Tip for exporting Maxima results to LaTeX

Leo Butler

February 7, 2023

Goal

Generate LaTeX code from Maxima code.

Setup

maxima-init.lisp

The command `org-babel-execute:maxima` in `lisp/ob-maxima.el` uses the Maxima command `batchload` to execute Maxima code. This is a very tight-lipped loader, so we over-write `batchload` with `batch`. We also `load` an init file:

```
#+begin_src maxima :tangle maxima-init.lisp :exports none
  (defun $batchload (file) (mfuncall '$batch file))
  ($load "./maxima-init.mac")
#+end_src
```

On tangling, this produces the `common-lisp` output file `maxima-init.lisp`. It will be pre-loaded into Maxima.

maxima-init.mac

Next, we need to create an init file for Maxima that will provide an output printer that produces LaTeX output. One option would be to use the `imaxima` printer. Here is another option that uses the `alt-display` package. The code replaces the default printer with `org_tex_display`. It also sets the `epilog` prompt, so that the final `#+begin_example` is terminated.

```
#+begin_src maxima :tangle maxima-init.mac :exports none
  load("alt-display.mac") $
  set_prompt('epilog,printf(false,"~%#+end_example")) $
  define_alt_display(org_tex_display(x),
    block([], printf(true,"#+end_example~%#+begin_export latex~%"),
      printf(true,"\\textcolor{blue}{(\\~a~d)} ",outchar,linenum-
1), tex(second(x)), printf(true,"~&#+end_export~%#+begin_example~%(input) "))) $
  set_alt_display(2,org_tex_display) $
  display2d:true $
  printf(true,"#+begin_example~%(input) ") $
  linenum : 0 $
#+end_src
```

An example

Here is an example that computes the derivative of a composite function.

```
(input)
read and interpret /tmp/babel-hhTrJS/maxima-0m0DnH.max
(gradef(f(u,v),f_1(u,v),f_2(u,v)),'done)
```

(%o1)

$$done$$

```
(input)
diff(f(x^2-y^2,x*y),x)
```

(%o2)

$$y\, f_2\left(x^2 - y^2, x\, y\right) + 2\, x\, f_1\left(x^2 - y^2, x\, y\right)$$

```
(input)
diff(f(x^2-y^2,x*y),y)
```

(%o3)

$$x\, f_2\left(x^2 - y^2, x\, y\right) - 2\, y\, f_1\left(x^2 - y^2, x\, y\right)$$

```
(input)
gnuplot_close()
```

Two annoyances

The initial line `read and interpret...` and that final, dangling input line
with `gnuplot_close()` are nuisances. They can be easily suppressed, but
that requires patching `ob-maxima.el`. That's another story.