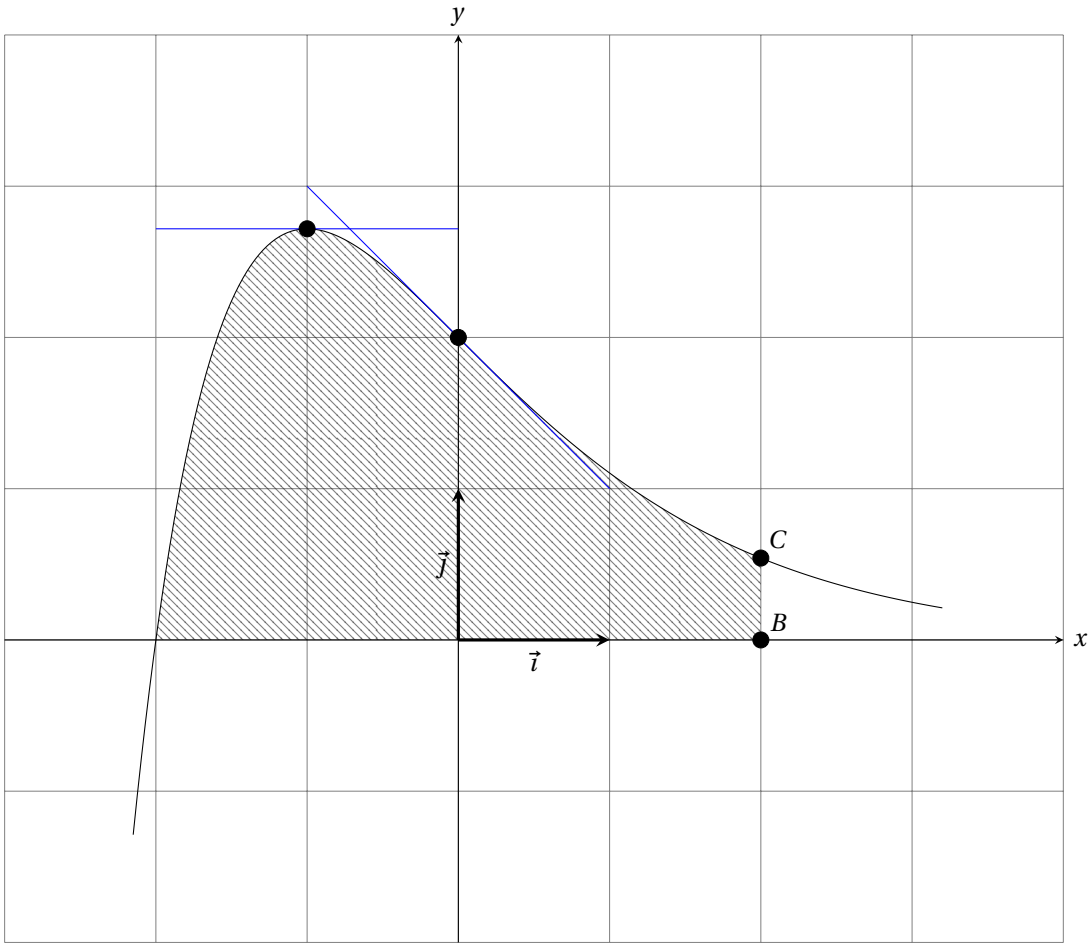
The two areas are $\ln 2 \approx 0.6931$ and $2\ln 2 - 1 \approx 0.3863$: `\tkzFnPrint[4]{integral(f,1,2)}`.

9.3 A simple area



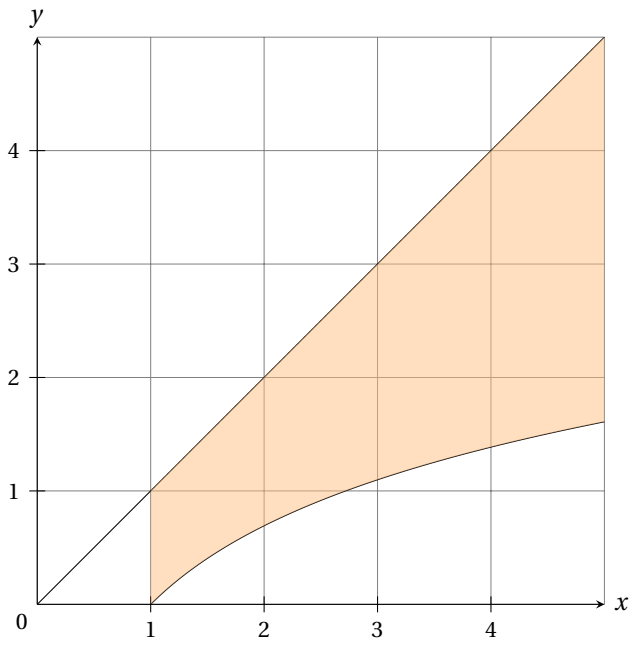
```
\begin{tikzpicture}[scale=1.75]
 \tkzFnInit[xmin=0,xmax=800,xstep=100,
            ymin=0,ymax=2000,ystep=400]
 \tkzFnGrid
 \tkzFnAxes
 \tkzFnDraw[name=f,domain=0:800]
     {(1/90000)*x^3-(1/100)*x^2+(113/36)*x}
 \tkzFnDefPointXY[draw](450,400){a}
 \tkzFnArea[fill=orange!50,domain=0:450](f)
 \tkzFnArea[fill=orange!80,domain=450:800](f)
\end{tikzpicture}
```

9.4 Area and hatching



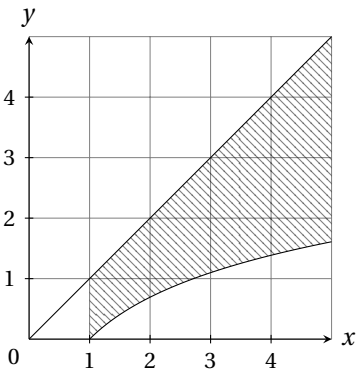

```
\begin{tikzpicture}[scale=2]
  \tkzFnInit[xmin=-3,xmax=4,ymin=-2,ymax=4]
  \tkzFnGrid(-3,-2)(4,4)
  \tkzFnAxes[ticks=false]
  \tkzFnDraw[name=f,domain=-2.15:3.2]{(2+x)*exp(-x)}
  \tkzFnArea[pattern=north west lines,domain=-2:2](f)
  \tkzFnTangent[draw,color=blue](f,0)
  \tkzFnTangent[draw,color=blue](f,-1)
  \tkzFnDefPoint(f,2){C}
  \tkzFnDefPointXY(2,0){B}
  \tkzFnDrawPoints(B,C)  \tkzFnLabelPoints[above right](B,C)
  \tkzFnRep
\end{tikzpicture}
```

9.5 Area between two curves, in color




```
\begin{tikzpicture}[scale=1.5]
  \tkzFnInit[xmin=0,xmax=5,ymin=0,ymax=5]
  \tkzFnGrid  \tkzFnAxes
  \tkzFnDraw[name=f,domain=0:5]{x}
  \tkzFnDraw[name=g,domain=1:5]{ln(x)}
  \tkzFnArea[fill=orange!50,domain=1:5](f,g)
\end{tikzpicture}
```

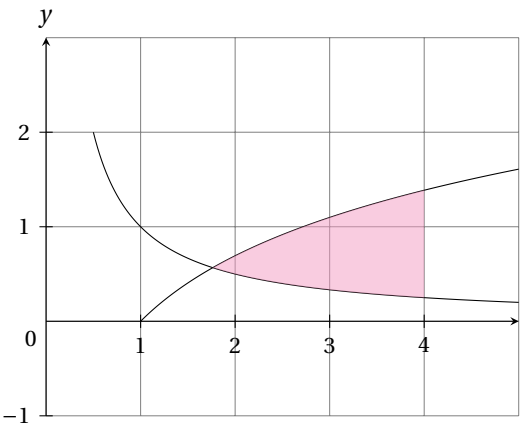
9.6 Area between two curves, with hatching




```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=0,xmax=5,ymin=0,ymax=5]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[name=f,domain=0:5]{x}
  \tkzFnDraw[name=g,domain=1:5]{ln(x)}
  \tkzFnArea[pattern=north west lines,domain=1:5](f,g)
\end{tikzpicture}
```

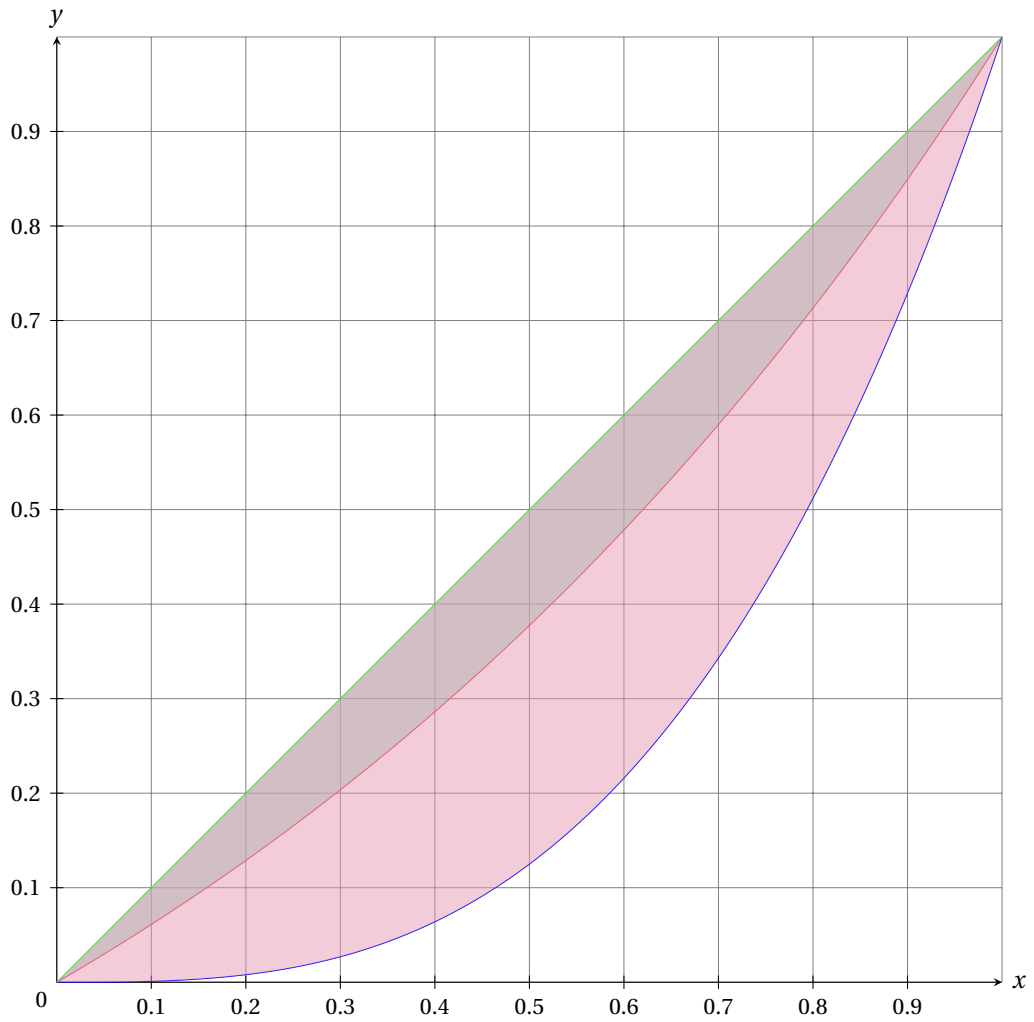
9.7 Area between two curves: option where

The curves intersect. With **where=above**, only the part where the graph of g is above the graph of f is shown.



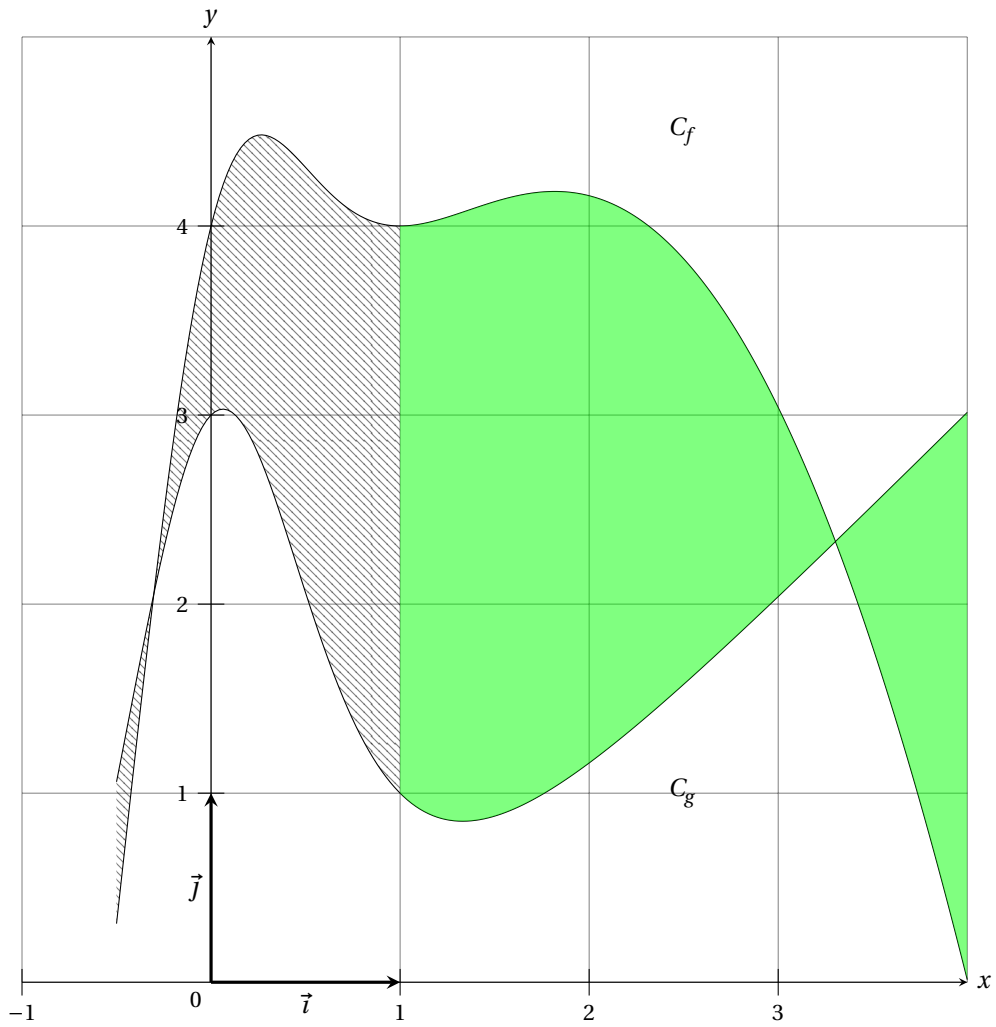

```
\begin{tikzpicture}[scale=1.25]
 \tkzFnInit[xmin=0,xmax=5,
            ymin=-1,ymax=3]
 \tkzFnGrid
 \tkzFnAxes
 \tkzFnDraw[name=f,domain=0.5:5]{1/x}
 \tkzFnDraw[name=g,domain=1:5]{ln(x)}
 \tkzFnArea[where=above,
            fill=magenta!50,
            domain=1:4](g,f)
\end{tikzpicture}
```

9.8 Area between two curves: Lorenz curves




```
\begin{tikzpicture}[scale=1.25]
  \tkzFnInit[xmin=0,xmax=1,ymin=0,ymax=1,xstep=0.1,ystep=0.1]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[name=f,color=red,domain=0:1]{(exp(x)-1)/(exp(1)-1)}
  \tkzFnDraw[name=g,color=blue,domain=0:1]{x^3}
  \tkzFnDraw[name=h,color=green,domain=0:1]{x}
  \tkzFnArea[fill=purple!40,domain=0:1](h,g)
  \tkzFnArea[fill=gray!60,domain=0:1](h,f)
\end{tikzpicture}
```

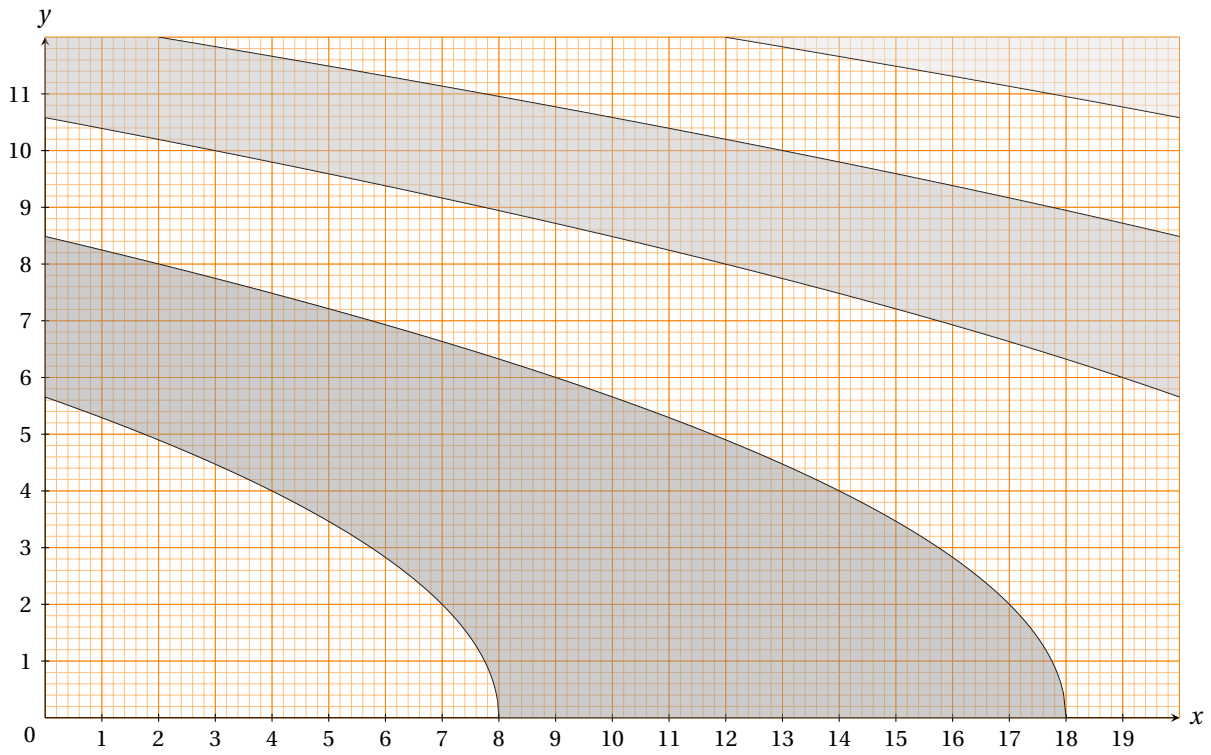
9.9 Mixing styles




```
\begin{tikzpicture}[scale=2.5]
   \tkzFnInit[xmin=-1,xmax=4,ymin=0,ymax=5]
   \tkzFnGrid
   \tkzFnAxes
   \tkzFnDraw[name=f,domain=-.5:4]{4*x-x^2+4/(x^2+1)^2}
   \tkzFnDraw[name=g,domain=-.5:4]{x-1+4/(x^2+1)^2}
   \tkzFnArea[fill=green,domain=1:4](f,g)
   \tkzFnArea[pattern=north west lines,domain=-.5:1](f,g)
   \tkzFnRep
   \tkzFnText(2.5,4.5){$C_f$}
   \tkzFnText(2.5,1){$C_g$}
\end{tikzpicture}
```

9.10 Level curves

With **tkz-fct**, you had to define macros for constant functions; now just write the number as the second argument (lines 10 to 14).




```
1  \begin{tikzpicture}[scale=.75]
2   \tkzFnInit[xmin=0,xmax=20,ymin=0,ymax=12]
3   \tkzFnGrid[color=orange,sub](0,0)(20,12)
4   \tkzFnAxes
5   \tkzFnDraw[name=a,samples=400,domain=0:8]{(32-4*x)^0.5}
6   \tkzFnDraw[name=b,samples=400,domain=0:18]{(72-4*x)^0.5}
7   \tkzFnDraw[name=c,samples=400,domain=0:20]{(112-4*x)^0.5}
8   \tkzFnDraw[name=d,samples=400,domain=2:20]{(152-4*x)^0.5}
9   \tkzFnDraw[name=e,samples=400,domain=12:20]{(192-4*x)^0.5}
10  \tkzFnArea[fill=gray!80,domain=0:8](b,a)
11  \tkzFnArea[fill=gray!80,domain=8:18](b,0)
12  \tkzFnArea[fill=gray!50,domain=2:20](d,c)
13  \tkzFnArea[fill=gray!50,domain=0:2](12,c)
14  \tkzFnArea[fill=gray!20,domain=12:20](12,e)
15 \end{tikzpicture}
```

10 Riemann sums

`\tkzFnRiemann[⟨options⟩](⟨name of the function⟩)`

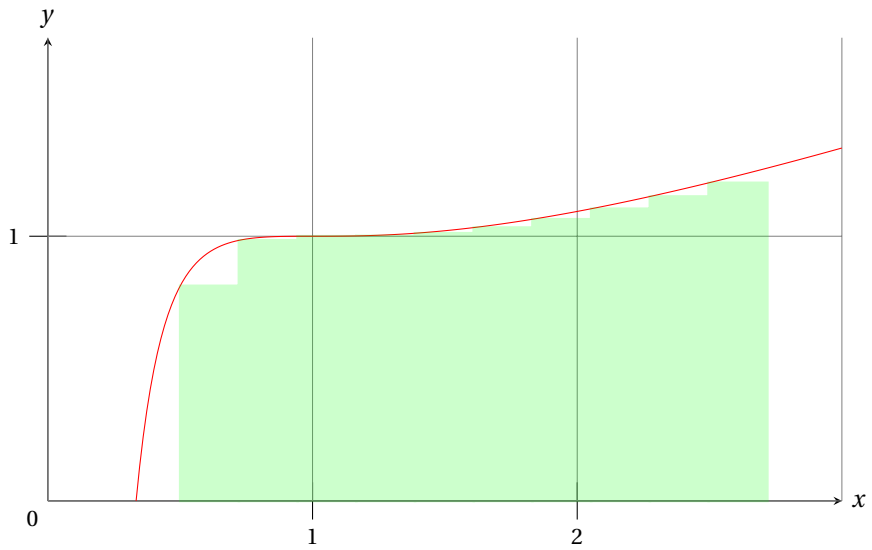
Draws the rectangles of a Riemann sum. The four macros of **tkz-fct** *are merged into one; the method is chosen with an option.*

options	default	definition
interval	xmin:xmax	interval where the method is applied
number	10	number of subintervals
method	left	height of the rectangles: see below
clip	false	the rectangles are clipped to the window

method	height of the rectangle on $[x_i, x_{i+1}]$
left	$f(x_i)$
right	$f(x_{i+1})$
mid	$f\left(\frac{x_i+x_{i+1}}{2}\right)$
inf	$\min\left(f(x_i), f(x_{i+1})\right)$
sup	$\max\left(f(x_i), f(x_{i+1})\right)$

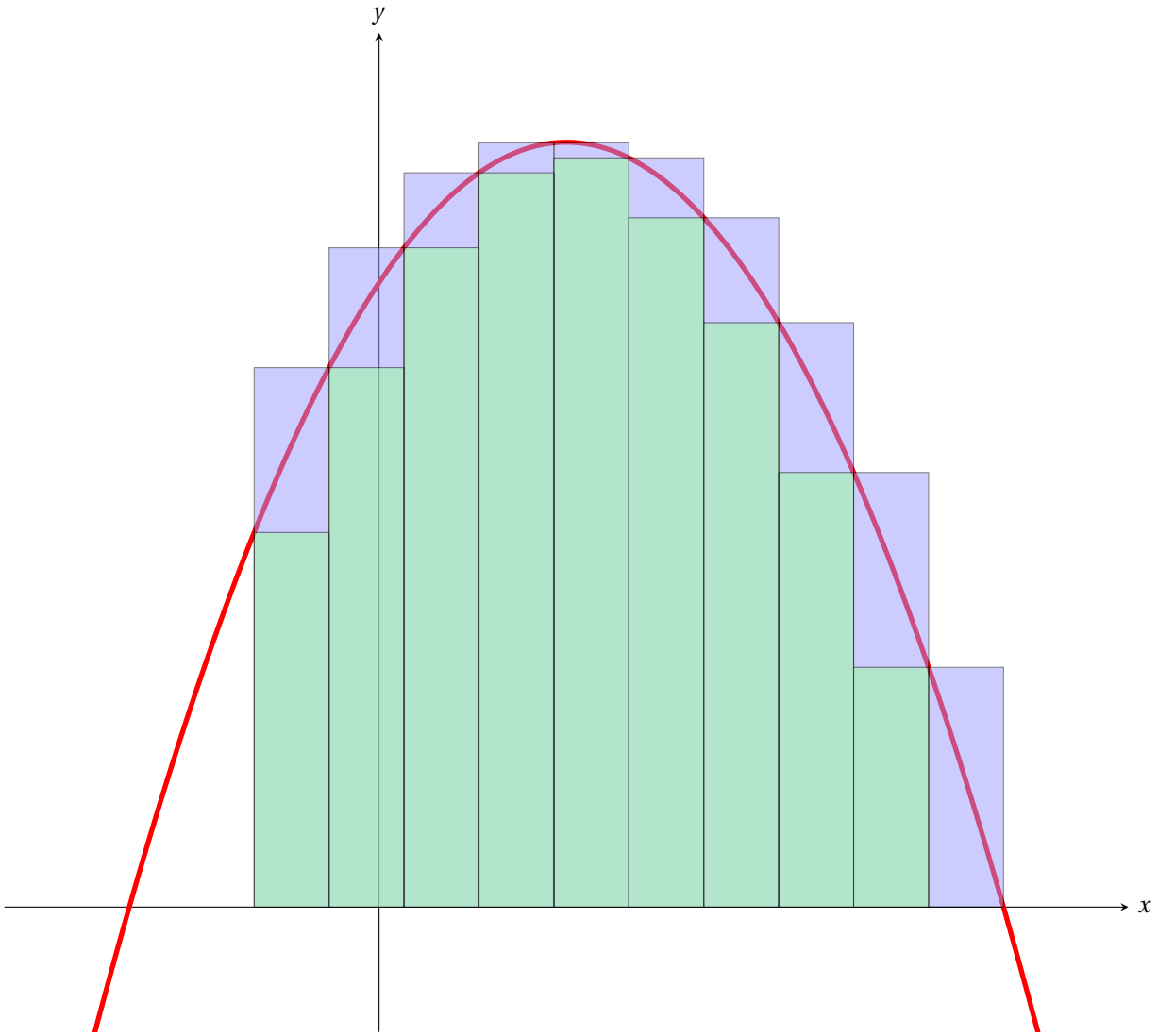
Other options are TikZ options; the style is **tkzfn riemann**. *The value of the sum is given by* `riemann(f,a,b,n,"left")`.

10.1 A Riemann sum



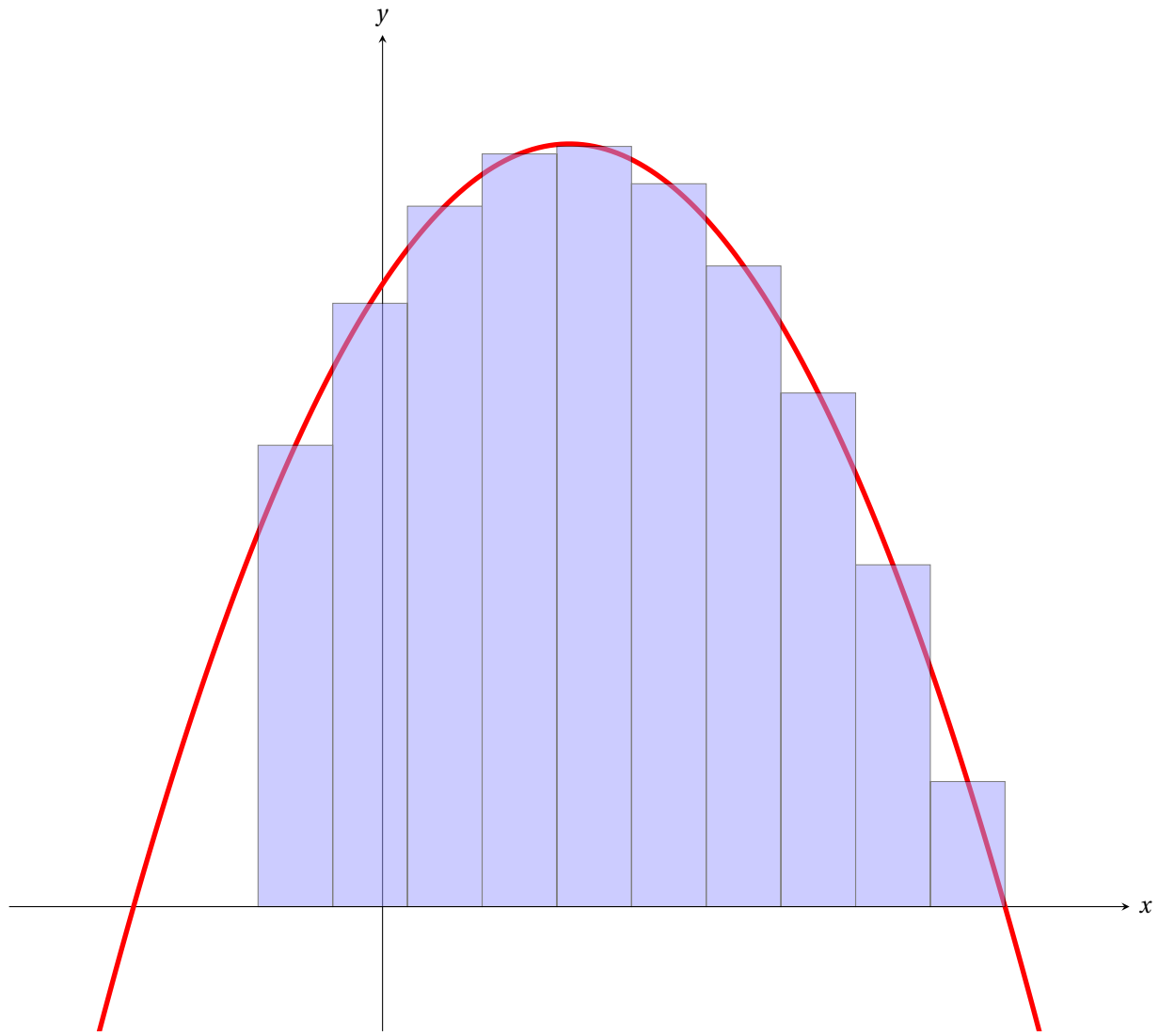

```
\begin{tikzpicture}[scale=3.5]
\tkzFnInit[xmin=0,xmax=3,ymin=0,ymax=1.75]
\tkzFnAxes
\tkzFnGrid(0,0)(3,1.75)
\tkzFnDraw[name=f,color=red,domain=1/3:3]{0.125*(3*x-1)+0.375*(3*x-1)/(x*x)}
\tkzFnRiemann[fill=green!40,opacity=.2,color=green,
              line width=1pt,interval=1/2:e,number=10](f)
\end{tikzpicture}
```

10.2 Methods inf and sup




```
\begin{tikzpicture}[scale=1.75]
  \tkzFnInit[xmin=-3,xmax=6,ymin=-2,ymax=14,ystep=2]
  \tkzFnAxes[ticks=false]
  \tkzFnDraw[name=f,line width=2pt,color=red,domain=-3:6]{(-x-2)*(x-5)}
  \tkzFnRiemann[method=sup,fill=blue!40,opacity=.5,interval=-1:5](f)
  \tkzFnRiemann[method=inf,fill=green!40,opacity=.5,interval=-1:5](f)
\end{tikzpicture}
```

10.3 Method mid




```
\begin{tikzpicture}[scale=1.75]
  \tkzFnInit[xmin=-3,xmax=6,ymin=-2,ymax=14,ystep=2]
  \tkzFnAxes[ticks=false]
  \tkzFnDraw[name=f,line width=2pt,color=red,domain=-3:6]{(-x-2)*(x-5)}
  \tkzFnRiemann[method=mid,fill=blue!40,opacity=.5,interval=-1:5](f)
\end{tikzpicture}
```

10.4 Values of the sums

For $f(x) = (-x-2)(x-5)$ on $[-1,5]$ with 10 rectangles:

inf	48.1
mid	54.18
sup	59.18
integral	54

```
\tkzFnDef{f}{(-x-2)*(x-5)}
\begin{tabular}{ll}
inf      & \tkzFnPrint{riemann(f,-1,5,10,"inf")}\\
mid      & \tkzFnPrint{riemann(f,-1,5,10,"mid")}\\
sup      & \tkzFnPrint{riemann(f,-1,5,10,"sup")}\\
integral & \tkzFnPrint{integral(f,-1,5)}
\end{tabular}
```

11 Special lines

11.1 Vertical lines: \tkzFnVLines

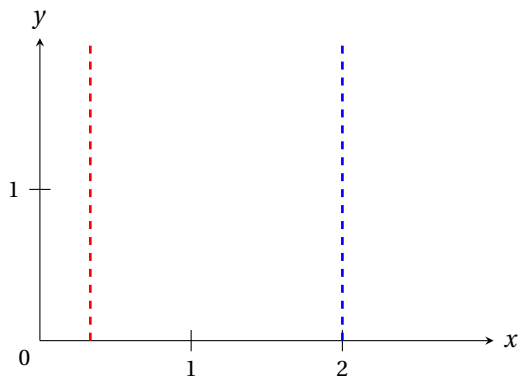
\tkzFnVLines[⟨TikZ options⟩]{⟨list⟩}

Draws the lines $x = a$ for the values a of the list. The list is written as with **\foreach** *(1,2,...,4) and its items are Lua expressions (1/3, 2*pi). \tkzFnVLine is a synonym. The style is* **tkzfn line**.

arguments	example	definition
list	{1,4}	draws the lines $x = 1$ and $x = 4$

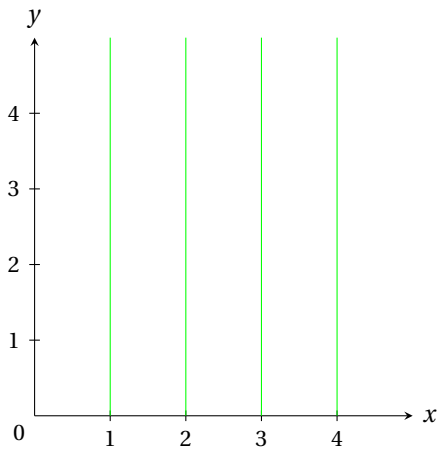
11.2 A vertical line

With **tkz-fct**, 1./3 was a problem; with Lua, 1/3 is correct.



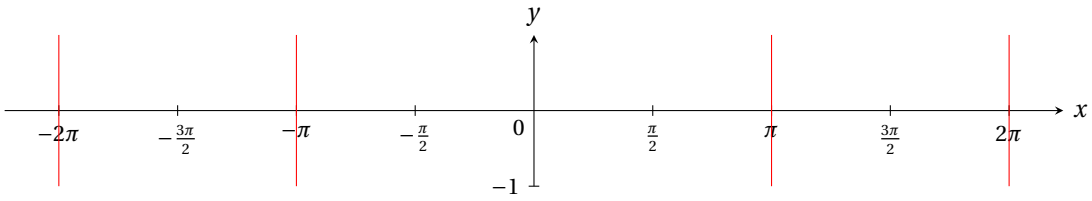

```
\begin{tikzpicture}[scale=2]
   \tkzFnInit[xmin=0,xmax=3,ymin=0,ymax=2]
   \tkzFnAxes
   \tkzFnVLine[color=blue,dashed,
               line width=1pt]{2}
   \tkzFnVLine[color=red,dashed,
               line width=1pt]{1/3}
\end{tikzpicture}
```

11.3 Vertical lines




```
\begin{tikzpicture}
 \tkzFnInit[xmin=0,xmax=5,ymin=0,ymax=5]
 \tkzFnAxes
 \tkzFnVLines[color=green]{1,2,...,4}
\end{tikzpicture}
```

11.4 Vertical lines at multiples of pi

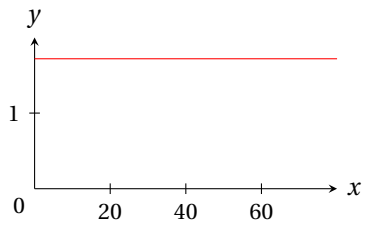



```
\begin{tikzpicture}
  \tkzFnInit[xmin=-7,xmax=7,ymin=-1,ymax=1]
  \tkzFnAxisY
  \tkzFnAxisX[trig=2]
  \tkzFnVLines[color=red]{-2*pi,-pi,pi,2*pi}
\end{tikzpicture}
```

11.5 Horizontal lines: \tkzFnHLines

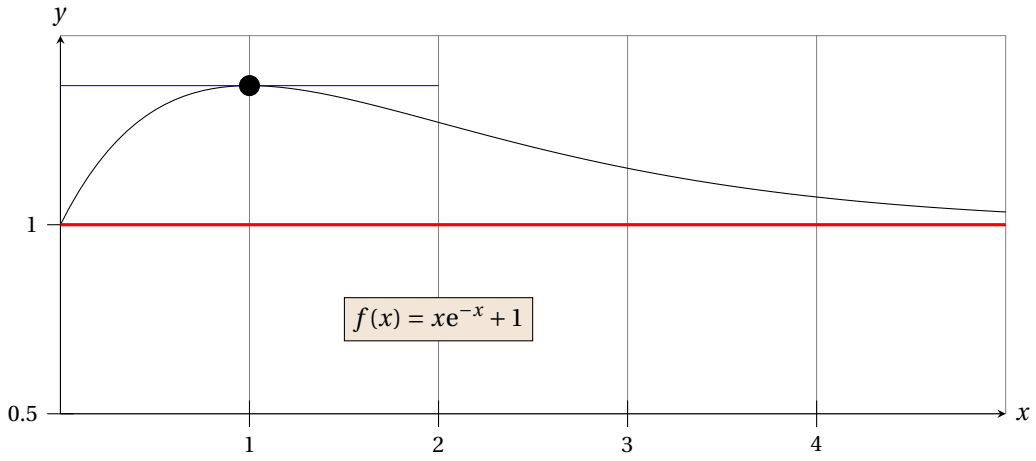
\tkzFnHLines[⟨TikZ options⟩]{⟨list⟩}

Draws the lines $y = b$. ***\tkzFnHLine*** *is a synonym.*



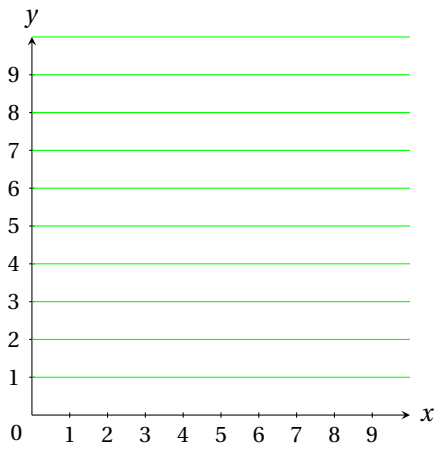

```
\begin{tikzpicture}
  \tkzFnInit[xmin=0,xmax=80,xstep=20,
             ymin=0,ymax=2]
  \tkzFnAxes
  \tkzFnHLine[color=red]{e-1}
\end{tikzpicture}
```

11.6 Horizontal asymptote



```
\begin{tikzpicture}[scale=2.5]
  \tkzFnInit[xmin=0,xmax=5,ymin=0.5,ymax=1.5,ystep=0.5]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[name=f,domain=0:10]{x*exp(-x)+1}
  \tkzFnHLine[color=red,line width=1.2pt]{1}
  \tkzFnTangent[draw,color=blue](f,1)
  \tkzFnText[draw,fill=brown!20](2,0.75){$f(x)=x \text{e}^{-x}+1$}
\end{tikzpicture}
```

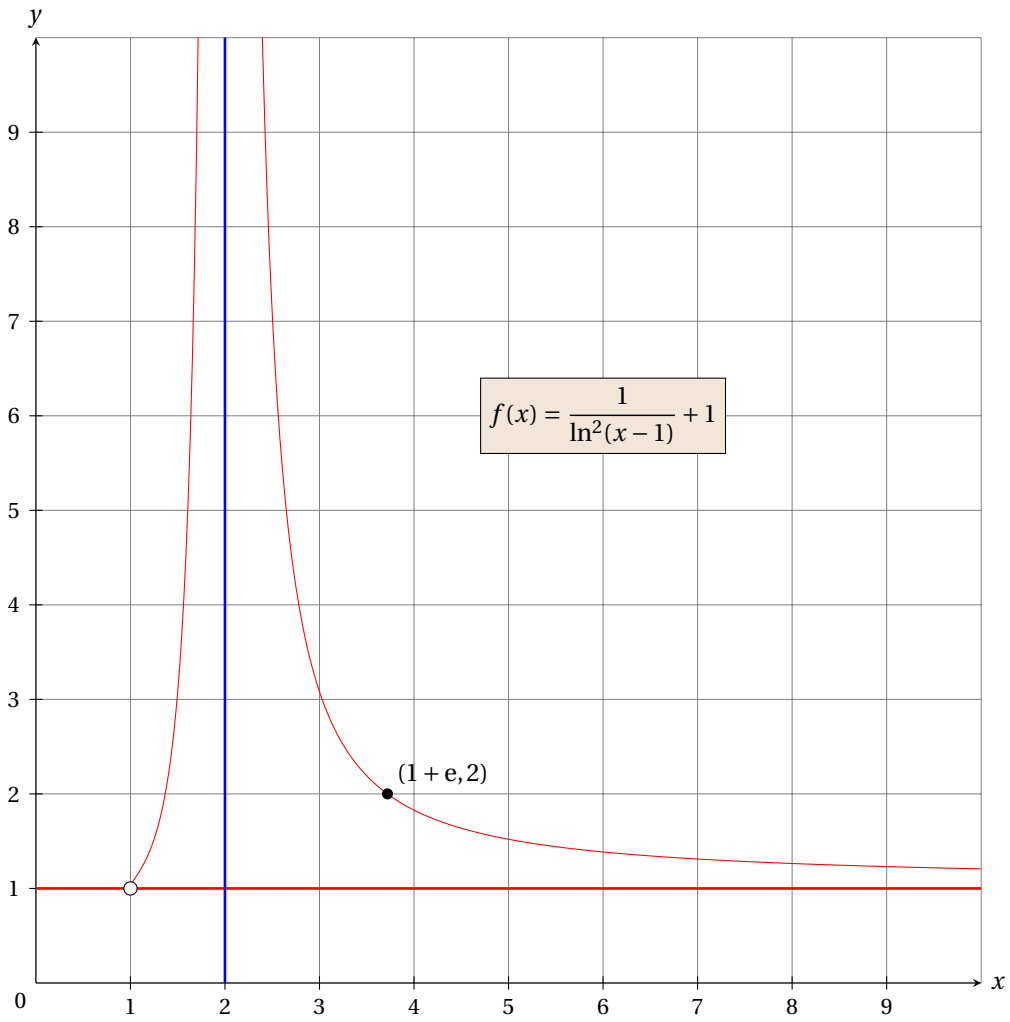
11.7 Horizontal lines




```
\begin{tikzpicture}[scale=.5]
 \tkzFnInit[xmin=0,xmax=10,ymin=0,ymax=10]
 \tkzFnAxes
 \tkzFnHLines[color=green]{1,2,...,10}
\end{tikzpicture}
```

11.8 Horizontal and vertical asymptotes

The curve is automatically interrupted at $x = 2$: there is no need to draw it in two parts.




```
\begin{tikzpicture}[scale=1.25]
 \tkzFnInit[xmin=0,xmax=10,ymin=0,ymax=10]
 \tkzFnGrid
 \tkzFnAxes
 \tkzFnDraw[name=f,color=red,samples=400,domain=1.001:10]{1+1/ln(x-1)^2}
 \tkzFnHLine[line width=1pt,color=red]{1}
 \tkzFnVLine[line width=1pt,color=blue]{2}
 \tkzFnDefPointXY(1,1){A}
 \tkzFnDrawPoints[fill=white,draw=Maroon,size=2pt](A)
 \tkzFnDefPoint[draw](f,{1+e}){B}
 \tkzFnLabelPoint[above right](B){$(1+\text{e}, 2)$}
 \tkzFnText[draw,color=black,fill=brown!20](6,6)
          {$f(x)=\dfrac{1}{\ln^2 (x-1)}+1$}
\end{tikzpicture}
```

12 Parametric curves

\tkzFnParam[⟨options⟩]{⟨$x(t)$⟩}{⟨$y(t)$⟩}

$x(t)$ and $y(t)$ are Lua expressions of the variable `t`.

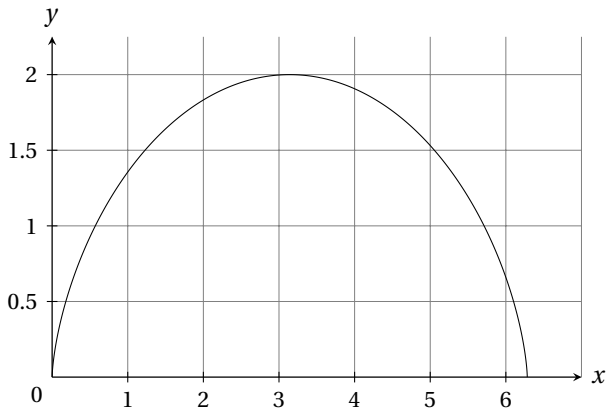
arguments	example	explanation
`x(t),y(t)`	`{t^3}{t^2}`	$x(t)=t^3,\ y(t)=t^2$

options	default	definition
`domain`	`0:2*pi`	interval of the values of t
`samples`	`200`	number of points
`clip`	`false`	the curve is clipped to the window

Other options are TikZ options.

12.1 Parametric curve, example 1

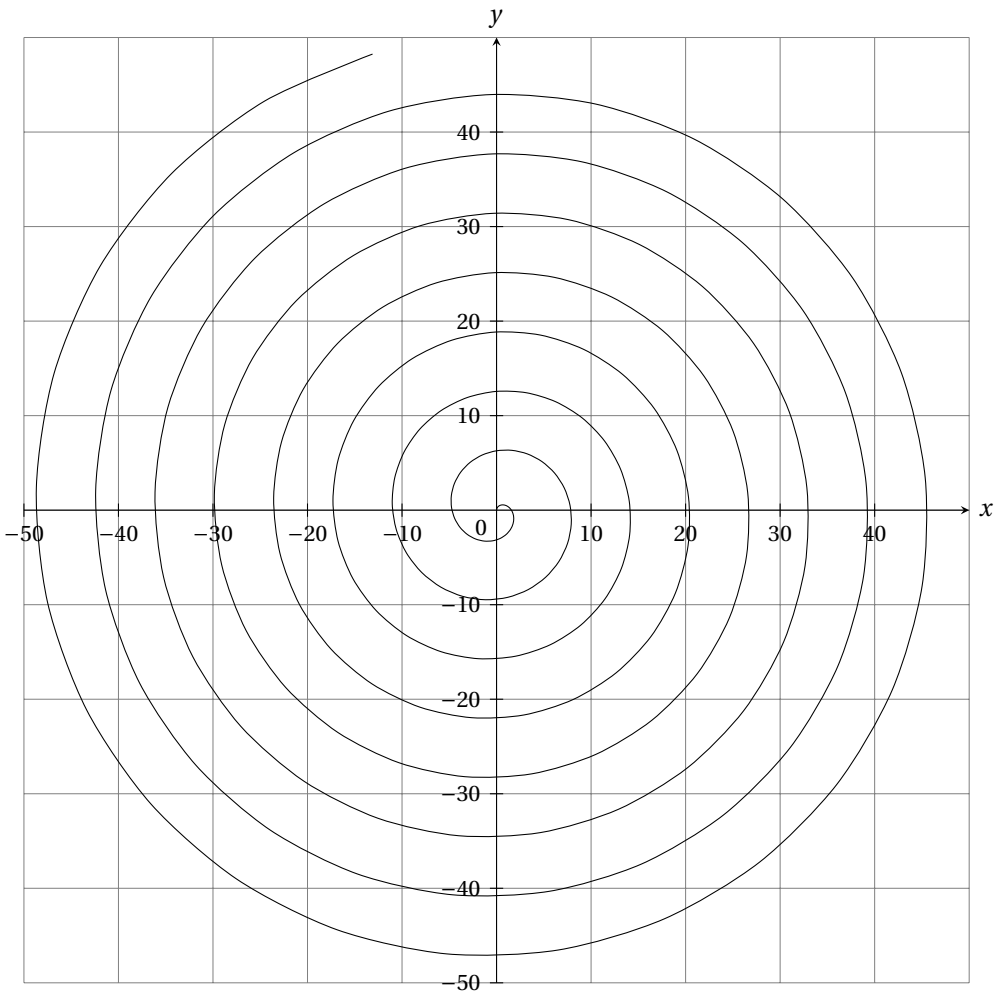
$$x(t) = t - \sin(t)$$
$$y(t) = 1 - \cos(t)$$




```
\begin{tikzpicture}
  \tkzFnInit[xmin=0,xmax=7,ymin=0,ymax=2.25,ystep=.5]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnParam[samples=400,domain=0:2*pi]{t-sin(t)}{1-cos(t)}
\end{tikzpicture}
```

12.2 Parametric curve, example 2

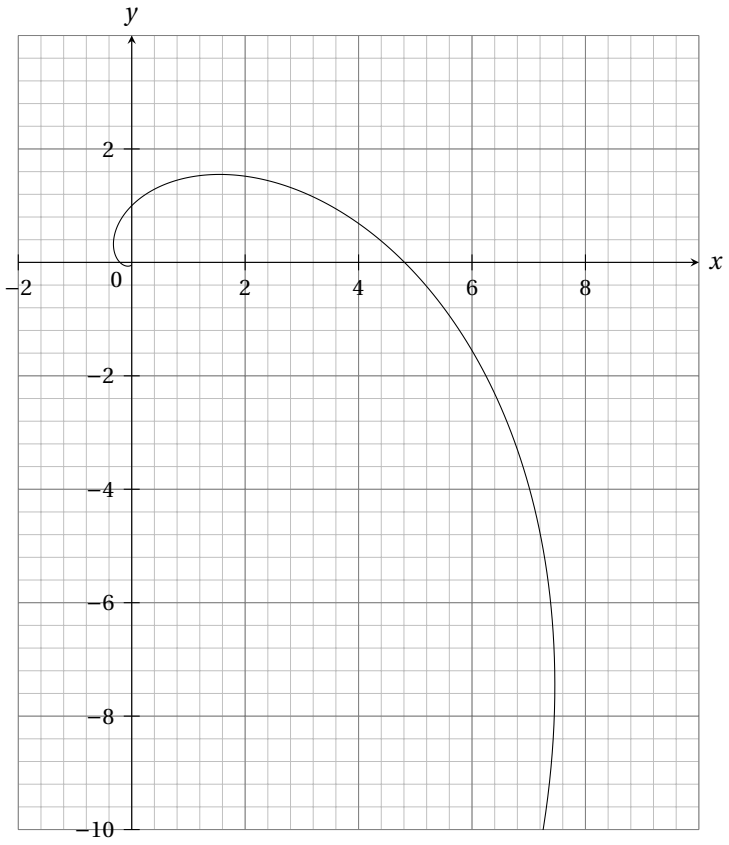
$$x(t) = t \times \sin(t)$$
$$y(t) = t \times \cos(t)$$




```
\begin{tikzpicture}[scale=1.25]
  \tkzFnInit[xmin=-50,xmax=50,xstep=10,
             ymin=-50,ymax=50,ystep=10]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnParam[smooth,samples=200,domain=0:50]{t*sin(t)}{t*cos(t)}
\end{tikzpicture}
```

12.3 Parametric curve, example 3

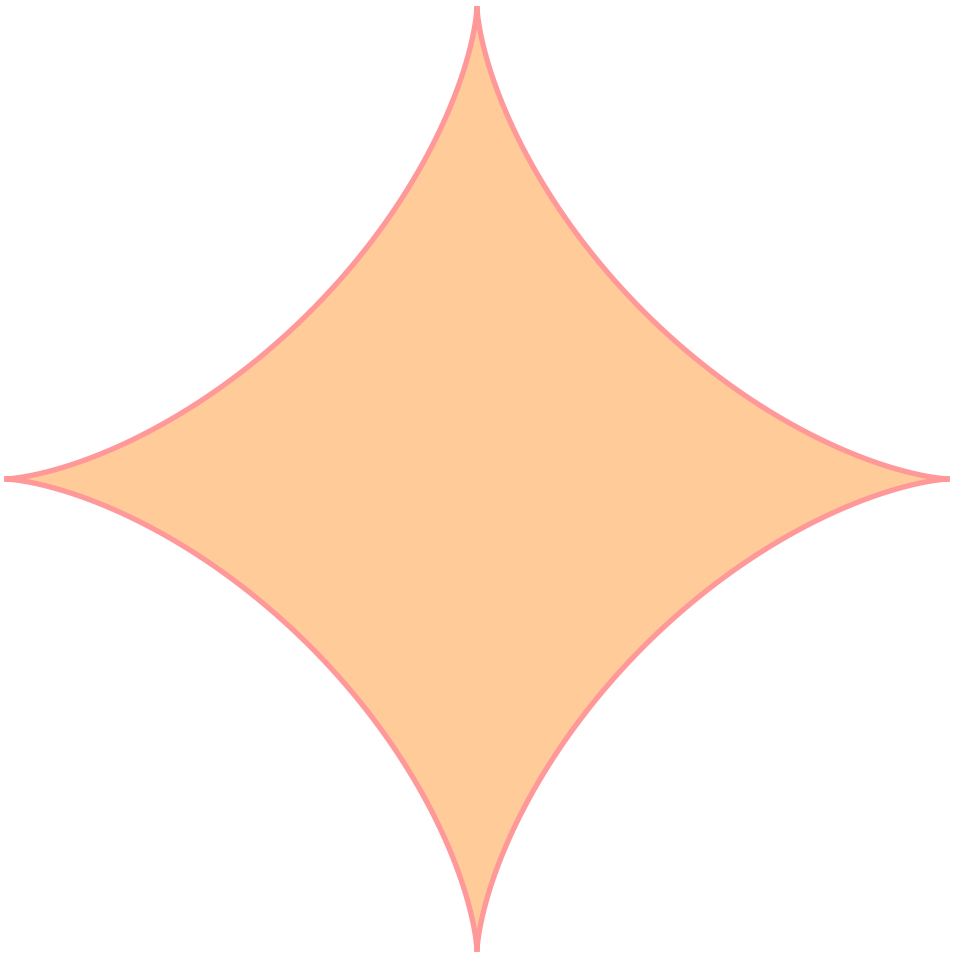
$$x(t) = \exp(t) \times \sin(t)$$
$$y(t) = \exp(t) \times \cos(t)$$




```
\begin{tikzpicture}[scale=1.5]
  \tkzFnInit[xmin=-2,xmax=10,xstep=2,ymin=-10,ymax=4,ystep=2]
  \tkzFnGrid[sub]
  \tkzFnAxes
  \tkzFnParam[clip,samples=400,domain=-pi:pi]{exp(t)*sin(t)}{exp(t)*cos(t)}
\end{tikzpicture}
```

12.4 Parametric curve, example 4

$$x(t) = \cos^3(t)$$
$$y(t) = \sin^3(t)$$



```
\begin{tikzpicture}[scale=1.25]
  \tkzFnInit[xmin=-1,xmax=1,xstep=.2,
             ymin=-1,ymax=1,ystep=.2]
  \tkzFnParam[color=red,line width=2pt,fill=orange,opacity=.4,
              samples=400,domain=0:2*pi]{cos(t)^3}{sin(t)^3}
\end{tikzpicture}
```

12.5 Parametric curve, example 5

Valentine's Day, version 1

$$x(t) = \sin^3(t)$$
$$y(t) = \cos(t) - \cos^4(t)$$



```
\begin{tikzpicture}[scale=4]
  \tkzFnInit[xmin=-1,xmax=1,ymin=-2,ymax=1]
  \tkzFnParam[samples=500,smooth,domain=-pi:pi,
              ball color=red,shading=ball]{sin(t)^3}{cos(t)-cos(t)^4}
\end{tikzpicture}
```

12.6 Parametric curve, example 6

Valentine's Day, version 2, from http://mathworld.wolfram.com/HeartCurve.html

$$x(t) = \sin(t)\cos(t)\ln(|t|)$$
$$y(t) = \sqrt{|t|}\cos(t)$$



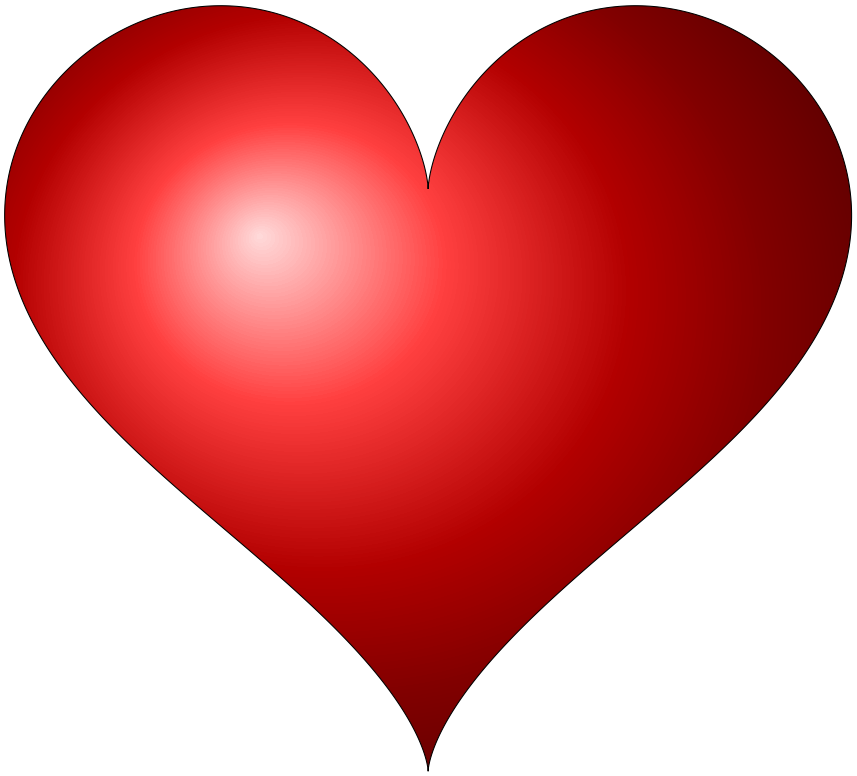
```
\begin{tikzpicture}[scale=1.5]
   \tkzFnInit[xmin=-.4,xmax=.4,xstep=.1,ymin=0,ymax=.7,ystep=.1]
   \tkzFnParam[samples=2000,smooth,domain=-1:1,
               ball color=red,shading=ball]
   {sin(t)*cos(t)*ln(abs(t))}{sqrt(abs(t))*cos(t)}
\end{tikzpicture}
```

☞ At $t = 0$, $\ln|t|$ is not defined: Lua skips this point.

12.7 Parametric curve, example 7

Valentine's Day, version 3, from http://en.wikipedia.org/wiki/Heart_(symbol)

$$x(t) = 16\sin^3(t)$$
$$y(t) = 13\cos(t) - 5\cos(2t) - 2\cos(3t) - \cos(4t)$$



```
\begin{tikzpicture}[scale=1.75]
  \tkzFnInit[xmin=-20,xmax=20,xstep=5,ymin=-25,ymax=15,ystep=5]
  \tkzFnParam[samples=400,smooth,domain=0:2*pi,
              ball color=red,shading=ball]
   {16*sin(t)^3}{13*cos(t)-5*cos(2*t)-2*cos(3*t)-cos(4*t)}
\end{tikzpicture}
```

13 Polar curves

`\tkzFnPolar[⟨options⟩]{⟨ρ(t)⟩}`

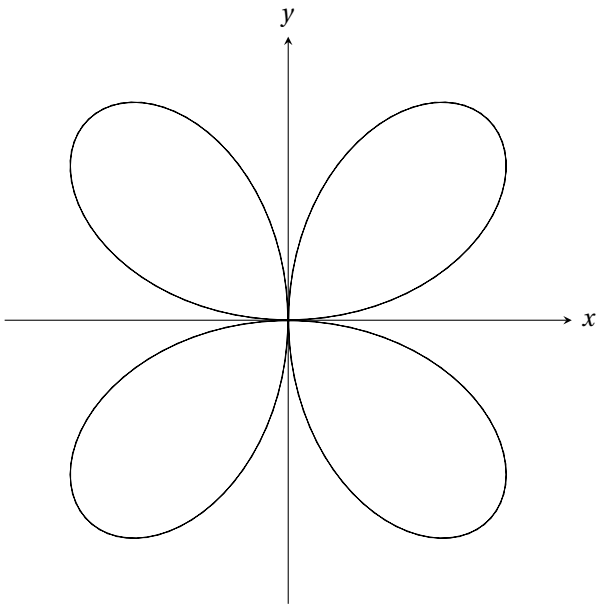
ρ(t) is a Lua expression of the variable `t`*; the point drawn is* $(\rho(t)\cos t, \rho(t)\sin t)$*. The bounds of the domain are Lua expressions:* `pi` *stands for* π*.*

options	default	definition
`domain`	`0:2*pi`	interval of the values of *t*
`samples`	`200`	number of points
`clip`	`false`	the curve is clipped to the window

Other options are TikZ options.

13.1 Polar equation, example 1

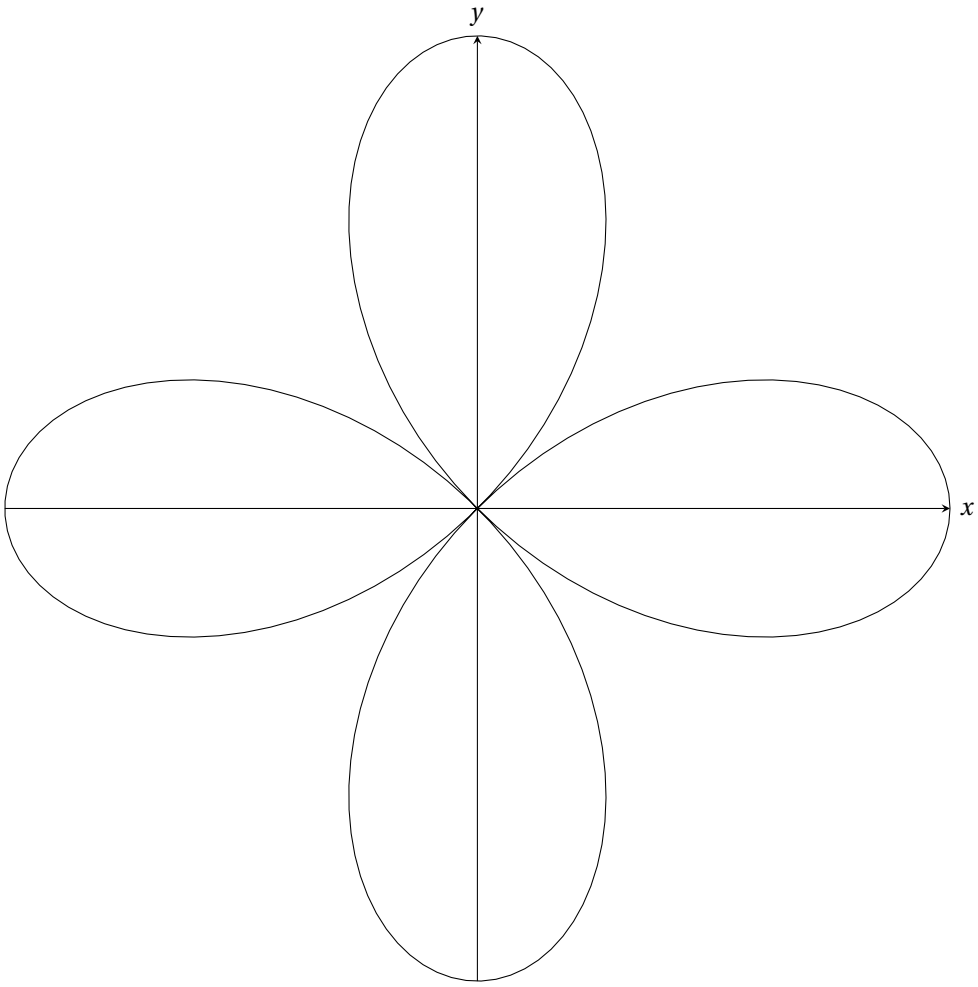
$\rho(t) = \cos(t)\sin(t)$




```
\begin{tikzpicture}[scale=0.75]
 \tkzFnInit[xmin=-0.5,xmax=0.5,
            ymin=-0.5,ymax=0.5,
            xstep=0.1,ystep=.1]
 \tkzFnAxes[ticks=false]
 \tkzFnPolar[domain=-2*pi:2*pi]{cos(t)*sin(t)}
\end{tikzpicture}
```

13.2 Polar equation, example 2

$\rho(t) = \cos(2t)$




```
\begin{tikzpicture}[scale=1.25]
   \tkzFnInit[xmin=-1,xmax=1,
              ymin=-1,ymax=1,
              xstep=.2,ystep=.2]
  \tkzFnAxes[ticks=false]
  \tkzFnPolar[domain=0:2*pi]{cos(2*t)}
\end{tikzpicture}
```

13.3 Polar equation: heart

From Mathworld : http://mathworld.wolfram.com/HeartCurve.html

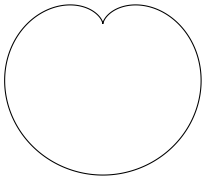
$$\rho(t) = 2 - 2\sin(t) + \sin(t)\frac{\sqrt{|\cos(t)|}}{\sin(t) + 1.4}$$



```
\begin{tikzpicture}[scale=.6]
  \tkzFnInit[xmin=-5,xmax=5,ymin=-5,ymax=5]
  \tkzFnPolar[domain=-pi:pi,
              samples=800,
              ball color=red,
              shading=ball]
    {2-2*sin(t)+sin(t)*sqrt(abs(cos(t)))/(sin(t)+1.4)}
\end{tikzpicture}
```

13.4 Polar equation, example 4

$$\rho(t) = 1 - \sin(t)$$

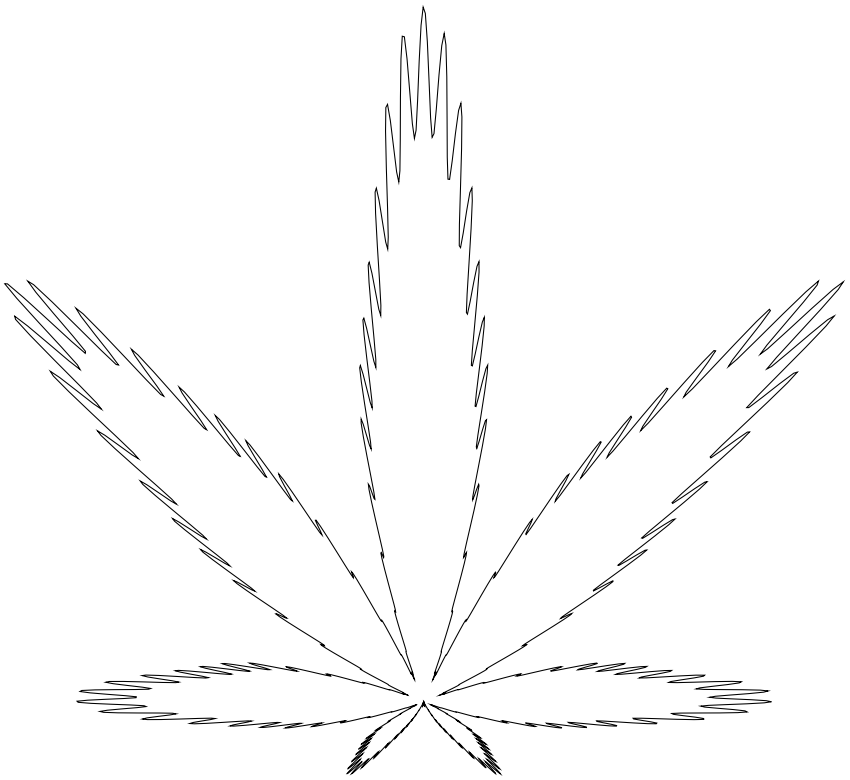


```
\begin{tikzpicture}[scale=1]
 \tkzFnInit[xmin=-5,xmax=5,ymin=-5,ymax=5]
 \tkzFnPolar[domain=0:2*pi,samples=400]{1-sin(t)}
\end{tikzpicture}
```

13.5 Polar equation: cannabis curve

Cannabis curve from mathworld : http://mathworld.wolfram.com/CannabisCurve.html

$\rho(t) = (1 + 0.9\cos(8t))(1 + 0.1\cos(24t))(1 + 0.1\cos(200t))(1 + \sin(t))$

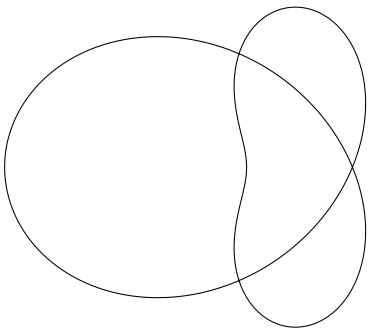


```
\begin{tikzpicture}[scale=2]
  \tkzFnInit[xmin=-5,xmax=5,ymin=-5,ymax=5]
  \tkzFnPolar[domain=0:2*pi,samples=2000]
  {(1+.9*cos(8*t))*(1+.1*cos(24*t))*(1+.1*cos(200*t))*(1+sin(t))}
\end{tikzpicture}
```

13.6 Scarabaeus curve

From mathworld : http://mathworld.wolfram.com/Scarabaeus.html

$\rho(t) = 1.6\cos(2t) - 3\cos(t)$



```
\begin{tikzpicture}[scale=1]
  \tkzFnInit[xmin=-5,xmax=5,ymin=-5,ymax=5]
  \tkzFnPolar[domain=0:2*pi,samples=400]{1.6*cos(2*t)-
3*cos(t)}
\end{tikzpicture}
```

14 Implicit curves

An implicit curve is the set of the points (x,y) such that $F(x,y)=0$: circle, ellipse, hyperbola, lemniscate, folium of Descartes, etc. The equation does not need to be solved: Lua computes F at the nodes of a grid that covers the window and finds the cells where F changes sign (*marching squares*). Each crossing point is then computed accurately by bisection, and the pieces are linked into continuous paths.

\tkzFnImplicit[⟨options⟩]{⟨$F(x,y)$⟩}

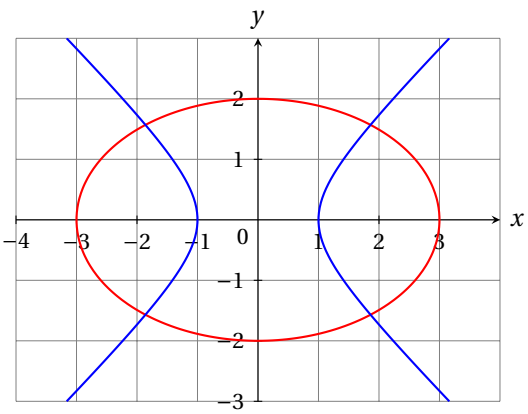
$F(x,y)$ is a Lua expression of the variables `x` *and* `y`*. An equation can also be written:* `{x^2 + y^2 = 4}` *is the same as* `{(x^2 + y^2)-(4)}`*. The curve is drawn in the window defined by* **\tkzFnInit**.

options	default	definition
`level`	`0`	value(s) c: curves $F(x,y)=c$
`samples`	`100`	number of cells on the longer side
`domain`	`xmin:xmax`	interval of the values of x
`ydomain`	`ymin:ymax`	interval of the values of y
`refine`	`true`	bisection on each edge of the grid
`clip`	`true`	the curve is clipped to the window

Other options are TikZ options. `level` *accepts a list as* **\foreach** *does:* `level={-2,-1,...,2}`.

14.1 Conics

$\frac{x^2}{9}+\frac{y^2}{4}=1$ and $x^2-y^2=1$

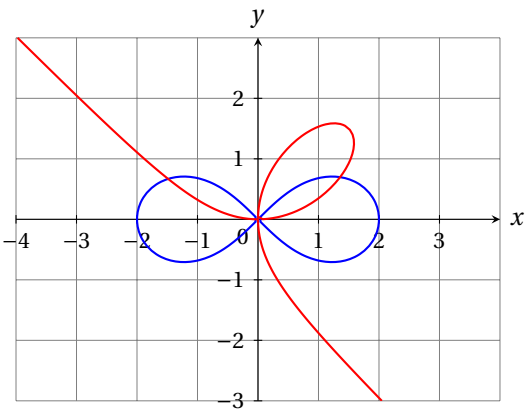



```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-4,xmax=4,ymin=-3,ymax=3]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnImplicit[thick,red]{x^2/9 + y^2/4 = 1}
  \tkzFnImplicit[thick,blue]{x^2 - y^2 = 1}
\end{tikzpicture}
```

14.2 Lemniscate of Bernoulli and folium of Descartes

$(x^2+y^2)^2=4(x^2-y^2)$ and $x^3+y^3=3xy$

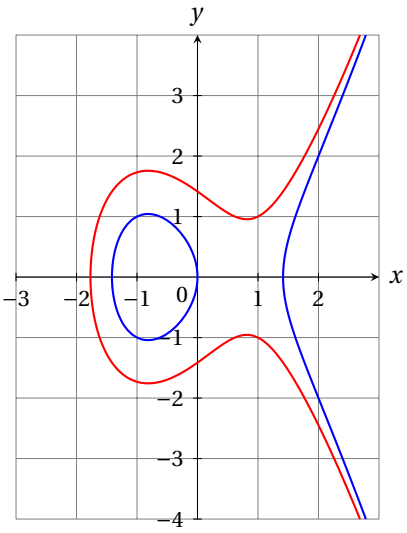
The double points at O are handled correctly.




```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-4,xmax=4,ymin=-3,ymax=3]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnImplicit[thick,blue]
    {(x^2+y^2)^2 = 4*(x^2-y^2)}
  \tkzFnImplicit[thick,red]{x^3 + y^3 = 3*x*y}
\end{tikzpicture}
```

14.3 An elliptic curve

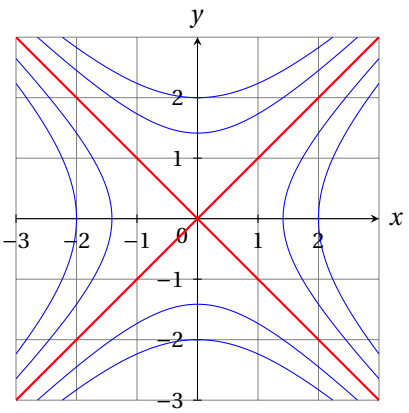
$y^2 = x^3 - 2x + 2$: the curve has one component; with $y^2 = x^3 - 2x$, it has two.




```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-3,xmax=3,ymin=-4,ymax=4]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnImplicit[thick,red]{y^2 = x^3 - 2*x + 2}
  \tkzFnImplicit[thick,blue]{y^2 = x^3 - 2*x}
\end{tikzpicture}
```

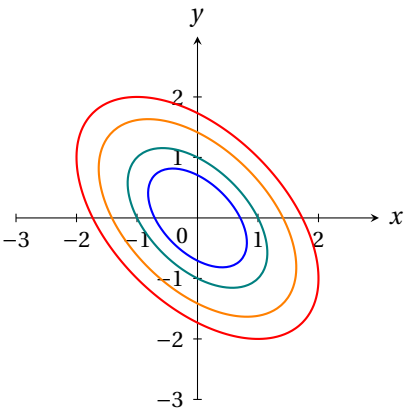
14.4 Level curves

The option `level` draws the curves $F(x,y) = c$ for several values of c. Here are the level curves of $F(x,y) = x^2 - y^2$: the level curve 0 is made of the two lines $y = x$ and $y = -x$.




```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-3,xmax=3,ymin=-3,ymax=3]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnImplicit[level={-4,-2,...,4},blue]
    {x^2 - y^2}
  \tkzFnImplicit[thick,red]{x^2 - y^2}
\end{tikzpicture}
```

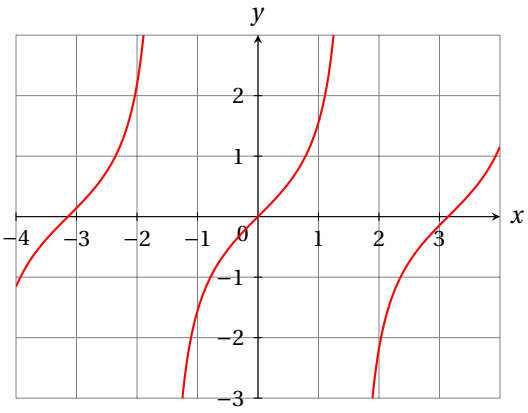
With a `\foreach` loop, each level can have its own color.




```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-3,xmax=3,ymin=-3,ymax=3]
  \tkzFnAxes
  \foreach \c/\col in {0.5/blue,1/teal,
                       2/orange,3/red}
    {\tkzFnImplicit[level=\c,thick,\col]
       {x^2 + x*y + y^2}}
\end{tikzpicture}
```

14.5 Discontinuous function

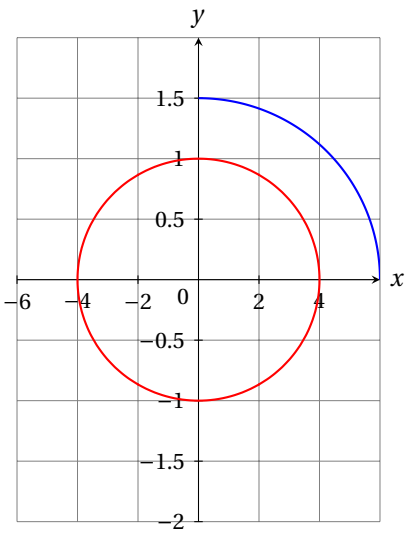
For $y = \tan(x)$, $F(x,y) = y - \tan(x)$ changes sign across the asymptotes without vanishing: these false crossings are detected and removed.




```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-4,xmax=4,ymin=-3,ymax=3]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnImplicit[thick,red]{y = tan(x)}
\end{tikzpicture}
```

14.6 Window, units and smoothing

The cells of the grid are squares on the page, even if **xstep** and **ystep** are different. The options **domain** and **ydomain** restrict the drawing to a part of the window. With **smooth**, closed curves are drawn with `smooth cycle`.



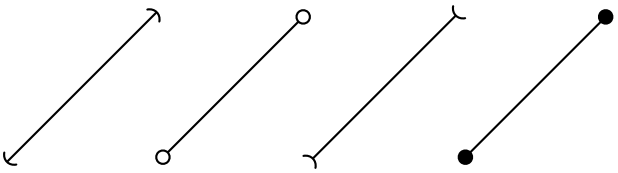

```
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=-6,xmax=6,xstep=2,ymin=-2,ymax=2,ystep=.5]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnImplicit[thick,red,smooth,samples=600]{x^2/16 + y^2 = 1}
  \tkzFnImplicit[thick,blue,domain=0:6,ydomain=0:2]{x^2/16 + y^2 = 1.5^2}
\end{tikzpicture}
```

14.7 Limits of the method

- A curve along which F does not change sign is not found: $(x-y)^2=0$ gives nothing, write $x-y=0$ instead.
- A detail smaller than a cell of the grid may be missed: increase **samples** (for example `samples=200`).
- Near a singular point, the drawing is accurate up to the size of a cell.

15 Symbols

Some authors add symbols to the curves to give more information to the reader. Here are a few possibilities (Ti*k*Z library **arrows**):

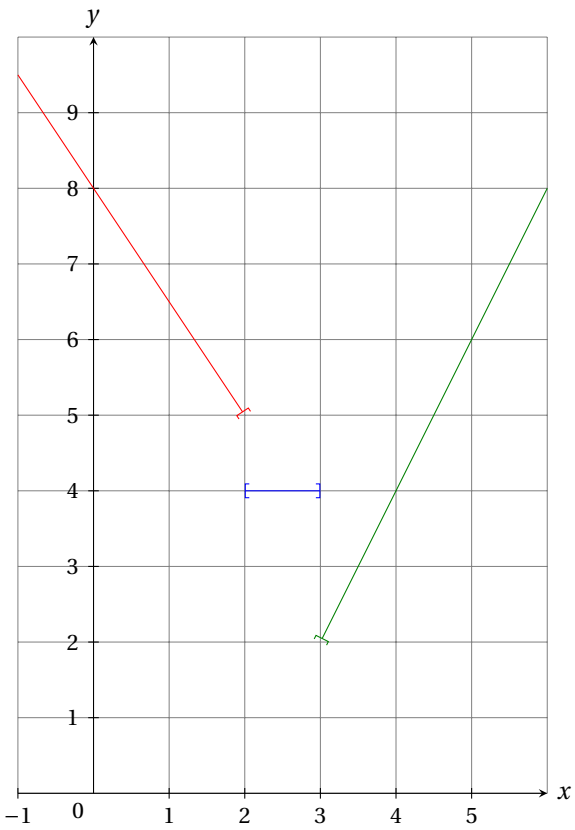


The following example is by **Simon Schläpfer**:

We want to draw

$$y = \begin{cases} 8 - 1.5x & \text{, if } x < 2 \\ 4 & \text{, if } 2 \le x \le 3 \\ 2x - 4 & \text{, if } x > 3 \end{cases}$$

The curves must not be clipped to the window (**clip=false**), otherwise the symbols at the ends would be cut.




```
\begin{tikzpicture}
  \tkzFnInit[xmin=-1,xmax=6,ymin=0,ymax=10]
  \tkzFnGrid[color=gray]
  \tkzFnAxes
  \tkzFnDraw[{-[},clip=false,color=red,domain=-1:2,samples=2]{8-1.5*x}
  \tkzFnDraw[{[-]},clip=false,color=blue,domain=2:3,samples=2]{4}
  \tkzFnDraw[{]-},clip=false,color=green!50!black,domain=3:6,samples=2]{2*x-4}
\end{tikzpicture}
```

16 Writing the results: American or French format

The numbers written by **tkz-fn** (tick labels, coordinates, results of computations) follow one of the two following conventions:

	American format (us)	French format (fr)
decimal separator	3.14	3,14
thousands separator	12,345	12 345 (thin space)
intervals	(a, b), $[a, b]$	$]a\,;b[$, $[a\,;b]$
coordinates	$A(2, 3)$	$A(2\,;3)$
origin	0	O

Thousands are separated only from five digits on: 1234 but 12,345.

16.1 Choosing the format

- By default, the format is French if the language of **babel** is French at the beginning of the document, American otherwise.
- The package options force the format: `\usepackage[us]{tkz-fn}` or `\usepackage[fr]{tkz-fn}`.
- In the document: `\tkzFnSetup{format=fr}`.

\tkzFnSetup{⟨options⟩}

options	default	definition
format	us or fr	format of the numbers
decimals	2	maximum number of decimal places of **\tkzFnPrint**
axes	school	presentation of the axes: school or box
arrows	end	arrows of the axes: end, both or none

16.2 Writing a result: \tkzFnPrint

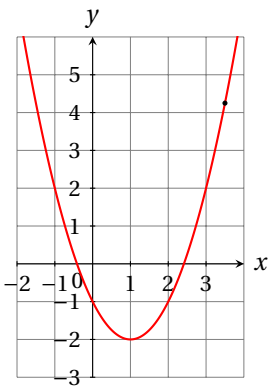
\tkzFnPrint[⟨decimals⟩]{⟨Lua expression⟩}

Computes the expression and writes the rounded result, without useless zeros. The functions defined with **\tkzFnDef** *can be used, as well as* `diff(f,x)`, `integral(f,a,b)` *and* `riemann(f,a,b,n,"left")`.

\tkzFnPrintInterval[⟨oo, cc, oc or co⟩]{⟨a⟩}{⟨b⟩}

Writes an interval; o = *open,* c = *closed. The endpoints are Lua expressions or TEX text (*`-\infty`*).*

16.3 Example



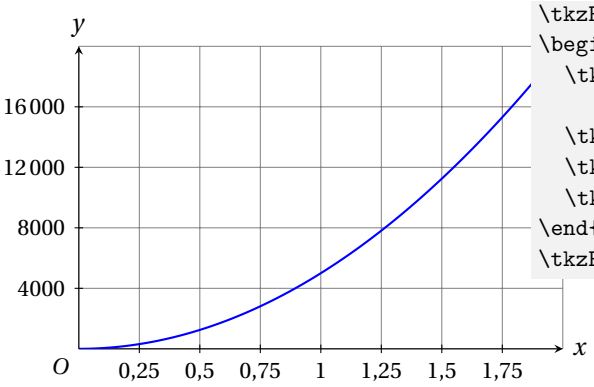
$A(3.5, 4.25)$, $f'(3) = 4$, -1.4375, 1,234,567.89, $I = (-\infty, 2.5]$
$A(3{,}5\,;\,4{,}25)$, $f'(3) = 4$, $-1{,}4375$, 1 234 567,89, $I =]-\infty\,;\,2{,}5]$

```
\tkzFnDef{f}{x^2-2*x-1}
\begin{tikzpicture}[scale=.5]
  \tkzFnInit[xmin=-2,xmax=4,
             ymin=-3,ymax=6]
  \tkzFnGrid \tkzFnAxes
  \tkzFnDraw[thick,red](f)
  \tkzFnDefPoint[draw](f,3.5){A}
\end{tikzpicture}

$A\tkzFnPrintCoord(A)$,
$f'(3)=\tkzFnPrint{diff(f,3)}$,
$\tkzFnPrint[4]{f(0.25)}$,
$\tkzFnPrint{1234567.891}$,
$I=\tkzFnPrintInterval[oc]{-\infty}{2.5}$

\tkzFnSetup{format=fr}
$A\tkzFnPrintCoord(A)$,
$f'(3)=\tkzFnPrint{diff(f,3)}$,
$\tkzFnPrint[4]{f(0.25)}$,
$\tkzFnPrint{1234567.891}$,
$I=\tkzFnPrintInterval[oc]{-\infty}{2.5}$
\tkzFnSetup{format=us}
```

16.4 Tick labels in French format




```
\tkzFnSetup{format=fr}
\begin{tikzpicture}[scale=.8]
  \tkzFnInit[xmin=0,xmax=2,xstep=.25,
             ymin=0,ymax=20000,ystep=4000]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[blue,thick]{5000*x^2}
\end{tikzpicture}
\tkzFnSetup{format=us}
```

17 With tkz-elements

tkz-elements works in centimeters, **tkz-fn** in the units of the coordinate system. When **tkz-elements** is loaded, every point defined with **\tkzFnDefPoint** or **\tkzFnDefPointXY** is also stored in the table z, with its coordinates in centimeters:

$$\texttt{z.A} = \left(\frac{x_A}{\text{xstep}}, \frac{y_A}{\text{ystep}}\right)$$

The point can then be used with the objects of **tkz-elements**: `circle(z.O, z.A)`, `line(z.A, z.B)`, etc. In the other direction, `tkzfn.point_from_z("B")` brings the point `z.B` back into the coordinate system, so that its coordinates can be written with **\tkzFnPrintCoord**.

If the units of the two axes are different, `circle(z.O, z.A)` is a circle on the page, not a circle in the coordinate system.

tkz-elements is not loaded by **tkz-fn**; the following example needs `\usepackage{tkz-elements}`.

```
\tkzFnDef{f}{x^2/2}
\begin{tikzpicture}
  \tkzFnInit[xmin=-3,xmax=3,ymin=-2,ymax=6,ystep=2]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[thick,red](f)
  \tkzFnDefPoint(f,2){A}       % A(2,2) in the system, z.A = (2,1) in cm
  \tkzFnDefPointXY(0,0){O}
  \directlua{
    C.OA = circle(z.O, z.A)     % tkz-elements works in cm
    z.B  = C.OA.south
  }
  \tkzGetNodes
  \draw[blue] (O) circle [radius=\tkzUseLua{C.OA.radius}];
  \tkzFnDrawPoints[red](A,O)
  \fill[blue] (B) circle (1.5pt) node[below right] {$B$};
\end{tikzpicture}

\directlua{tkzfn.point_from_z("B")}
$B\tkzFnPrintCoord(B)$   % (0, -4.47) in the coordinate system
```

17.1 In Lua

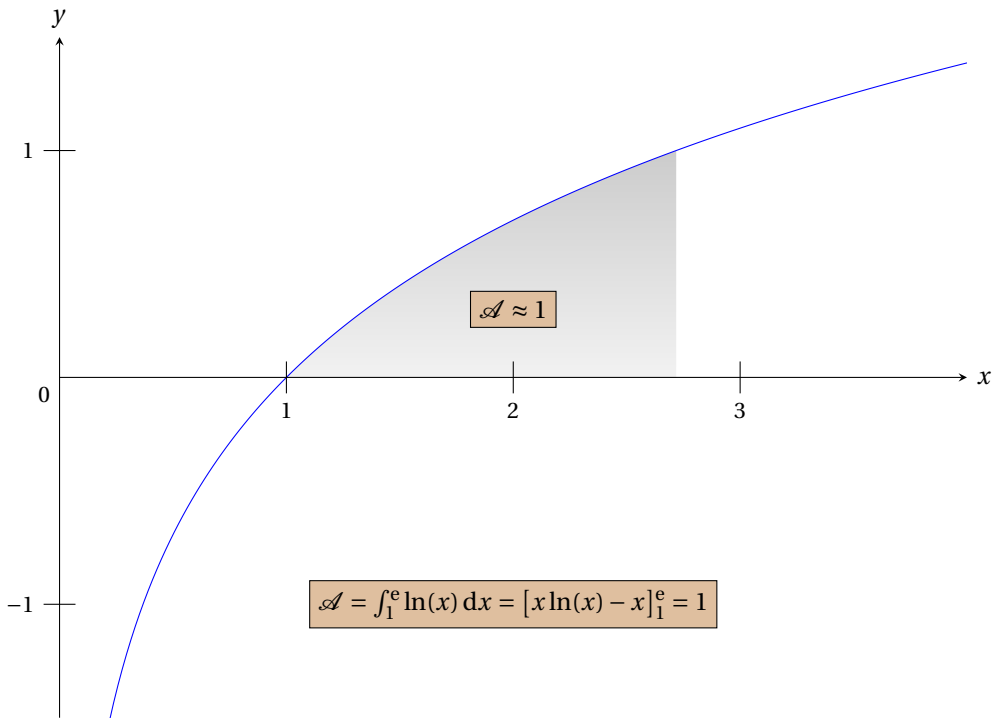
The objects of **tkz-fn** are available in Lua:

- `FN.f(2)`: value of the function f at 2;
- `tkzfn.fct("sin(x)")`: creates a function (`FN.g = tkzfn.fct("sin(x)")`); a Lua function can also be given;
- `tkzfn.derivative(f, x)`, `tkzfn.integral(f, a, b)`;
- `tkzfn.to_page(x, y)` and `tkzfn.to_user(X, Y)`: conversions between the coordinate system and the page;
- `tkzfn.points.A`: coordinates of A in the coordinate system.

18 A few examples

18.1 TikZ and tkz-fn

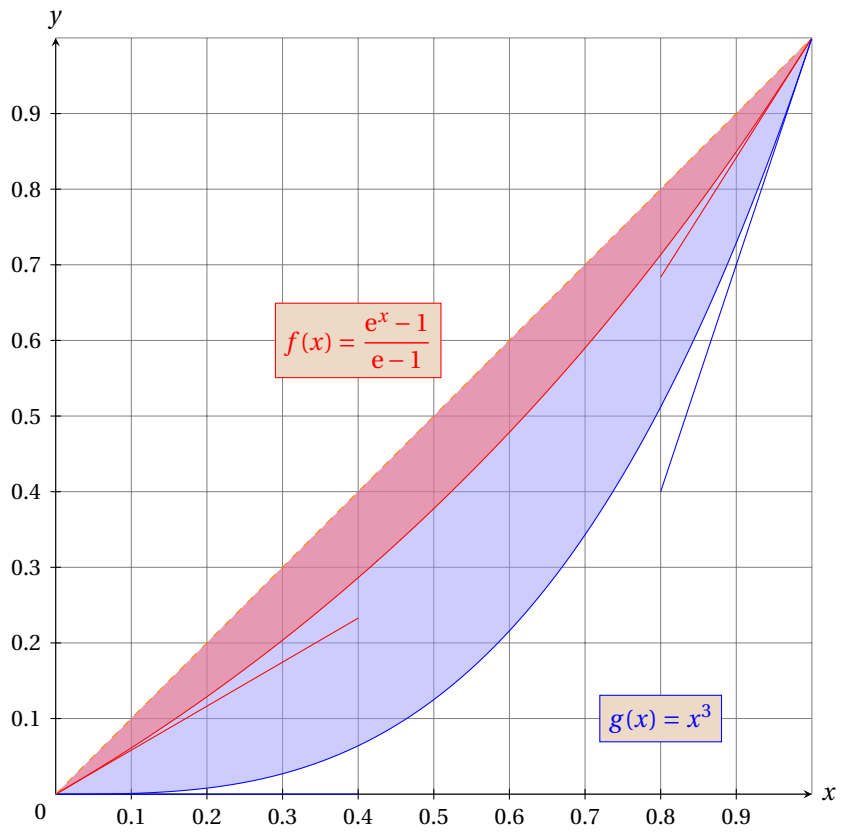
TikZ code and **tkz-fn** code can be mixed. Here the region is filled with a TikZ shading; the value of the integral is computed by Lua.




```
\begin{tikzpicture}[scale=3]
  \tkzFnInit[xmin=0,xmax=4,ymin=-1.5,ymax=1.5]
  \tkzFnAxisY
  \tkzFnDef{f}{ln(x)}
  \tkzFnArea[fill=none,top color=gray!80,bottom color=gray!20,
             domain=1:e](f)
  \tkzFnDraw[color=blue,domain=0.2:4](f)
  \tkzFnAxisX
  \tkzFnText[draw,color=black,fill=brown!50](2,-1)
    {$\mathcal{A} = \int_1^{\text{e}}\ln(x)\,\text{d}x =
     \big[x\ln(x)-x\big]_{1}^{\text{e}} = 1$}
  \tkzFnText[draw,color=black,fill=brown!50](2,0.3)
    {$\mathcal{A}\approx\tkzFnPrint[6]{integral(f,1,e)}$}
\end{tikzpicture}
```

18.2 Lorenz curves

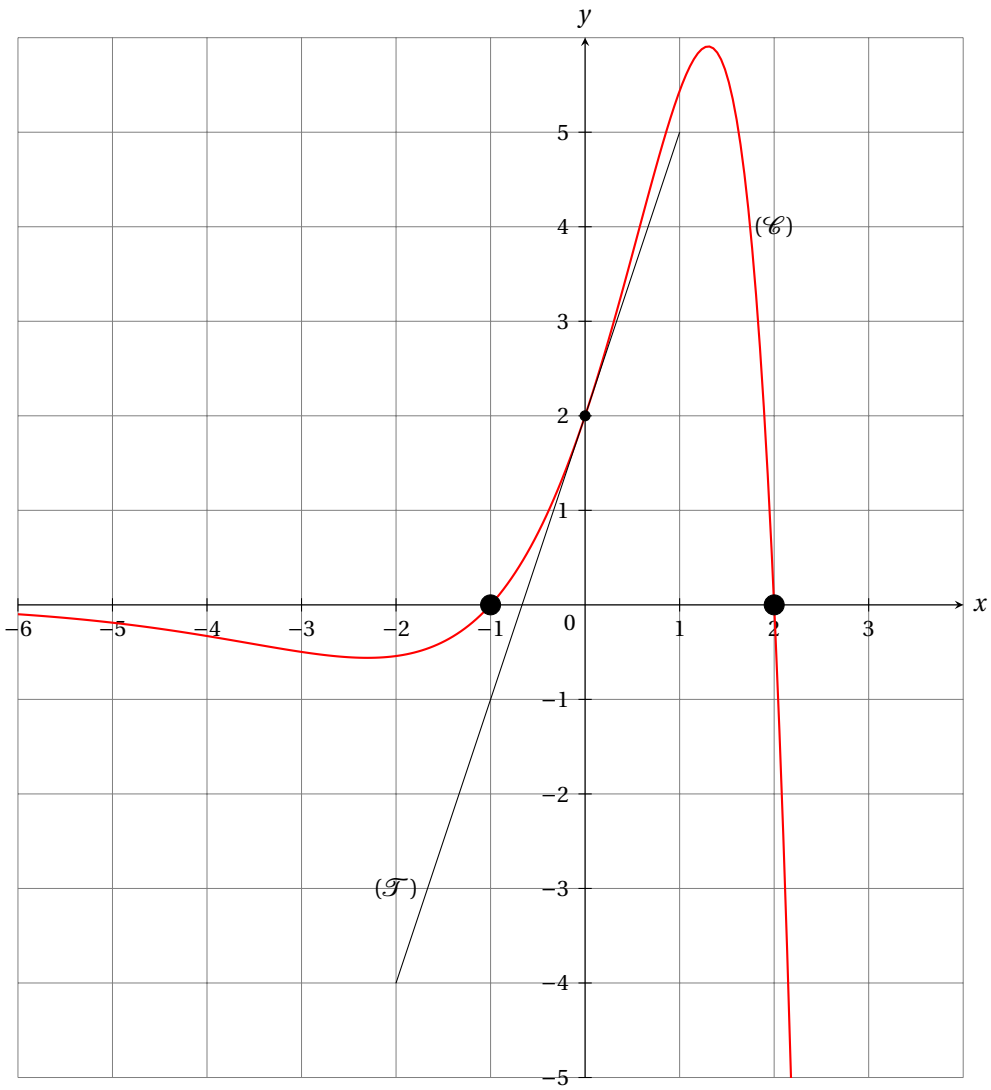
$f(x) = \dfrac{e^x - 1}{e - 1}$ and $g(x) = x^3$




```
\begin{tikzpicture}[scale=1]
  \tkzFnInit[xmin=0,xmax=1,ymin=0,ymax=1,xstep=0.1,ystep=0.1]
  \tkzFnGrid(0,0)(1,1)
  \tkzFnAxes
  \tkzFnDef{f}{(exp(x)-1)/(exp(1)-1)}
  \tkzFnDef{g}{x^3}
  \tkzFnDef{h}{x}
  \tkzFnArea[fill=blue!40,domain=0:1](h,g)
  \tkzFnArea[fill=red!60,domain=0:1](h,f)
  \tkzFnDraw[color=red](f)
  \tkzFnTangent[kl=0,kr=0.4,color=red](f,0)
  \tkzFnTangent[kl=0.2,kr=0,color=red](f,1)
  \tkzFnText[draw,color=red,fill=brown!30](0.4,0.6)
          {$f(x)=\dfrac{\text{e}^x-1}{\text{e}-1}$}
  \tkzFnDraw[color=blue](g)
  \tkzFnTangent[kl=0,kr=0.4,color=blue](g,0)
  \tkzFnTangent[kl=0.2,kr=0,color=blue](g,1)
  \tkzFnText[draw,color=blue,fill=brown!30](0.8,0.1){$g(x)=x^3$}
  \tkzFnDraw[color=orange,dashed](h)
\end{tikzpicture}
```

18.3 Exponential curve

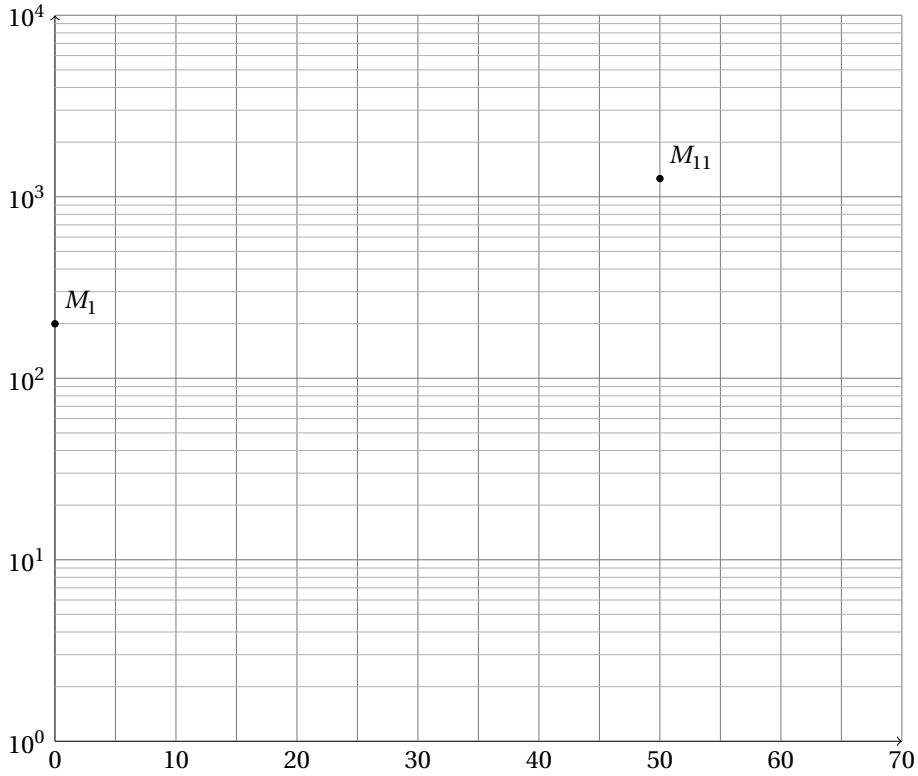
$f(x) = (-x^2 + x + 2)e^x$




```
\begin{tikzpicture}[scale=1.25]
   \tkzFnInit[xmin=-6,xmax=4,ymin=-5,ymax=6]
   \tkzFnGrid
   \tkzFnAxes
   \tkzFnDraw[name=f,color=red,thick,domain=-6:2.1785]{(-x*x+x+2)*exp(x)}
   \tkzFnTangent[draw,kl=2](f,0)
   \tkzFnDefPointXY[draw,size=3pt](2,0){b}
   \tkzFnDefPointXY[draw,size=3pt](-1,0){c}
   \tkzFnText(2,4){($\mathcal{C}$)}
   \tkzFnText(-2,-3){($\mathcal{T}$)}
\end{tikzpicture}
```

18.4 Logarithmic axis

Here the coordinate system is only used to place the lines; the labels are written with Ti*k*Z.



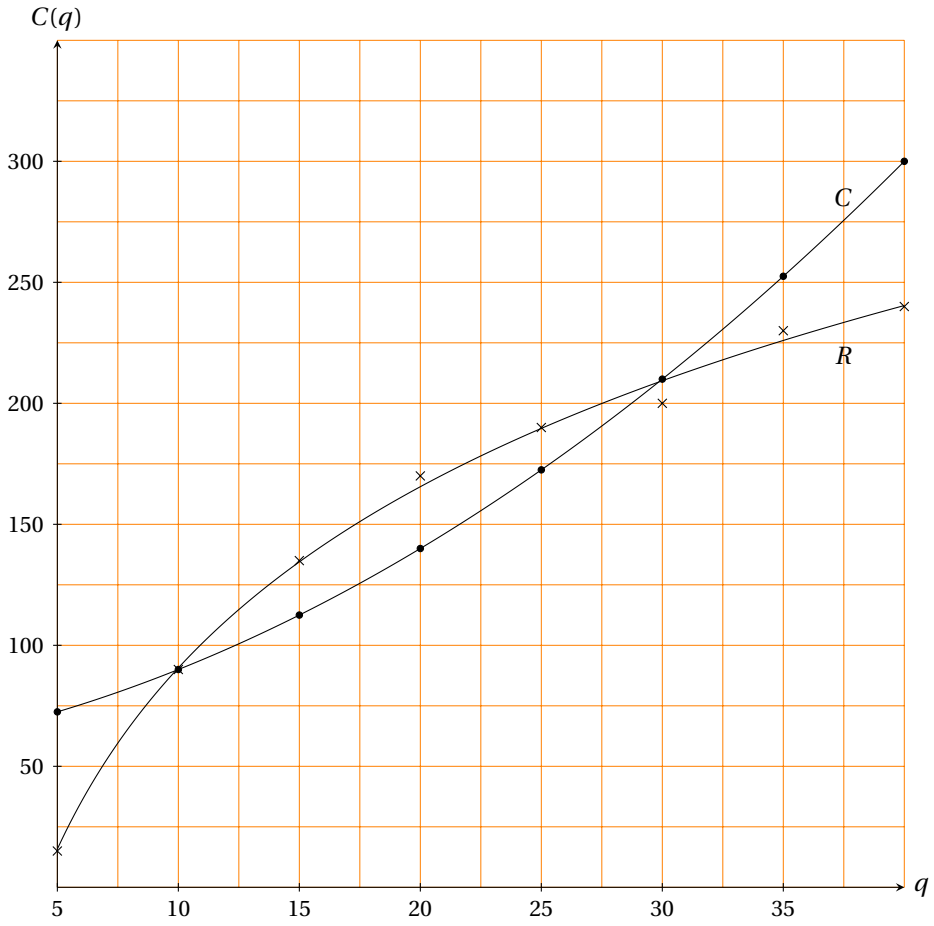

```
\begin{tikzpicture}[scale=0.8]
 \tkzFnInit[xmin=0,xmax=14,ymin=0,ymax=12]
 \draw[thin,->] (0,0) -- (14,0);
 \draw[thin,->] (0,0) -- (0,12);
 \foreach \x/\xtext in {0/0,2/10,4/20,6/30,8/40,10/50,12/60,14/70}
     {\draw[shift={(\x,0)}] node[below] {$\xtext$};}
 \foreach \y/\z in {0/0,3/1,6/2,9/3,12/4}
     {\draw[shift={(0,\y)}] node[left] {$10^{\z}$};}
 \tkzFnVLines[gray,thin]{1,2,...,14}
 \tkzFnHLines[gray,thin]{3,6,...,12}
 \foreach \y in {0,3,...,9}{
   \tkzFnHLines[thin,gray!60]{\y+3*log10(2),\y+3*log10(3),\y+3*log10(4),
     \y+3*log10(5),\y+3*log10(6),\y+3*log10(7),\y+3*log10(8),\y+3*log10(9)}}
 \tkzFnDefPointXY(0,6.90){a}
 \tkzFnDefPointXY(10,9.30){b}
 \tkzFnDrawPoints(a,b)
 \tkzFnLabelPoint(a){$M_{1}$}
 \tkzFnLabelPoint(b){$M_{11}$}
\end{tikzpicture}
```

The values `0.903`, `1.431`, … used with **`tkz-fct`** are replaced by their expressions: $3\log 2 \approx 0.903$.

18.5 A bit of everything

`\tkzFnDrawData[⟨TikZ options⟩]{⟨x1/y1,x2/y2,…⟩}`

Draws points given by their coordinates in the coordinate system (equivalent of `\tkzDefSetOfPoints` *and* `\tkzDrawSetOfPoints`*). The options are those of a TikZ* `plot`*; default:* `only marks, mark=*`.




```
\begin{tikzpicture}[scale=.8]
 \tkzFnInit[xmin=5,xmax=40,ymin=0,ymax=350,xstep=2.5,ystep=25]
 \tkzFnGrid[color=orange]
 \tkzFnAxes[xlabel=$q$,ylabel=$C(q)$,dx=5,dy=50]
 \tkzFnDraw[name=C,domain=5:40]{0.1*x^2+2*x+60}
 \tkzFnMarkPoints(C){5,10,...,40}
 \tkzFnDraw[domain=5:40]{100*ln(x)-158}
 \tkzFnText(37.5,285){$C$}
 \tkzFnText(37.5,220){$R$}
 \tkzFnDrawData[mark=x,mark size=3pt]
   {5/15,10/90,15/135,20/170,25/190,30/200,35/230,40/240}
\end{tikzpicture}
```

18.6 Interpolation

We look for an interpolating polynomial on the interval $[-1,1]$ of the function f defined by:

$$f(x) = \frac{1}{1+8x^2}$$

The interpolating polynomial is the Lagrange polynomial built on twenty-one equally spaced points; its degree is 20.

With **tkz-fct**, the coefficients of the polynomial had to be computed elsewhere and the polynomial written in Horner form. With Lua, the polynomial is computed directly from the points: a Lua function is stored in the table FN with tkzfn.fct. The graph of f is in red, the interpolating polynomial in blue, the points used in yellow.

18.6.1 The code

```
\tkzFnDef{f}{1/(1+8*x^2)}
\directlua{
  local xs, ys = {}, {}
  for i = 0, 20 do
    xs[i] = -1 + i/10
    ys[i] = FN.f(xs[i])
  end
  FN.P = tkzfn.fct(function(x)
    local s = 0
    for i = 0, 20 do
      local l = 1
      for j = 0, 20 do
        if j == i then else l = l*(x-xs[j])/(xs[i]-xs[j]) end
      end
      s = s + ys[i]*l
    end
    return s
  end)
}
\begin{tikzpicture}
  \tkzFnInit[xmin=-1,xmax=1,ymin=-1.8,ymax=1.2,xstep=0.1,ystep=0.2]
  \tkzFnGrid
  \tkzFnAxes
  \tkzFnDraw[samples=400,line width=4pt,color=red,opacity=.5](f)
  \tkzFnDraw[samples=400,line width=1pt,color=blue](P)
  \tikzset{tkzfn point/.style={draw=black,fill=yellow}}
  \tkzFnMarkPoints[size=3pt](f){-1,-0.9,...,1}
\end{tikzpicture}
```

The result is on the next page, where the Runge phenomenon can be seen.

18.6.2 The figure



Figure 1: Interpolation: $\frac{1}{1+8x^2}$

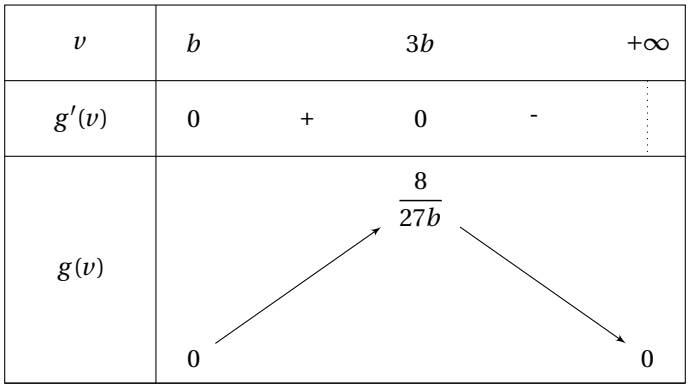
18.7 Van der Waals curves

Let v be the volume of a mass of fluid and p its pressure. b and k are two positive real numbers. We study a formula proposed by Van der Waals that relates these variables.

$$p(v) = \frac{-3}{v^2} + \frac{3k}{v-b}$$

defined on the interval $I = (b, +\infty)$.

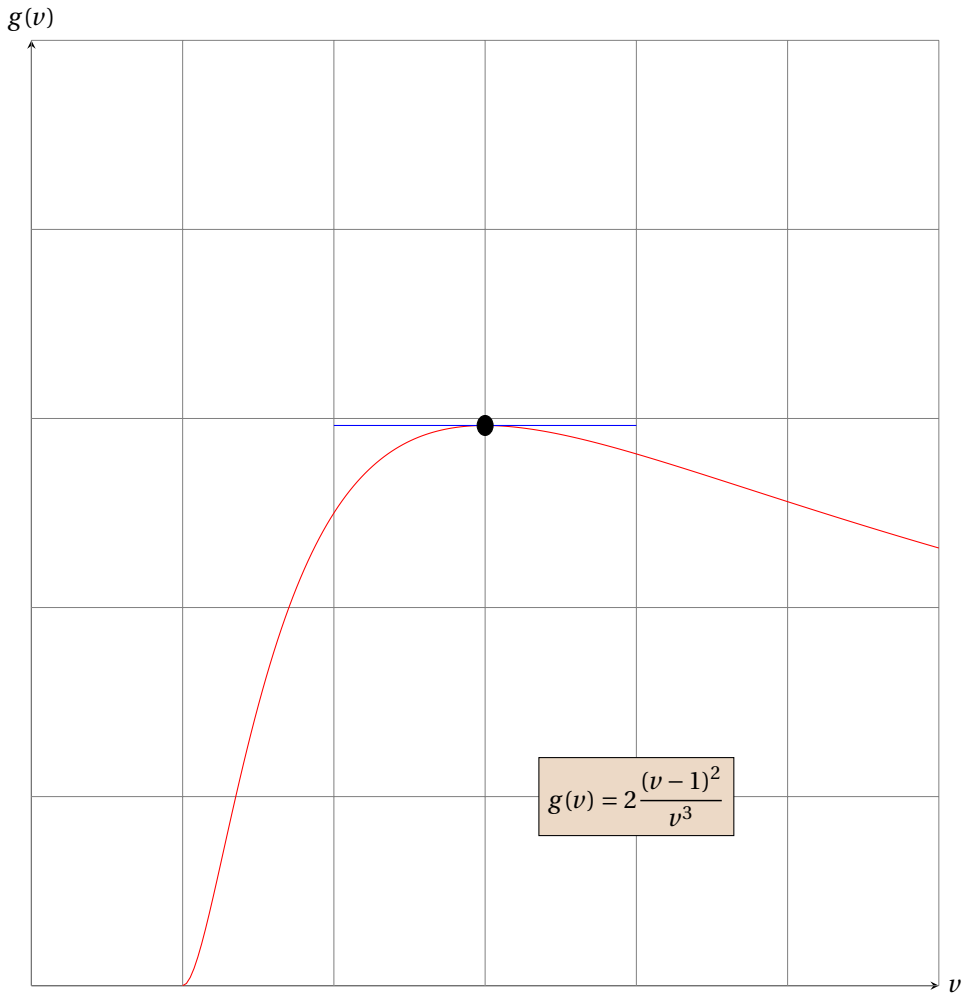
18.7.1 Variation table




```
\begin{tikzpicture}
\tkzTab
{ $v$      /1,
  $g'(v)$  /1,
  $g(v)$   /3
}
{ $b$ ,
  $3b$ ,
  $+\infty$
}
{0,$+$,$0$,$-$,t}
{-/           $0$    /,
 +/$\dfrac{8}{27b}$  /,
 -/  $0$             /}
\end{tikzpicture}
```

18.7.2 First curve, b=1

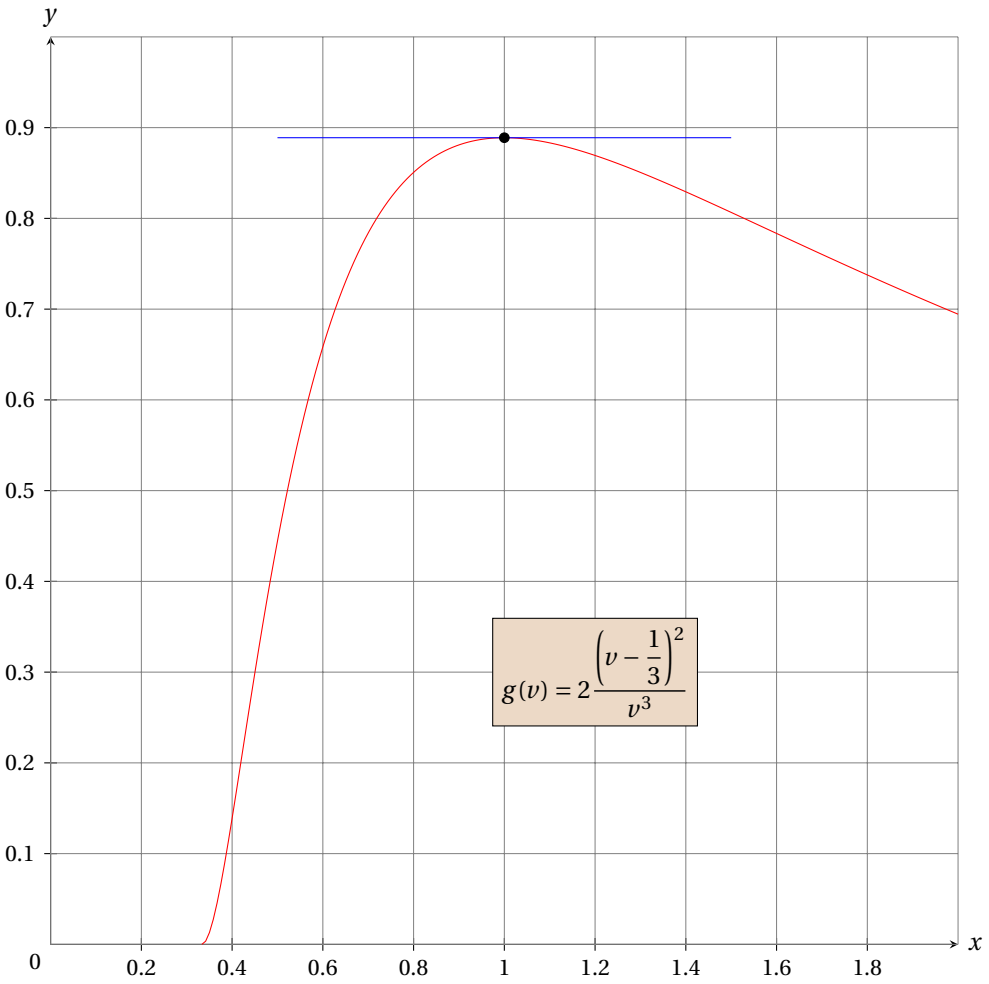
A few curves for $1 \leq v \leq 6$




```
\begin{tikzpicture}[xscale=2,yscale=2.5]
   \tkzFnInit[xmin=0,xmax=6,ymin=0,ymax=0.5,ystep=0.1]
   \tkzFnAxisX[ticks=false,label=$v$]
   \tkzFnAxisY[ticks=false,label=$g(v)$]
   \tkzFnGrid(0,0)(6,0.5)
   \tkzFnDraw[name=g,color=red,domain=1:6]{(2*(x-1)*(x-1))/(x*x*x)}
   \tkzFnTangent[color=blue,draw](g,3)
   \tkzFnText[draw,fill=brown!30](4,0.1){$g(v)=2\dfrac{(v-1)^2}{v^3}$}
\end{tikzpicture}
```

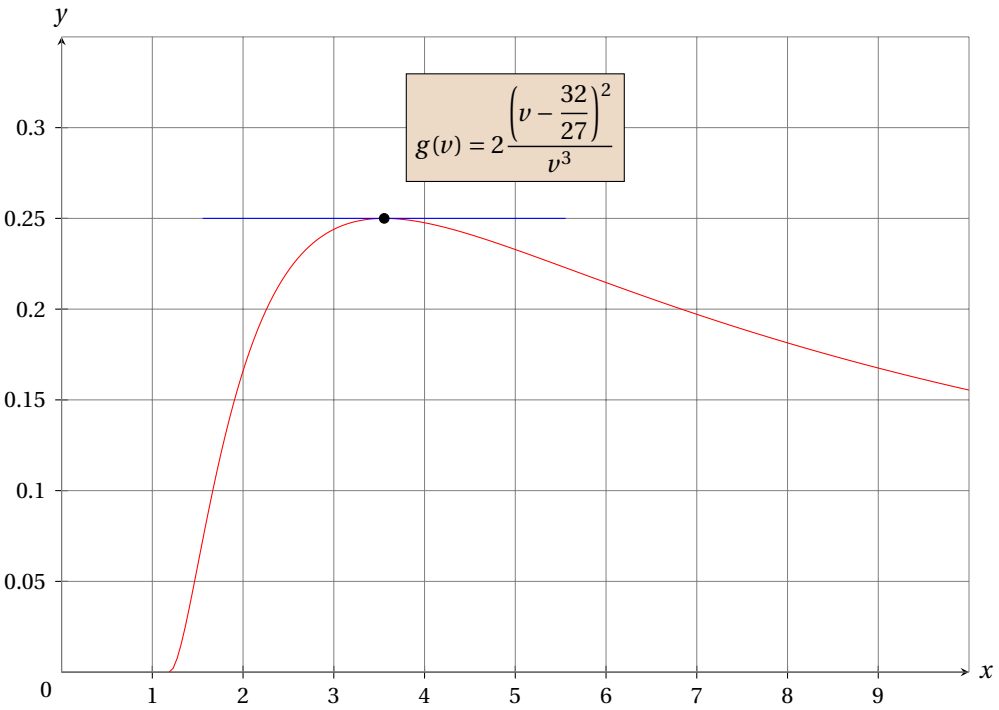
The maximum is reached at $v = 3$: $g(3) = \dfrac{8}{27} \approx 0.2963$.

18.7.3 Second curve, b=1/3




```
\begin{tikzpicture}[scale=1.2]
  \tkzFnInit[xmin=0,xmax=2,xstep=0.2,ymin=0,ymax=1,ystep=0.1]
  \tkzFnAxes
  \tkzFnGrid(0,0)(2,1)
  \tkzFnDraw[name=g,color=red,domain=1/3:2]{(2*(x-1/3)^2)/x^3}
  \tkzFnTangent[draw,color=blue,kr=.5,kl=.5](g,1)
  \tkzFnText[draw,fill=brown!30](1.2,0.3)
             {$g(v)=2\dfrac{\left(v-\dfrac{1}{3}\right)^2}{v^3}$}
\end{tikzpicture}
```

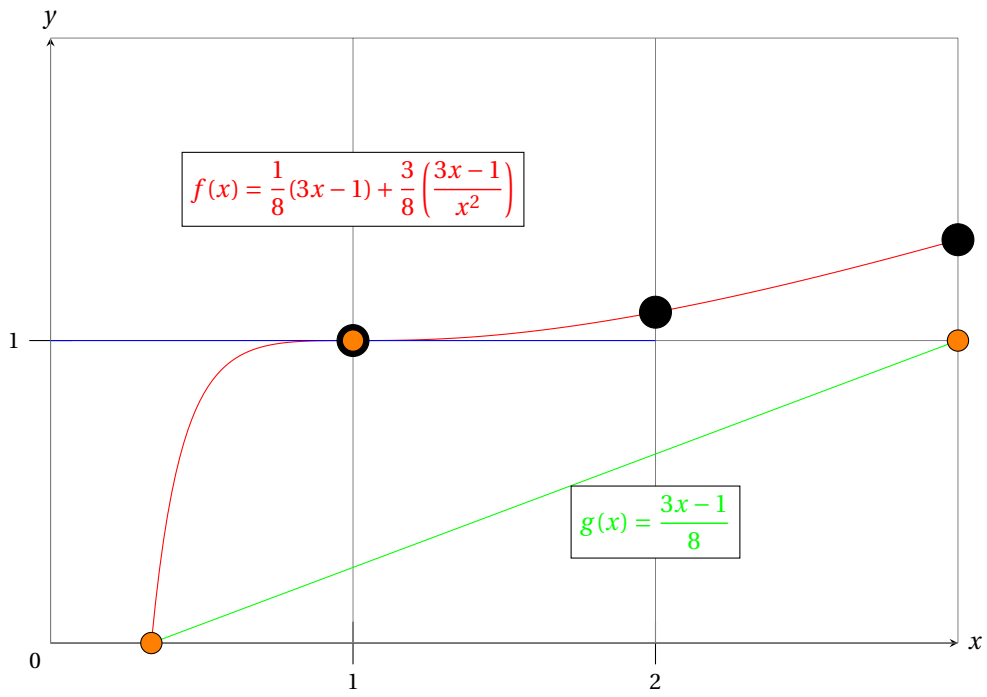
18.7.4 Third curve, b=32/27




```
\begin{tikzpicture}[scale=1.2]
    \tkzFnInit[xmin=0,xmax=10,ymin=0,ymax=.35,ystep=0.05]
    \tkzFnAxes
    \tkzFnGrid(0,0)(10,.35)
    \tkzFnDraw[name=g,color=red,domain=32/27:10]{(2*(x-32/27)^2)/x^3}
    \tkzFnTangent[draw,color=blue,kr=2,kl=2](g,{32/9})
    \tkzFnText[draw,fill=brown!30](5,0.3)
            {$g(v)=2\dfrac{\left(v-\dfrac{32}{27}\right)^2}{v^3}$}
\end{tikzpicture}
```

18.8 Critical values

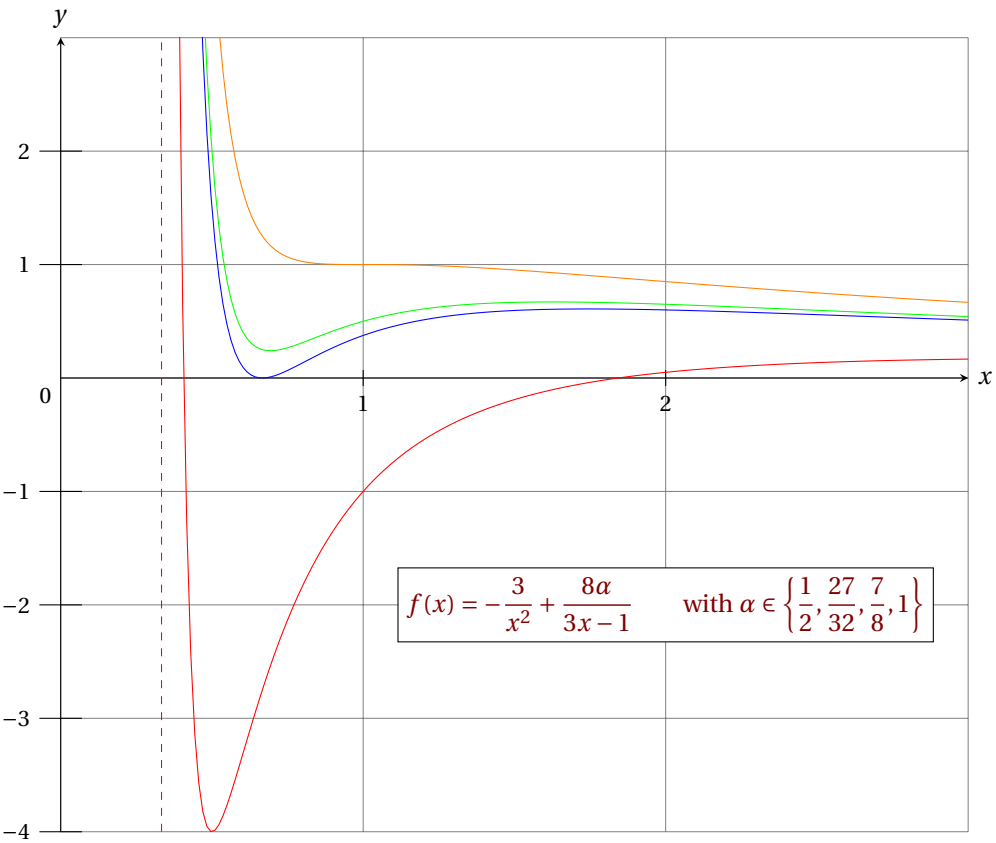
18.8.1 Van der Waals curves




```
\begin{tikzpicture}[scale=4]
  \tkzFnInit[xmin=0,xmax=3,ymin=0,ymax=2]
  \tkzFnAxes
  \tkzFnGrid(0,0)(3,2)
  \tkzFnDraw[name=f,color=red,domain=1/3:3]{0.125*(3*x-1)+0.375*(3*x-1)/(x*x)}
  \tkzFnMarkPoints(f){2,3}
  \tkzFnTangent[draw,color=blue](f,1)
  \tkzFnDraw[name=g,color=green,domain=1/3:3]{0.125*(3*x-1)}
  \tkzFnMarkPoints[size=1pt,fill=orange](g){1/3,3}
  \tkzFnDefPointXY[draw,size=1pt,fill=orange](1,1){F}
  \tkzFnText[draw,fill=white,text=red](1,1.5)
  {$f(x)=\dfrac{1}{8}(3x-1)+\dfrac{3}{8}\left(\dfrac{3x-1}{x^2}\right)$}
  \tkzFnText[draw,fill=white,text=green](2,0.4){$g(x) = \dfrac{3x-1}{8}$}
\end{tikzpicture}
```

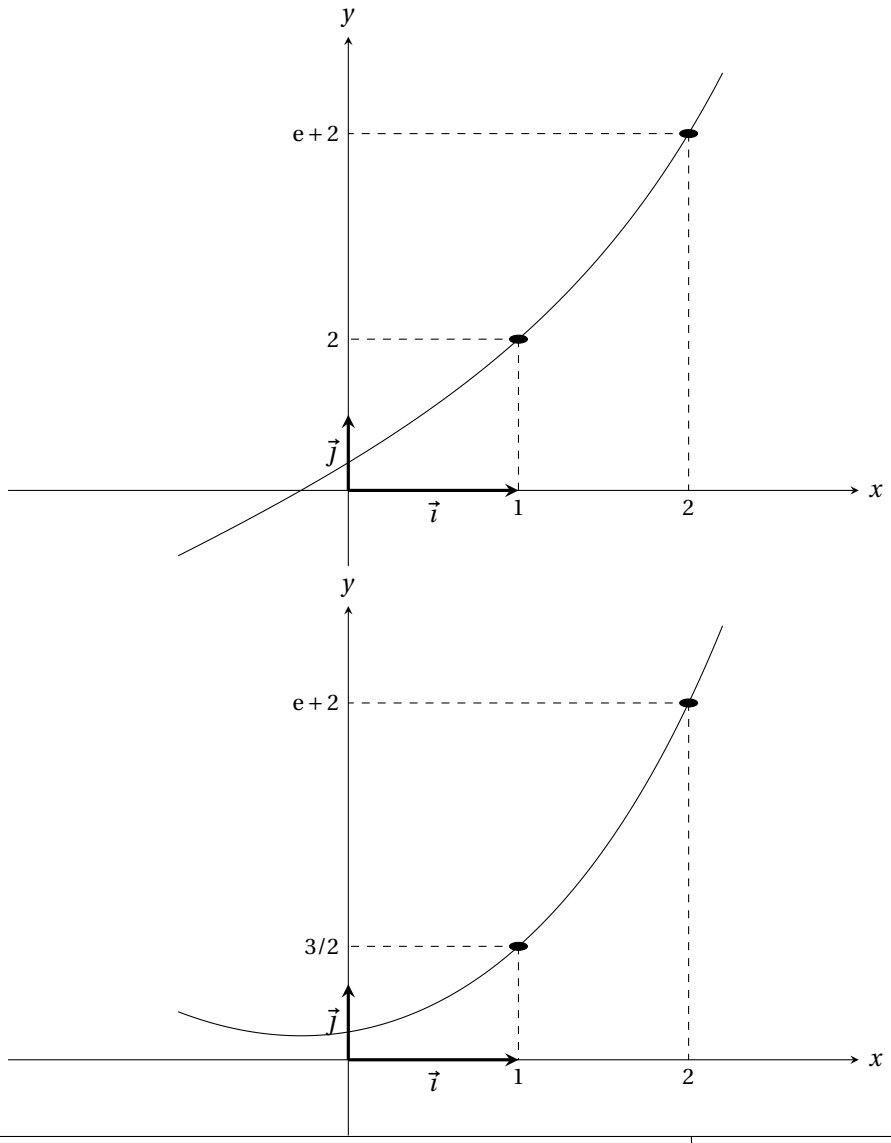
18.8.2 Van der Waals curves (continued)

The \foreach loop replaces the four definitions; the loop variable is used in the expression.




```
\begin{tikzpicture}[xscale=4,yscale=1.5]
  \tkzFnInit[xmin=0,xmax=3,ymin=-4,ymax=3]
  \tkzFnGrid(0,-4)(3,3)
  \tkzFnAxes
  \tkzFnVLine[color=red,dashed]{1/3}
  \foreach \a/\c in {0.5/red,0.84375/blue,1/orange,0.875/green}
    {\tkzFnDraw[color=\c,domain=0.35:3]{-3/(x*x)+8*\a/(3*x-1)}}
  \tkzFnText[draw,fill=white,text=Maroon](2,-2)
   {$f(x)=-\dfrac{3}{x^2}+\dfrac{8\alpha}{3x-1}$ \hspace{.5cm}
   with $\alpha \in
   \left\{\dfrac{1}{2}, \dfrac{27}{32}, \dfrac{7}{8}, 1\right\}$}
\end{tikzpicture}
```

19 Examples with the packages alterqcm and tkz-tab

Questions	Answers
Figure 1 shows the graph of a function f defined on $[0,+\infty)$ and figure 2 the graph of an antiderivative of f on $[0,+\infty)$. 	
1. What is the area, in square units, of the region bounded by the graph of f, the x-axis and the lines $x = 1$ and $x = 2$?	☐ $\mathrm{e} + \frac{3}{4}$ ☐ $\mathrm{e} + \frac{1}{2}$ ☐ 1

Questions	Answers
The function k, defined and positive on $[0,+\infty)$, is given by its variation table.	

x	0		1		3	$+\infty$
$k(x)$		↗		↘		↗ $+\infty$

2. Which of the following tables is the variation table of the function g defined on $[0,+\infty)$ by $g(x)=\dfrac{1}{k(x)}$?	☐ Table A ☐ Table B ☐ Table C

Table A

x	0		1		3	$+\infty$
$g(x)$		↗		↘		↗ $+\infty$

Table B

x	0		1		3	$+\infty$
$g(x)$		↘		↗		↘ $-\infty$

Table C

x	0		1		3	$+\infty$
$g(x)$		↗		↘		↗ 0

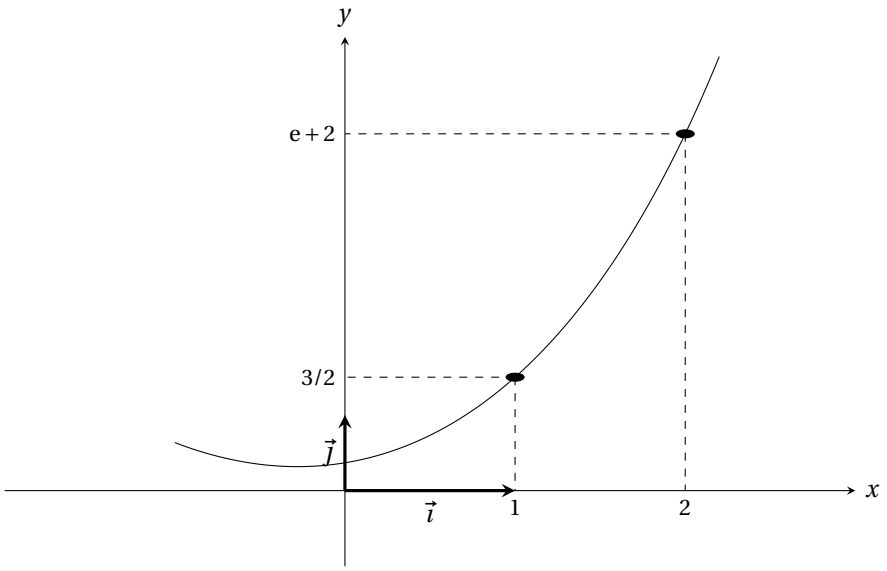
3. Let h be the function defined on $\mathbf{R}$ by $h(x)=\mathrm{e}^x-x+1$, and $\mathscr{C}$ its graph in an orthonormal coordinate system $(O;\vec{\imath},\vec{\jmath})$.	☐ The line $y=1$ is an asymptote of $\mathscr{C}$ ☐ The line $x=0$ is an asymptote of $\mathscr{C}$ ☐ The line $y=-x+1$ is an asymptote of $\mathscr{C}$
4. In economics, the marginal cost is the cost of producing one more unit, and it is taken to be the derivative of the total cost. In a company, a study shows that the marginal cost $C_m(q)$, in thousands of dollars, as a function of the number q of items produced is: $C_m(q)=3q^2-10q+\dfrac{2}{q}+20.$	☐ $C_r(q)=q^3-5q^2+2\ln q+20q+9984$ ☐ $C_r(q)=q^3-5q^2+2\ln q+20q-6$ ☐ $C_r(q)=6q-10-\dfrac{2}{q^2}$

Here is the code of the graphs of f and of its antiderivative:

19.0.1 First graph

```
\begin{tikzpicture}[xscale=2.25,yscale=1]
  \tkzFnInit[xmin=-2,xmax=3,ymin=-1,ymax=6]
  \tkzFnAxes[ticks=false]
  \tkzFnDraw[samples=100,domain=-1:2.2]{x+exp(x-1)}
  \tkzFnDefPointXY[draw](1,2){pt1}
  \tkzFnShowCoord[xlabel=$1$,ylabel=$2$](pt1)
  \tkzFnDefPointXY[draw](2,2+e){pt2}
  \tkzFnShowCoord[xlabel=$2$,ylabel=$\text{e}+2$](pt2)
  \tkzFnRep
\end{tikzpicture}
```

19.0.2 Second graph




```
\begin{tikzpicture}[xscale=2.25,yscale=1]
  \tkzFnInit[xmin=-2,xmax=3,ymin=-1,ymax=6]
  \tkzFnAxes[ticks=false]
  \tkzFnDraw[samples=100,domain=-1:2.2]{x*x/2+exp(x-1)}
  \tkzFnDefPointXY[draw](1,1.5){pt1}
  \tkzFnShowCoord[xlabel=$1$,ylabel=$3/2$](pt1)
  \tkzFnDefPointXY[draw](2,2+e){pt2}
  \tkzFnShowCoord[xlabel=$2$,ylabel=$\text{e}+2$](pt2)
  \tkzFnRep
\end{tikzpicture}
```

The answer to the first question can be checked with Lua: $\int_1^2 f(x)\,\mathrm{d}x \approx 3.2183$ and $\mathrm{e} + \frac{1}{2} \approx 3.2183$.

Code of a variation table:

```
\begin{tikzpicture}
     \tkzTabInit[lgt=1,espcl=2]{$x$/0.5,$k(x)$/1.5}
     {$0$,$1$,$3$,$+\infty$}
     \tkzTabVar{-/            /,%
                +/            /,%
                -/            /,%
                +/ $+\infty$  /}%
\end{tikzpicture}
```

20 From tkz-fct to tkz-fn

20.1 Correspondence of the macros

tkz-fct / tkz-base	tkz-fn
\tkzInit	\tkzFnInit
\tkzGrid[sub,subxstep,subystep]	\tkzFnGrid[sub,subdx,subdy]
\tkzAxeXY, \tkzAxeX, \tkzAxeY	\tkzFnAxes, \tkzFnAxisX, \tkzFnAxisY
\tkzDrawX, \tkzDrawXY	\tkzFnAxisX[ticks=false], \tkzFnAxes[ticks=false]
\tkzAxeX[trig=2]	\tkzFnAxisX[trig=2]
\tkzClip, \tkzRep, \tkzText	\tkzFnClip, \tkzFnRep, \tkzFnText
\tkzFct{x**2}	\tkzFnDraw{x^2} or \tkzFnDraw[name=f]{x^2}
\tkzDefPointByFct(2)	\tkzFnDefPoint(f,2){A}
\tkzDefPoint(1,2){A}	\tkzFnDefPointXY(1,2){A}
\tkzDrawPoint, \tkzLabelPoint	\tkzFnDrawPoints, \tkzFnLabelPoint
\tkzSetUpPoint	style tkzfn point
\tkzPointShowCoord	\tkzFnShowCoord
\tkzDrawTangentLine[with=b](1)	\tkzFnTangent(g,1)
\tkzDrawArea[domain=1:2]	\tkzFnArea[domain=1:2](f)
\tkzDrawAreafg[between=a and b]	\tkzFnArea(f,g)
\tkzDrawRiemannSum, ...Inf, ...Sup	\tkzFnRiemann[method=left](f), inf, sup
\tkzDrawRiemannSumMid	\tkzFnRiemann[method=mid](f) (see below)
\tkzFctPar, \tkzFctPolar	\tkzFnParam, \tkzFnPolar
\tkzVLine(s), \tkzHLine(s)	\tkzFnVLine(s), \tkzFnHLine(s)
\tkzDefSetOfPoints, \tkzDrawSetOfPoints	\tkzFnDrawData
\tkzXH	no equivalent yet

20.2 What changes

1. Expressions are Lua expressions: x^2 instead of x**2, ln(x) or log(x) for the natural logarithm, 1/3 is a decimal quotient.
2. The variable is always x, even when **xstep** is not 1: \x is no longer used.
3. A function is given by its name; the option **with** and the references a, b, … are gone. There is no longer a limit of 26 functions.
4. \tkzFctLast and \tkzFctgnua no longer exist: to place points or tangent lines without drawing the curve, define the function with \tkzFnDef.
5. Horner form is no longer needed and powers of negative numbers are correct.
6. The lengths **kl** and **kr** of the half tangent lines are always given in units of the x-axis; the style of the tangent lines (**tkzfn tangent**) is empty by default, whereas **tkz-fct** drew arrows.
7. With **method=mid**, the height of a rectangle is the value at the midpoint of the interval; with **tkz-fct**, it was the mean of the values at the endpoints.
8. Curves are clipped to the window and interrupted at vertical asymptotes.
9. Numbers are written in the American or the French format, without **numprint**.
10. New presentations of the axes: arrows at both ends (**arrows=both**) and "box" axes (**box**).

21 List of the macros

- **\tkzFnSetup**{⟨format=us or fr,decimals=2,axes=school or box,arrows=end⟩}
- **\tkzFnInit**[⟨xmin=-5,xmax=5,xstep=1,ymin=-5,ymax=5,ystep=1,enlarge⟩]
- **\tkzFnGrid**[⟨dx,dy,sub,subdx,subdy⟩](⟨x_1,y_1⟩)(⟨x_2,y_2⟩)
- **\tkzFnAxes**[⟨dx,dy,xlabel,ylabel,ticks,labels,line,origin,trig,decimals,arrows,box,tick length⟩]
- **\tkzFnAxisX**[⟨label,label style,step,trig,ticks,labels,line,decimals,arrows⟩]
- **\tkzFnAxisY**[⟨label,label style,step,ticks,labels,line,decimals,arrows⟩]
- **\tkzFnClip**[⟨margin⟩]
- **\tkzFnRep**[⟨options⟩]
- **\tkzFnText**[⟨options⟩](⟨x,y⟩){⟨text⟩}
- **\tkzFnDef**{⟨name⟩}{⟨expression⟩}
- **\tkzFnDraw**[⟨domain,samples=200,name,clip=true,mark,smooth⟩](⟨name⟩) or {⟨expression⟩}
- **\tkzFnParam**[⟨domain=0:2*pi,samples=200,clip=false⟩]{⟨$x(t)$⟩}{⟨$y(t)$⟩}
- **\tkzFnPolar**[⟨domain=0:2*pi,samples=200,clip=false⟩]{⟨$\rho(t)$⟩}
- **\tkzFnImplicit**[⟨level=0,samples=100,domain,ydomain,refine=true,clip=true⟩]{⟨$F(x,y)$⟩}
- **\tkzFnDefPoint**[⟨draw,size⟩](⟨f,x⟩){⟨name⟩}
- **\tkzFnDefPointXY**[⟨draw,size⟩](⟨x,y⟩){⟨name⟩}
- **\tkzFnDrawPoints**[⟨size⟩](⟨A,B,…⟩)
- **\tkzFnMarkPoints**[⟨size⟩](⟨f⟩){⟨list⟩}
- **\tkzFnLabelPoint**[⟨options⟩](⟨A⟩){⟨text⟩}, **\tkzFnLabelPoints**[⟨options⟩](⟨A,B,…⟩)
- **\tkzFnShowCoord**[⟨xlabel,ylabel,decimals⟩](⟨A⟩)
- **\tkzFnTangent**[⟨kl=1,kr=1,draw,clip=true⟩](⟨f,a⟩)
- **\tkzFnTangent**[⟨at=zeros|extrema,domain,samples=400,...⟩](⟨f⟩)
- **\tkzFnZeros**[⟨at=zeros,domain,samples=400,name=z,draw⟩](⟨f⟩)
- **\tkzFnArea**[⟨domain,where=both,samples,clip=true⟩](⟨f⟩) or (⟨f,g⟩)
- **\tkzFnRiemann**[⟨method=left,interval,number=10,clip=false⟩](⟨f⟩)
- **\tkzFnVLines**[⟨options⟩]{⟨list⟩}, **\tkzFnHLines**[⟨options⟩]{⟨list⟩}
- **\tkzFnDrawData**[⟨options⟩]{⟨x_1/y_1,…⟩}
- **\tkzFnPrint**[⟨decimals⟩]{⟨expression⟩}
- **\tkzFnPrintCoord**[⟨decimals⟩](⟨A⟩)
- **\tkzFnPrintInterval**[⟨oo, cc, oc, co⟩]{⟨a⟩}{⟨b⟩}

21.1 Styles

style	default
tkzfn curve	
tkzfn tangent	
tkzfn point	draw=black,fill=black
tkzfn area	fill=lightgray,fill opacity=.5
tkzfn riemann	draw,fill=gray!20
tkzfn line	
tkzfn grid	help lines
tkzfn subgrid	very thin,opacity=.5
tkzfn axis	->,>=stealth
tkzfn ticks	thin
tkzfn tick label	font=\small
tkzfn show coord	dashed,thin
tkzfn rep	->,>=stealth,very thick
tkzfn box	
tkzfn text	

Index

\clip, 9

\foreach, 22, 32, 51, 68, 70, 88

\tkzAxeXY, 8
\tkzClip, 8
\tkzDefPoint, 20
\tkzDefSetOfPoints, 80
\tkzDrawSetOfPoints, 80
\tkzDrawX, 8, 9
\tkzFctgnua, 92
\tkzFctLast, 22, 92
\tkzFnArea, 15, 39, 93
\tkzFnArea: options
    clip, 39
    domain, 39
    samples, 39
    where, 39
\tkzFnAxes, 9, 10, 93
\tkzFnAxes: options
    arrows, 9
    box, 9
    decimals, 9
    dx, dy, 9
    labels, 9
    line, 9
    origin, 9
    style, 9
    tick length, 9
    ticks, 9
    trig, 9
    xlabel, ylabel, 9
\tkzFnAxisX, 9, 10, 93
\tkzFnAxisY, 9, 10, 93
\tkzFnClip, 9, 93
\tkzFnDef, 7, 15, 33, 73, 92, 93
\tkzFnDef: arguments
    expression, 15
    name, 15
\tkzFnDefPoint, 20, 75, 93
\tkzFnDefPoint: arguments
    function,x, 20
    name of the point, 20
\tkzFnDefPoint: options
    draw, 20
    size, 20
\tkzFnDefPointXY, 20, 75, 93
\tkzFnDraw, 12, 15, 93
\tkzFnDraw: options
    clip, 15
    domain, 15
    mark=…, 15

name, 15
samples, 15
smooth, 15
\tkzFnDrawData, 80, 93
\tkzFnDrawPoints, 20, 93
\tkzFnGrid, 8, 93
\tkzFnGrid: options
dx, 8
dy, 8
subdx, 8
subdy, 8
sub, 8
\tkzFnHLine, 52
\tkzFnHLines, 52, 93
\tkzFnImplicit, 68, 93
\tkzFnImplicit: options
clip, 68
domain, 68
level, 68
refine, 68
samples, 68
ydomain, 68
\tkzFnInit, 6, 8, 12, 15, 68, 93
\tkzFnInit: options
enlarge x, enlarge y, 8
enlarge, 8
xmax, 8
xmin, 8
xorigin, 8
xstep, 8
ymax, 8
ymin, 8
yorigin, 8
ystep, 8
\tkzFnLabelPoint, 20, 93
\tkzFnLabelPoints, 20, 93
\tkzFnMarkPoints, 22, 93
\tkzFnNbZeros, 36
\tkzFnParam, 56, 93
\tkzFnParam: arguments
x(t),y(t), 56
\tkzFnParam: options
clip, 56
domain, 56
samples, 56
\tkzFnPolar, 63, 93
\tkzFnPolar: options
clip, 63
domain, 63
samples, 63
\tkzFnPrint, 7, 73, 93
\tkzFnPrintCoord, 24, 25, 36, 75, 93
\tkzFnPrintInterval, 73, 93

\tkzFnRep, 9, 93
\tkzFnRiemann, 48, 93
\tkzFnRiemann: options
    clip, 48
    interval, 48
    method, 48
    number, 48
\tkzFnSetup, 73, 93
\tkzFnSetup: options
    arrows, 73
    axes, 73
    decimals, 73
    format, 73
\tkzFnShowCoord, 24, 93
\tkzFnTangent, 15, 27, 36, 37, 93
\tkzFnTangent: arguments
    function,a, 27
    function, 27
\tkzFnTangent: options
    at, 27
    clip, 27
    draw, 27
    kl, 27
    kr, 27
\tkzFnText, 10, 26, 93
\tkzFnVLine, 51
\tkzFnVLines, 51, 93
\tkzFnVLines: arguments
    list, 51
\tkzFnZeros, 36, 37, 93
\tkzFnZeros: options
    at, 36
    domain, 36
    draw, 36
    name, 36
    samples, 36
    size, 36
\tkzGrid, 8
\tkzInit, 8
\tkzRep, 8
\tkzSetUpPoint, 20
\tkzText, 8