Contents

1 TODO Equation image tooltips over LaTeX fragments in org-mode

This post has one goal: to fontify LaTeX fragments in org-mode. I previously looked at doing something like this with the cursor, but now I want to try with the mouse. I want the highlighting so they stand out more clearly from text, and it would be nice to make them clickable to toggle the image.

The variable org-latex-regexps contains a list of lists which seem to contain regular expressions for latex fragments. The inline LaTeX doesn't seem to be covered there. These don't seem to be affected by the preview code, but it is still nice to see them in the text. Rasmus pointed out that the inline fragments can contain all kinds of things that aren't even previewable!

'org-latex-regexps' contains regexp patterns We will use those to build a function that matches fragments, and puts some text properties on the match to make them clickable.

Given a fragment, we can get a preview like this.

```
(let ((f (make-temp-file "ltx" nil ".png")))
  (org-create-formula-image "$e^x$"
                            f
                            nil (current-buffer)
                            org-latex-create-formula-image-program)
 f)
```

What we need to do then, is make a function for font-lock that will put a tool tip on our fragments, and generate the image for us on the fly. I am going to set a property on the text during font-lock, and we will use that property in the tool-tip. So, first the tooltip function.

```
(defun latex-fragment-tooltip (object window position)
  (save-excursion
    (goto-char position)
    (let* ((f (concat temporary-file-directory
                      (md5 (get-text-property (point) 'latex-fragment))
                      ".png"))
           (png))
      (unless (file-exists-p f)
        (org-create-formula-image
         (get-text-property (point) 'latex-fragment)
         f
         '(:foreground default :background default
                      :scale 4.0 :html-foreground "Black"
                      :html-background "Transparent" :html-scale 4.0
                      :matchers
                      ("begin" "$1" "$" "$$" "\\(" "\\["))
         (current-buffer)
         org-latex-create-formula-image-program))
      (message (propertize
                " " 'display
                (create-image f))))))
```

latex-fragment-tooltip

Now, we need to propertize the fragments.

```
(defun propertize-fragment ()
  (put-text-property
   (match-beginning 0) (match-end 0)
   'latex-fragment (match-string-no-properties 0))
  (put-text-property
   (match-beginning 0) (match-end 0)
   'help-echo 'latex-fragment-tooltip))

(defun next-latex-overlay-1 (limit)
  "Overlay images on \(eqn\) up to LIMIT."
  (while (re-search-forward "\\\\([^]*?\\\\)" limit t)
    (propertize-fragment)))

(defun next-latex-overlay-2 (limit)
  "Overlay images on \[eqn\] up to LIMIT."
  (while (re-search-forward "\\\\\\[[^]*?\\\\\\]" limit t)
    (propertize-fragment)))

(defun next-latex-overlay-3 (limit)
  "Overlay images on $$eqn$$ up to LIMIT."
  (while (re-search-forward "\\$\\$[^]*?\\$\\$" limit t)
    (propertize-fragment)))

(defun next-latex-overlay-4 (limit)
  "Overlay images on $eqn$ up to LIMIT.
this is less robust than useing \(\)"
  (while (re-search-forward "\\([^$]\\|^\\)\\(\\(\\$\\([^
\n,;.$][^$\n
]*?\\(\n[^$\n
]*?\\)\\{0,2\\}[^
\n,.$]\\)\\$\\)\\)\\([-        .,?;:'\")]\\|$\\)" limit t)
    (propertize-fragment)))

(defun next-latex-overlay-5 (limit)
  "Overlay images on latex math environments up to LIMIT."
  (while (re-search-forward
          "^[ \\t]*\\(\\\\begin{\\([a-zA-Z0-9\\*]+\\)[^\\000]+?\\\\end{\\2}\\)"
          limit t)
    (propertize-fragment)))
```

Finally, add the font lock keywords.

```
(font-lock-add-keywords
 nil
 '((next-latex-overlay-1 0 font-lock-keywords)
   (next-latex-overlay-2 0 font-lock-keywords)
   (next-latex-overlay-3 0 font-lock-keywords)
   (next-latex-overlay-4 0 font-lock-keywords)
   (next-latex-overlay-5 0 font-lock-keywords)))
```

$$\int_0^x 5dx$$

$$\int_0^x 500dx$$

1. $\exp x$
2. $$\exp 2x$$
3. $\exp 3x$
4. $$\exp 4x$$