λ Pic

.

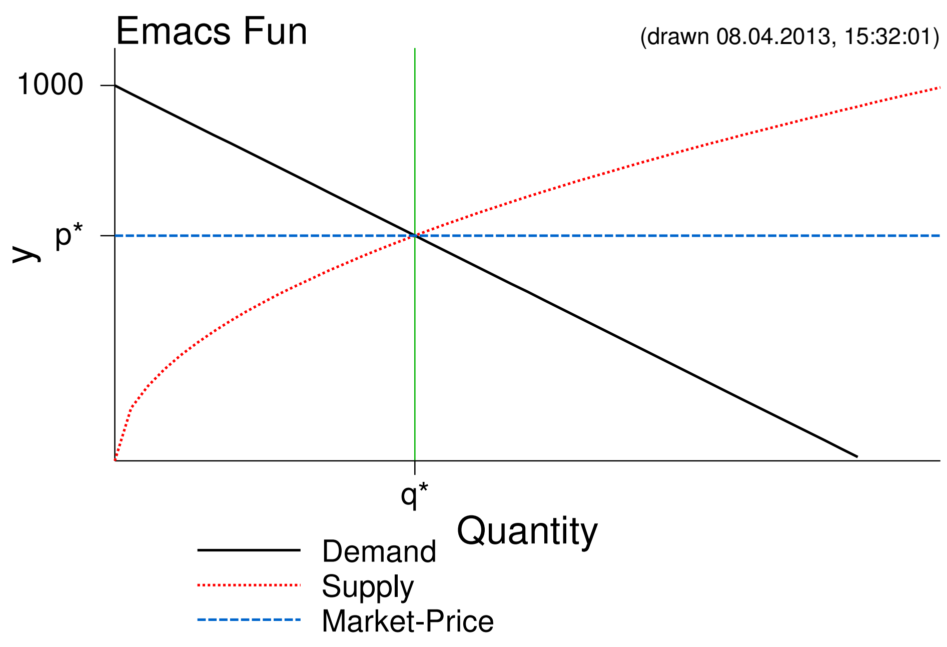


Table of Contents

1 Overview

For some current notes I needed a simple possibility to insert diagrams of $\mathbb{R} \mapsto \mathbb{R}$ functions. And I needed it fast[1]. For availability reasons, and because it matched my needs, I choose Ploticus[2] and for simplicity reasons Emacs Lisp[3]. Consequently text based, the pictures are embedded as scalable vector graphics (SVG). If you don't know Emacs Lisp or even Lisp at all, this may not be easy going. No knowledge of Ploticus is required.

As my experience with Emacs Lisp is small, be aware that the code may be brittle, suboptimal, and - of cource - changing.

2 Example

A diagram is entered into the org file like this

```
#+BEGIN_SRC emacs-fun
((xmax . 10) (ymax . 100) (title . "First Picture")) ;(ref:configuratio

(Square (x) (* x x))     ;(ref:Square)
(Cube   (x) (* x x x))
#+END_SRC
```

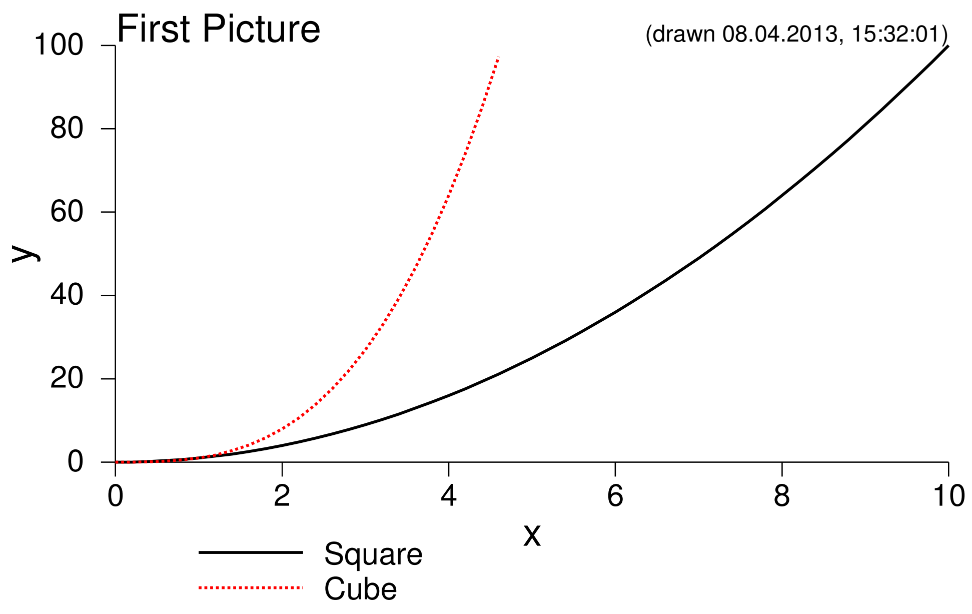
The Lisp code will be shown in the export almost verbatim (i.e. exept the references) as below.

Functions, like Square, that is $f(x) = x^2$, are given in a lispish way.

If there need to be some configurations, the must go into the first form (that can span several lines, btw.).

```
((xmax . 10) (ymin . 0) (ymax . 100) (title . "First Picture"))

(Square (x) (* x x))
(Cube   (x) (* x x x))
```



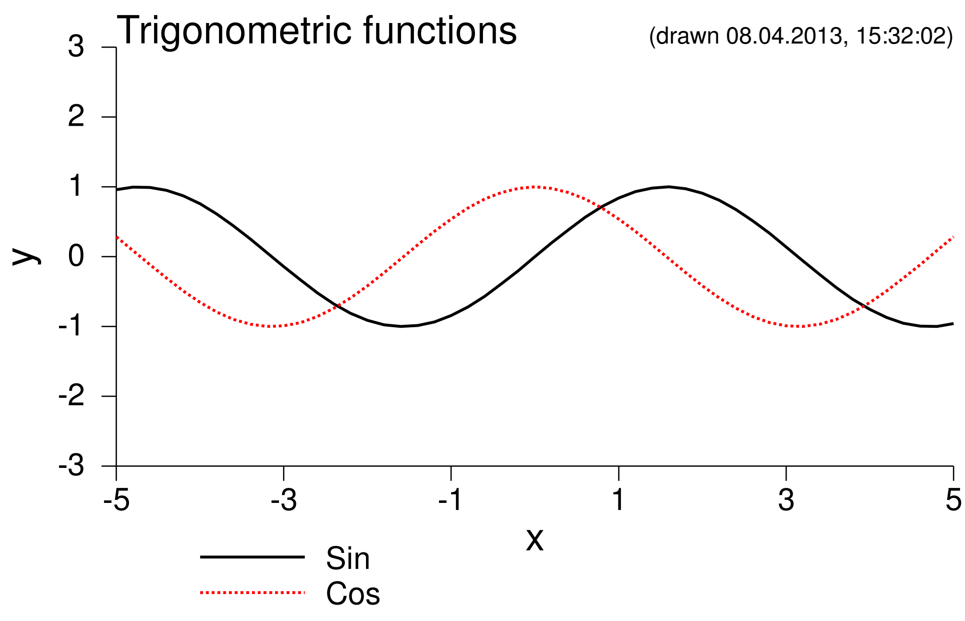
In case you are new to org mode: To generate HTML from your (or this) org file, press C-c C-e b (The HTML will be opened in the browser immediately. More commands here.)

By default the images get a filename constructed from buffer-name with a running number (%buffer-name%__figure_%n%.svg).

The name of links into code-lines, btw., are given with parentheses, that is a link coded like ;(ref:hello) is referred to as [[(hello)]]. (You have to omit the spaces, of course.)

```
((xmin . -5) (xmax . 5) (ymin . -3) (ymax . 3) (title . "Trigonomet

(Sin (x) (sin x))
(Cos (x) (cos x))
```

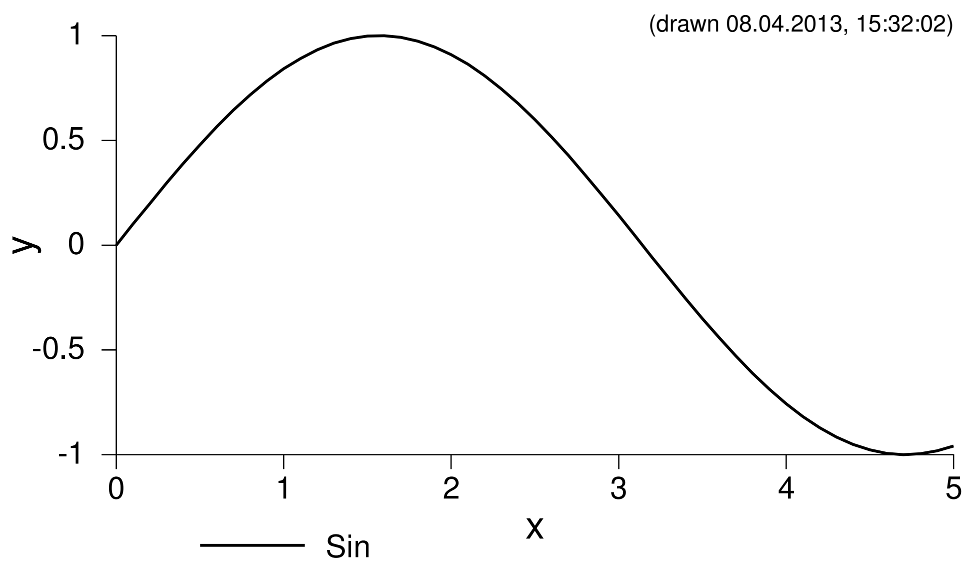


In case the configuration part of several images is similar, it can be put into the defaults in an emacs-lisp block. :export results is needed for the block to be executed. (The print statement is to inhibit printing.)

```
(emacs-fun-defaults '((xmin . 0) (xmax . 5)
                      (ymin . -1) (ymax . 1)
                      (xlabel . "x") (ylabel . "y")
                      (title . "")))
```

Now we can be short:

```
(Sin (x) (sin x))
```



Be aware that side effects of emacs-lisp blocks may change the Emacs VM. So these defaults settings are now in effect for the complete Emacs session.

2.1 Example

The top picture on this page uses some additional features, and its code is given below. (If something is confusing here, the code here and the picture code above may have gotten out of synch. I'm sorry, please look into the source.)

- Stubs: To give hand made labels at the axis, an association list called xstubs or ystubs can be used.
- Vertical lines are no functions. The current way to draw them is a bit contrived and not very flexible: we have to give xstubs, and every xstub generates a vertical line.
- The size of the picture can be given like this: (width . 700) (height . 400)

```
#+BEGIN_SRC emacs-fun :exports results
((title . "Emacs Fun")
 (xmax . 1100) (ymax . 1100) (xlabel . "Quantity")
 (ystubs . ((1000 . "1000") (600 . "p*")))
 (xstubs . ((400 . "q*")))
 (width . 700) (height . 400))

(Demand (x) (- 1000 x))
(Supply (x) (* 30 (sqrt x)))
(Market-Price (x) 600)
#+END_SRC
```

3 Parameters

Name	Default	Description
title	"Picture"	Title shown in the picture.
xmin	0	Minimal value shown at x-Axis
xmax	500	Maximal value shown at x-Axis
ymin	0	Minimal value shown at y-Axis
ymax	500	Maximal value shown at y-Axis
xlabel	"x"	Label of x-Axis
ylabel	"y"	Label of y-Axis
width	600	With of the embedded picture
height	400	Height of the embedded picture
xstubs	nil	Stubs, see Example above
ystubs	nil	Stubs, see Example above

4 Installation

- Get Emacs 24[3].
- Get Ploticus[2]
- Put ob-emacs-fun.el into the Lisp directory of org mode. A usual place is the lisp/org subdirectory of your Emacs installation directory. You should already see files like ob-emacs-lisp.el there.
- Adjust your emacs configuration to enable Emacs Fun, as described below.

4.1 Emacs Configuration

I have the following sniplets in my .emacs to get this running. In order of occurrence.

- Usage of $\TeX$ via MathJax:

```
(setq org-export-html-mathjax-options
      '((path "http://cdn.mathjax.org/mathjax/latest/MathJax.js?conf
        (scale "100")
        (align "left")
        (indent "2em")
        (mathml t)))
```

- Style and MathJax

  (See TODO, especially the line numbering doesn't work.)

```
(setq org-export-html-preamble
  "<link rel='stylesheet' type='text/css' href='org.css' />
   <script type='text/x-mathjax-config'>
     MathJax.Hub.Config({TeX: { equationNumbers: {autoNumber: 'all'
   </script>")
```

- Allow

  This allows emacs-fun and emacs-lisp blocks to ALWAYS be executed without asking. So take care if you have files from people you should not trust. Emacs is no sandbox.

```
(defun my-org-confirm-babel-evaluate (lang body)
  (not (or (string= lang "emacs-fun")
           (string= lang "emacs-lisp"))))
(setq org-confirm-babel-evaluate 'my-org-confirm-babel-evaluate)
```

- Load

```
(require 'ob-emacs-fun)
(org-babel-do-load-languages
 'org-babel-load-languages
 '((emacs-fun . t)))
```

5 Todo

5.1 Should

- Rename short named variables, like default. They must get names with lower probability of interference.
- Is there a more idiomatic way to set the style for org mode HTML export?
- MathJax equation numbers don't work.

5.2 Maybe

- Usage of org-mode postamble might be enhanced (Currently, org.css makes it invisible.):
  - can author be set from .emacs
  - time stamp language shall be English
- The formula (in $\TeX$) might be generated from code too. In that case, if a function gets a ;(ref:..) - the (latex-)formula should get the same marker. (Hm. A line doesn't generally correspond to a function.)

6 Issues

- Among the strengts of babel is the possibility to pass data from block to block. I don't do this here, thus abandoning the possibility. How could we use it?
- I'm somehow not comfortable with all exported files being generated into my org directory.

7 Open Questions

- I want to know the number of the code block (blocks enumerated as sequence). Currently I use a counter and an advise to org-export.
- Do I have access to the name of the code block, if there is any?

8 Feedback

Please use the Github issue tracking system.

Footnotes:

[1] As you might suspect from the example picture, it was for an economics lecture. I highly recommend Principles of Economics for Scientists by Antonio Rangel, a compact one semester introduction into micro economics.

[2] http://ploticus.sourceforge.net/ Just install it. Ploticus must be in the path.

[3] I tested in Emacs 24.1.1 and Emacs 24.2