1 A little bit of perl

"Perl" stands for *practical extraction and report language.* You will find many books on the language in the bookstore, for it is very popular in applications to the internnet. Our use of perl will exclusive as an editing tool. All of the scripts we use either build a Maple worksheet, or a LaTeX file or massage a set of existing LaTeX files, that is, make substitutions, deletions and additions to the files. These are tasks that could be done by hand with a good text editor, but the time required to complete the task would be infinite because boredom, and/or confusion, would set in about halfway thru.

We truly don't need to know any perl to run the scripts, as we have seen, but if you want to be able to modify them it is necessary to know a little bit. A good way to learn a little bit is by looking at examples. Here are a couple small, but useful ones.

1.1 Global substitutions in multiple files

Suppose you have several files that contain the same mispelled word. This can be done file by file using a text editor. A good editor, like vi or pfe, or word97, will allow you to make global substitutions in a file. For example, in vi (which is on the chisel disk) the following sequence in a TeXShell would change each occurence of *thru* to *through* in the file bill.tex.

```
vi bill.tex

:1,$s/thru/through/g

:wq
```

On the other hand, here is a short perl script (called glosub.pl) which can make the same substitution, for example, in all tex files in the current directory by executing the sequence

```
perl -S  glosub  thru  through  *.tex
```

Just copy the script into an editor and save the file as glosub.pl in X:\chisel\bin, where X is the letter of the drive where you installed chisel.

1.1.1 glosub.pl

```
#!perl  -i.bak
  $opt = $ARGV[0];
  if  ($opt =~ /\-h/) {
  print "Usage:   perl -S glosub old new files
  \n";
  print "         This script replaces all occurences of old by new in
  files.\n";
  exit;
  } $old = $ARGV[0]; shift;
  $new = $ARGV[0]; shift;
  $/ = "";
  while(<>){
```

```
s/$old/$new/g;
print;
} print "done";
exit;
```

You can learn a little bit of perl just by reading through this script, and testing it out on some test files in a temporary directory. A lot you can pick up from context, but there are many things can only be learned by reading and/or seeing them demonstrated. For example, the top line activates the perl option -i, which says to modify the files in place, but make back up files first with names like filename.bak where filename is one of the files being processed. Similarly, the line $/ = ""; says to process the files a paragraph (separated by blank lines) at a time.

I mention here that there is a lot of documentation that comes with the perl5 on the chisel disk. After you have installed it, look in X:\Perl\doc, where X is the drive letter where you install Chisel.

1.2 Recolorizing the eps files that are exported from Maple

When you export a Maple worksheet to Latex, the plots that are in the worksheet are exported as monochrome plots. However, they can be recolorized by observing which shade of grey each of the sixteen named colors in Maple converts to and make the appropriate inverse substitution in the eps file. Here is a perl script that performs the substitution. Note that it also changes the thickness of lines in the plot and removes the border around the plot.

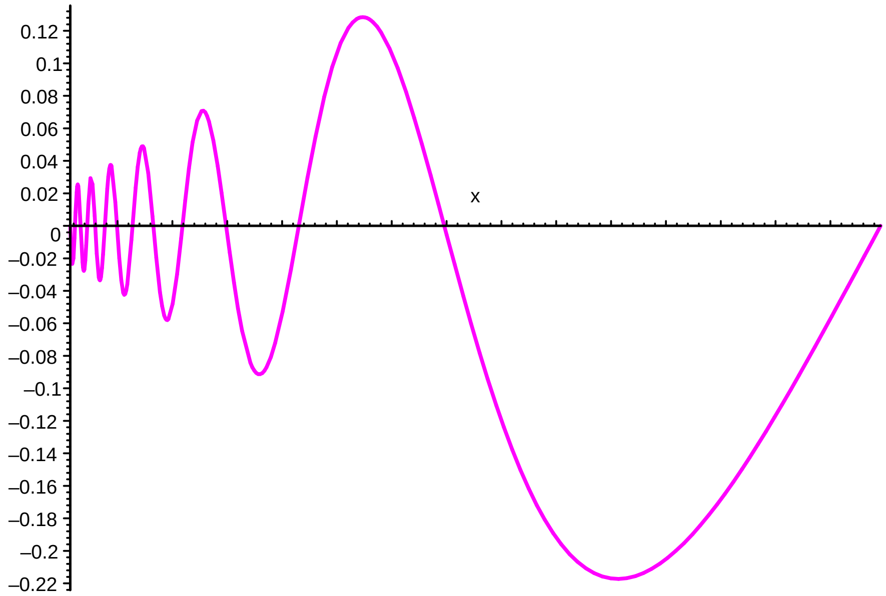
1.2.1 clreps.pl

```
#!perl  -i.bak
  $opt = $ARGV[0];
  $rel5 = "no";
  $comout = "off";
  if  ($opt =~ /\-h/) {
  print "Usage:   perl -S clreps files  to colorize black and
  white\n";
  print "         eps files generated when exporting Maple
  worksheets\n";
  print "         to LaTeX.  Also thickens up lines in line
  drawing\n";
  print "         and removes the border around the
  plot.\n";
  exit;
  } while(<>){
  if (/drawborder true def/){s/drawborder true def/drawborder false
  def/;
  $rel5 ="yes";} s/thin 3 def/thin 13 def/;
  s/medium 7 def/medium 17 def/;
  s/thick 16 def/thick 26 def/;
  s/0.646078 G/.64706 .16471 .10471 C/;   #tan, i think
```

```
s/0.701882 G/.43922 .85882 .57647  C/;  # aquamarine
s/0 G/0 0 0 C/;  # black
s/0.11 G/0 0 1 C/;  # blue
s/0.179922 G/.13725 .13725 .55686 C/;  # navy
s/0.593843 G/1 .498 0 C/;  # coral
s/0.7 G/0 1 1 C/;   # cyan
s/0.30549 G/.64706 .16471 .16471 C/; # brown
s/0.55498 G/.8000 .4980 .1961 C/; # gold
s/0.59 G/0 1 0 C/;  # green
s/0.752941 G/.75294 .75294 .75294 C/; # gray
s/0.592431 G/.623529 .623529 .372549 C/; # khaki
s/0.41 G/1 0 1 C/; # magenta
s/0.291451 G/.55686 .13725 .41961 C/; # maroon
s/0.37451 G/.80000000 .19607843 .19607843 C/; # orange
s/0.82949 G/.55686 .13725 .41961 C/; # pink
s/0.774196 G/.91764706 .67843137 .91764706 C/; #plum
s/0.3 G/1 0 0 C/; # red
s/0.42698 G/.55686275 .41960784 .13725490 C/; # sienna
s/0.844706 G/.67843137 .91764706 .91764706 C/; # turquiose
s/0.234157 G/.30980392 .18431373 .30980392 C/; # violet
s/0.832784 G/.84705882 .84705882 .74901961 C/; # wheat
s/ 1 G/1 1 1 C/; # white
s/0.89 G/1 1 0 C/; # yellow
#  This takes off the border for release 4 eps files
if  (/The following draws a box/ && $rel5 eq "no") {$comout =
"on";} if  ($comout eq "on") {s/./\%/; if (/end of
if/) {$comout = "off";}} print;
} exit;
```

We can test this by export this worksheet to latex and then in a TeXshell, issue the command perl -S clreps lit*.eps

```
plot(sin(1/x)*x,x=1/(14*Pi)..1/Pi,
numpoints=100,thickness=3,color=magenta);
```


0.12
0.1
0.08
0.06
0.04
0.02
0
–0.02
–0.04
–0.06
–0.08
–0.1
–0.12
–0.14
–0.16
–0.18
–0.2
–0.22
x