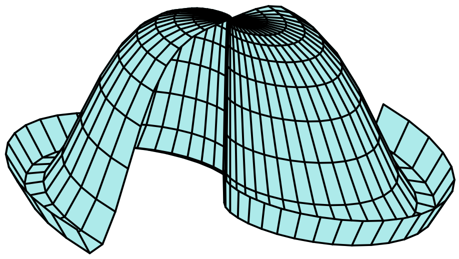
Making books with Maple

Carl Eberhart, carl@ms.uky.edu
Mathematics Department, University of Kentucky



June 9, 1998

Contents

1 Making books of worksheets

1.1 What is a book of worksheets?

MapleVr4 has three features not present in previous releases. One can have multiple worksheets open at one time in a session. Also, one can insert ***bookmarks*** to paragraphs in a worksheet and ***hyperlinks*** to (bookmarks in) worksheets. The combined effect of these features is that one can link together collections of worksheets into a book of worksheets, so that once one of the worksheets has been opened, one can successively open other worksheets linked to it with the hyperlinks in that worksheet. For example, this worksheet is in a small book of worksheets, another of which is perls.mws . Clicking on the hyperlink will open up the worksheet perls.mws. You can insert a hyperlink with the insert menu above.

There are many ways to links books of worksheets together. The simplest structure is to have a table of contents worksheet, which contains a possibly annotated list of hyperlinks to the worksheets in the book, together with a hyperlink back to the table of contents in each worksheet. This way one can get from any worksheet in the book to any other worksheet by way of hyperlinks. So one can read the book by just opening any worksheet in the book, which leads to the table of contents worksheet, which then leads to all the other worksheets.

This small book of worksheets has a table of contents called toc.mws . Usually, I add a link back to a table of contents at the bottom of each worksheet in the book.

Although a table of contents worksheet is easily constructed by hand, it would be handy to have a way to construct one outside of Maple if you want to try out various combinations of worksheets. You can use a perl5 script *mkcontents.pl which I wrote for this purpose.*

1.2 mkcontents.pl

This perl5 script constructs a MapleVr4 worksheet which contains a list of hyperlinks to the worksheets in a collection of worksheets.

Usage:

1. perl -S mkcontents -h *prints a short help message and exits.*

2. ***perl -S mkcontents tocname fileofwkshts***

where ***tocname is the name of the table of contents worksheet to be produced (eg toc.mws), and fileofwkshts is the file containing the names of the worksheets in the 'book', which are to appear as a hyperlink in tocname.***

In order to use this script, you will need to have perl5 installed on your machine. Perl5 comes on the Chisel disk.

Of course, additional hyperlinks can be inserted into the book as you read it and make notes to yourself. It is easy to add a worksheet to the book. Just put a link to the table of contents worksheet and then add a line to the table of contents containing a link the to new worksheet. This way you can ***grow your own book*** as the semester wears on.

1.3 Bookmarks

Now, how can we use the capability of making bookmarks in a worksheet? In any paragraph in the worksheet, one can have one bookmark. The way to insert a bookmark is to choose Edit in the Bookmark option of the View Menu (Just below Zoom). A separate window opens up and one can see what to do from there. I have inserted two bookmarks in this worksheet already. When you click on a bookmark, the cursor goes to the paragraph which contains that bookmark. So if you have a long worksheet, you can use the bookmarks to navigate around in the worksheet.

Bookmarks are useful to the reader of a book of worksheet in at least two ways. First if the writer has inserted any bookmarks the reader can use them to guide his reading. Second the reader can insert his own bookmarks to help him locate a particular paragraph in the worksheet later. Since the reader of a worksheet is often the writer of the worksheet (or one of them), the same remarks apply to him.

One feature of hyperlinks that we haven't mentioned is that the link can actually take the cursor to a paragraph marked by a bookmark in the worksheet. This enables one to use bookmarks in constructing an index worksheet for a book of worksheets. Such a worksheet would consist of an alphabetized list of all of the bookmarks in the book of worksheets. Each item in the list is equipped with a hyperlink to the paragraph in the worksheet marked by the bookmark. So when you click on an item in the index, the worksheet containing the bookmark is opened up and the cursor is placed at the top of the paragraph which contains the bookmark.

Index worksheets would be very cumbersome to construct by hand, althought it could be done. I have a perl5 script which accomplishes this task quickly.

1.4 mkindex.pl

This perl5 script produces a MapleVr4 worksheet which contains a sorted list of all bookmarks in a collection of worksheets. These bookmarks are linked to the appropriate worksheet, thus giving an active index. Also, the index worksheet contains a link to a 'table of contents' worksheet. This enables one to construct an index worksheet to a 'book of worksheets'.

Usage:

1. ***perl -S mkindex -h prints a short help message and exits.***

2. ***perl -S mkindex indexname tocname fileofwkshts***

where indexname is the name of the index worksheet to be produced (eg index.mws), and tocname and fileofwkshts are as in mkcontents.

For examples of complete books of worksheets, see Visual Problem Solving with Maple by Paul Eakin and Carl Eberhart, and the Problem Solving with Maple: A handbook for calculus, by Carl Eberhart. Both of these are on the Chisel Disk.

We think there is a lot of potential in the idea of a book of worksheets.

2 Making Latex Images of Worksheets

If you have a single worksheet, you can simply print it off from Maple using the print option in the file menu. This works nicely for class handouts, quizzes, exams, and short papers. However, if you have a book of worksheets with a table of contents and an index, or just a set of worksheets that you want to be an integrated document (pages numbered consecutively, etc), you can export the worksheets to LaTeX and work with these to get a LaTeX image of the book of worksheets.

This worksheet describes some perl5 scripts which I been developing to massage files exported from MapleV worksheets, before latexing them. In order to use them, you need to have perl5 installed on your machine. It is on the Chisel disk.

An overview The idea here is to make the *.mws worksheet files the source files, so that any changes by hand are made in the actual worksheet rather than in the latex file exported from Maple. The script mknotes.pl takes a collection of exported worksheets, modify each one in the manner described below, and constructs a rootfile which contains the formatting commands and then 'includes' the files into the rootfile.

2.1 mknotes.pl

Usage:

1. **perl -S mknotes -h** to get a help message and exit.

2. **perl -S mknotes rootname worksheetfile** , where rootname is a name (such as calcnotes.tex) you have chosen for the rootfile, and worksheetfile is the name of a file containing the names of the worksheets you have exported to latex. One name per line in the order you want them to be in the document.

3. **perl -S mknotes rootname worksheet.tex**, where rootname is as above and worksheet.tex is the name of a worksheet exported to latex.

4. **perl -S mknotes -t rootname worksheetfile infofile**, where rootname and worksheetfile are as in 2. and infofile is the name of a file containing three pieces of information separated by blank lines. Use mknotes -h to get the format details.

5. **The i option:** Change mknotes to mknotes -i in 2. or 3. and mknotes -t to mknotes -it in 4. to suppress the input cells in the LaTeX output. This is helpful when preparing papers or exams where you might not want the input cells to appear.

6. **The w option** is used if you want to wrap text around selected diagrams (only in mknotes54.pl)

Effects : The rootfile is opened and some appropriate latex2e lines are written to it. If the -t option has been chosen, the info file is opened and the TITLE, AUTHOR, and FRONTPIECE information is extracted and written into the rootfile (if you don't supply an infofile, dummy stuff is written in). Then the worksheet file is opened and the files listed there are processed one by one.

2.1.1 Things that mknotes.pl does

Fixes a bug in the formatting of mapleinput lines. The current edition of maple2e.sty does not respect embedded carriage returns in an input cell. When one is developing the definition of a maple word, it is natural to have long input cells with lots of embedded carriage returns. When this is exported to latex, the result is that input lines overflow badly. mknotes fixes this problem for the most part. (You can see the embedded carriage returns if you choose the "show invisible characters" option on the View Menu.)

```
tabular := proc (f ,obj, v )
local t, A, m,i, n, x,ob,c,Z, vars,zrow;
t := convert(map(lhs-rhs = 0,f),list);
if nargs = 2 then vars := sort(convert(indets(t,name),list))
else vars := v fi;
ob:=subs([seq(h=0,h=vars)],obj);
zrow := linalg[genmatrix]([obj=0],vars,'flag');
zrow[1,nops(vars)+1] := ob;
A := linalg['genmatrix'](t,vars,'flag');
m := linalg['rowdim'](A); n := linalg['coldim'](A);
A := linalg[stackmatrix](zrow,A);
A := linalg[stackmatrix]([op(vars),Z],A);
end:
```

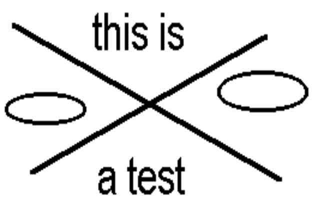


Figure 1: TEST1

Enables inclusion of externally drawn graphics files. If you have a bmp or eps file drawn with an external program pasted into your worksheet, and you want this to appear in the latex output, do the following: 1. Make sure a copy of the graphic file is in the current directory. 2. Underneath the pasted in picture in the worksheet, open a new paragraph and type in the name of the graphic file, including its suffix, followed by its dimensions, width first, in inches. Center this and change the paragraph style to diagram. (If you don't have such a style, you just create it from the Format Menu. Don't give it any properties, except possibly change the color to very light grey to reduce it's visibility.). When you export this, mknotes makes the substitutions needed, even if there is no eps file.

Handles user defined character style bookmark. Create a character style called Bookmark, with green font say. Then when you make a bookmark in your worksheet from the View menu, also highlight a word or phrase in that paragraph and convert it to the character style bookmark, leaving 4 to 8 blanks after the highlighted word. Then mknotes inserts an index marker there so that you can place the word in the index. This can be done even when you don't have a bookmark in the worksheet. After you latex the rootfile, a file with the same name as the rootfile, but with an idx suffix is created. Run makeindex on that file and another file with the same name but with suffix ind is created. You can change the name of that file to something like index.tex and add the line include{index} to the bottom of your rootfile, and relatex it. Voila! An index.

Sample: See above

Gets rid of leading hyperlinks. Hyperlinks which are in the worksheets and direct people to the menu page don't want to be in the latex output. If they are in a paragraph by themselves, mknotes takes them out.

Gets rid of QTR. QTR appears whenever you have a hyper link or any user defined character style. Use is made of this in locating bookmarks and leading hyperlinks. After that I have no further use for it, so I just get rid of it.

Makes LaTeX substitutions. Maple has several made several passive operations, for example, Int, Sum, and Diff are all operations which do not carry out the operation but generate the typesetting symbol for the operation. However, there is no passive union, intersection, or setminus in MapleVr4. Mknotes fixes this (sort of), and also adds some other latex substitutions (for ray, line, seg, angle, and triangle). Below are several examples, from which you can get the idea of what can be done. More work needs to be done on this portion of mknotes For example, you can add your own latex macros and modify mknotes to recognize these. (Note: In release 5, there is a character style, LaTeX, which can be used to pass raw LaTeX commands and macros.)

Here is an example of an inline substitution: $\boldsymbol{A}$ **union** ($\boldsymbol{B}$ **union** $\boldsymbol{C}$). the same thing won't work on displayed output.

```
A union (B union C);
```

$$\mathbf{union}(\boldsymbol{A},\ \boldsymbol{C},\ \boldsymbol{B})$$

What one can do make an ampersand definition or a proc definition.

```
`&union` := proc(A,B) options operator;
cat(`(`,A,` Union `,B,`)`) end;
```

$$\boldsymbol{\&union} := \mathbf{proc}(\boldsymbol{A},\ \boldsymbol{B})\ \mathbf{option}\ \boldsymbol{operator};\ \mathbf{cat}(`(`,\ \boldsymbol{A},\ `\ \cup`,\ \boldsymbol{B},\ `)`)\ \mathbf{end}$$

```
A &union ( B &union C );
```

$$(\boldsymbol{A}\ \cup (\boldsymbol{B}\ \cup \boldsymbol{C}))$$

Or you can just make a plain ol proc.

```
uni := proc(A,B) cat(`(`,A,` Union `,B,`)`)
end;
```

$$\boldsymbol{uni} := \mathbf{proc}(\boldsymbol{A},\ \boldsymbol{B})\ \mathbf{cat}(`(`,\ \boldsymbol{A},\ `\ \cup`,\ \boldsymbol{B},\ `)`)\ \mathbf{end}$$

```
uni(A,uni(B,C));
```

$$\mathbf{uni}(\boldsymbol{A},\ \mathbf{uni}(\boldsymbol{B},\ \boldsymbol{C}))$$

One can set up some things using '*'.

```
R*Intersect*uni(ray(C,D),ray(A,B));
```

$$\boldsymbol{R\ Intersect}\ \mathbf{uni}(\mathbf{ray}(\boldsymbol{C},\ \mathrm{D}),\ \mathbf{ray}(\boldsymbol{A},\ \boldsymbol{B}))$$

Here is some inline geometry stuff: $\boldsymbol{angle\ RVD = ray\ VR\ Union\ ray\ VD}$ and some more $\boldsymbol{A}$ **subset**($\boldsymbol{B}$ **minus** $\boldsymbol{C}$) also some more $\boldsymbol{triangle\ abc}$ **Union**(**segment**($\boldsymbol{FT}$) $\boldsymbol{Intersect}$ **line**($\boldsymbol{AT}$))

Here is $\boldsymbol{angle\ \alpha}$

```
angle*RVD = (ray*VR)* Union(ray*VD);
```

$$\boldsymbol{angle\ RVD = ray\ VR}\ \mathbf{Union}(\boldsymbol{ray\ VD})$$

Turns off input. The input used to generate the formatted output is suppressed, so you don't have to erase those lines from the worksheet.

Puts in newpages and vfills. Create two new paragraph styles newpage and vfill. Then where you want newpages or vfills, insert a blank paragraph in that style.

2.2 mkcolor.pl

This script act just like mknotes, except it adds color commands to the latex source. So, you can make color postscript files. Even if, you don't have a color printer the color adds a lot when you are using ghostview as a reader.

Unlike the three scripts mknotes.pl, mkindex.pl, and mkcontents.pl, mkcolor.pl does not come with the chisel disk. It comes in the zipfile with this worksheet. After you unzip the file, move the script to the directory X:\chisel\bin, where X is the drive letter where you have installed chisel.

2.3 mknotes5.pl

This version of mknotes is tailored to work better with release 5 of Maple. It has the color commands built into it, and also implements the use of a style file called wrapfig.sty. When this is included as a package, you can wrap text around a figure, like so.

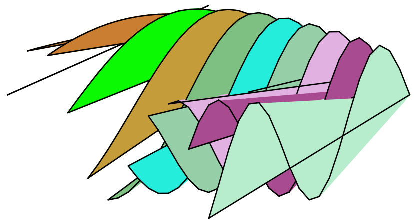


Figure 2: perls01.eps

The character styles are exported more cleanly to latex in release 5 than they were in release 4. In release 4, if you wanted to mark a bookmark with the character style Bookmark, you needed to leave 2 to 5 spaces after the bookmark, in order for the perl script mknotes.pl to know where the index item ended. In release 5, you needn't bother with this. Another feature that has been added to mknotes5.pl is that the modified tex files have been marked as modified and the date of modification noted.

Unlike the three scripts mknotes.pl, mkindex.pl, and mkcontents.pl, mkotes5.pl does not come with the chisel disk. It comes in the zipfile with this worksheet. After you unzip the file, move the script to the directory X:\chisel\bin, where X is the drive letter where you have installed chisel. As of 6/9/98, the latest version of mknotes5.pl is mknotes54.pl. To see a more complete description of the styles processed by mknotes54.pl, see the worksheet parag.mws.

2.4 clreps.pl

This script massages the eps plot files that are generated when a maple worksheet is exported to LaTeX. The eps plot files generated when a worksheet is exported to LaTeX are monochrome, the lines are too thin, and there is a black border around each picture. clreps.pl restores the standard 24 colors, thickens up the lines, and removes the border. If a greyscale number does not come from a standard color, then a linear interpolation is made between the two nearest colors.

Unlike the three scripts mknotes.pl, mkindex.pl, and mkcontents.pl, clreps.pl does not come with the chisel disk. It comes in the zipfile with this worksheet. After you unzip the file, move the script to the directory X:\chisel\bin, where X is the drive letter where you have installed chisel.

Usage:

perl -S clreps *.eps will modify all of the eps files in the current directory

3 Paragraph and character styles.

You can read generally about paragraph and character styles in the help sheets that come with Maple (see Creating Styles for those details). What we want to do in this worksheet is describe some of the predefined styles and most of the user defined styles which are used by the perl script mknotes.pl in preparing the tex file created when a worksheet is exported to LaTeX. This documentation is not complete: We are still adding user defined styles and modifiying mknotes.pl to make use of the styles. So this is only a snapshot. As of 6/9/98, the latest version of mknotes.pl is mknotes54.pl

When a worksheet using these styles is exported to LaTeX, the paragraphs which are in the style Bullet Item (for example) are inserted into the tex file with a \begin{Bullet Item} at the beginning and an \end{Bullet Item} at the end of the paragraph. If some character string is formatted with a character style, say for example, Help Normal, the exported string begins with \QTR{ and ends with }. Here is a sample, with the exported LaTeX following:

- This is an example **Bullet Item** with a String formatted in Help Normal.

```
\begin{Bullet Item}
This is an example \textbf{Bullet Item} with a
\QTR{Help
Normal}{String formatted in Help Normal.}
\end{Bullet Item}
```

Note that boldface is not a character style. It gets exported as \textbf{ }.

The style file maple2e.sty which must be used (as a package) will process some of these styles, but most of them are ignored, unless something additional is done by the user. The perl script mknotes.pl can be used to activate those exported paragraph styles. For example, it has been written to implement the Bullet Item, and Dash Item styles in LaTeX. For details on how mknotes works, see the mknotes.pl section of the worksheet perls.mws.

These styles can be modified from the Styles option of the format menu. See the help topic on

3.1 Pre-defined Paragraph styles

There are 18 pre-defined paragraph styles in MapleV5.

Normal The normal style is the style that is most common in a worksheet. Most of the text is in the normal paragraph form. It is important when typing in the normal style not to press the enter key when you get to the end of the line. Simply keep typing and the text will wrap automatically. Text entered in this way will reformat naturally if the width of the worksheet is changed.

Bullet Item, List Item, and Dash Item. These three styles each offset the text a few points to the right. This paragraph is set in the Bullet Item style, as you tell. To change the style of a paragraph, simply put the cursor in the paragraph and then use the mouse to scroll to the style you want in the upper left hand corner of the worksheet. Try this by changing the style of this paragraph to Dash Item.

Heading 1, Heading 2, ..., Heading 4. These 4 styles are used to head up sections, subsections, and subsubsections.

Title and Author styles. These styles are used for the title and author of a worksheet.

Error, Warning, Diagnostic. These styles are used by the Maple program when formatting error messages.

Fixed width, Text Output. These styles are used by the Maple program to format output.

Maple Plot, and Maple Output. Also used by Maple to format output.

3.2 User Defined paragraph styles

There are currently 10 or so user defined paragraph styles.

Problem, Theorem, Definition The styles Problem, Theorem, Definition were created to set off or clearly mark a problem, theorem, or definition where one is stated in the worksheet. The perl script mknotes.pl has been written to implement many of these these styles in LaTeX. The script mknotes5.pl implements all of these.

Definition. A point is that which has no breadth or length.

asis The asis style was defined and mknotes was written so that the hard newlines inserted into a paragraph are respected in the LaTeX output. Thus, this style is used to display various kinds of code such as C code or perl scripts which have been pasted into a normal paragraph in a worksheet.

```
#!perl
print "any last request?";
$line = <STDIN>;
if ($line =~ /^y/i) { print $line;}
exit;
```

texcomment. The style texcomment was defined to enable one to insert nearly invisible comments in a worksheet. The font color is set as a light shade of gray, and when the exported worksheet is massaged with mknotes, the texcommented paragraphs are commented out with %'s.

diagram. One can paste a graphic into a Maple worksheet, but unless the graphic was copied from a Maple plot the graphic is not exported when the worksheet is exported to LaTeX . The paragraph style diagram is used to mark the spot in the worksheet where the graphic is pasted. The font color is very light grey and the paragraph should contain only one line which has in it the name of the graphic, and the dimensions. If you have the style file wrapfig.sty available, you can add an additional argument to the line: put the character r if you want the picture on the right, l if you want it on the left.

3.3 Pre-defined Character Styles

There are 13 pre-defined character styles. To format a phrase with a character style, use the mouse to color (or select) the phrase, then use the mouse to scroll down to the desired style (in the upper left hand corner of the worksheet.

LaTeX Character strings marked with the LaTeX style are passed onto the LaTeX without change, hence the LaTeX commands that you want to put into the file should be put in the LaTeX mode.

Hyperlink This style is used to mark hyperlinks. I have trained mknotes.pl to remove leading hyperlinks, such as links to a table of contents worsheet that is often put at the bottom of a worksheet. There are numerous hyperlinks throughout this worksheet.

2d Output, 2d Comment, 2d Input, and 2d Math. These styles are used by the Maple program to format certain character strings. I leave them alone. Below is a phrase which has been formatted with these styles, what it looks like exported to LaTeX, and what it looks like after being massaged by mknotes.pl.

how now brown cow.
\begin{maplegroup}
\QTR{2D Comment}{how}
\QTR{2D Input}{now}
\QTR{2D Math}{brown}
\QTR{2D
Output}{cow}.
\end{maplegroup}
After massage –
{how} {\color[rgb]{1,0,0}now} {brown}
{\color[rgb]{0,0,1}cow}.

Help Heading, Help Normal, Maple Input, Output Labels, Plot Title, Plot Text, and Popup. These are also used by Maple. I haven't paid attention to them, yet.

3.4 User Defined Character Styles

There are several of these we have defined and there will likely be more. Mknotes.pl is written to preprocess them in the exported LaTeX worksheet.

Bookmark or bookmark. Use this style to mark a paragraph you have bookmarked from the View Menu. We have colored the text dark green, that way the bookmark will be visible in the worksheet and also make it into the index when you export to LaTeX. Here is what the paragraph looks like after export –

\QTR{paragraph}{Bookmark or bookmark.} Use
this style to mark a
paragraph you have bookmarked from the View Menu. That way the

\QTR{Bookmark}{bookmark} will be visible in
the worksheet and also
make it into the index when you export to LaTeX. Here is what the
paragraph looks like after export –
After mknotes.pl is applied to the paragraph, it looks like this –
\paragraph{Bookmark or bookmark.}
Use this style to mark a
paragraph you have bookmarked from the View Menu. That way the
{\color[rgb]{0,.5,0}\index{bookmark}bookmark} will be visible in the
worksheet and also
make it into the index when you export to LaTeX. Here is what the
paragraph looks like after export –

index Use the index style to mark other index items in a paragraph. Since there is only one bookmark per paragraph permitted in Maple, this is another character style to mark index items. We have colored the text a dark green, just as with the bookmark style to make it stand out in the paragraph. If you wanted to make the color different from that of bookmark, you can do this by modifying mknotes.pl appropriately.

paragraph Use the paragraph style to mark the heading of a paragraph. Most of the boldface above is set in the paragraph character style.

4 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 internet. We use it here as an editing tool. All of the scripts we use here 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 large.

You don't need to know any perl to run the scripts, as we have seen, but if you want to be able to modify a script or write your own, you need to know a little bit, and a good way to learn a little bit is to look at examples. Here are a couple of useful ones.

4.1 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.

4.1.1 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. Also, there are books on perl. *One good one is Learning Perl on Win32 Systems*, by Randal Swartz, Erik Olson, and Tom Christiansen, published by O'Reilly and Associates, Inc.

4.2 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 clreps.pl that performs the substitution. Note that it also changes the thickness of lines in the plot and removes the border around the plot.

4.2.1 clreps.pl

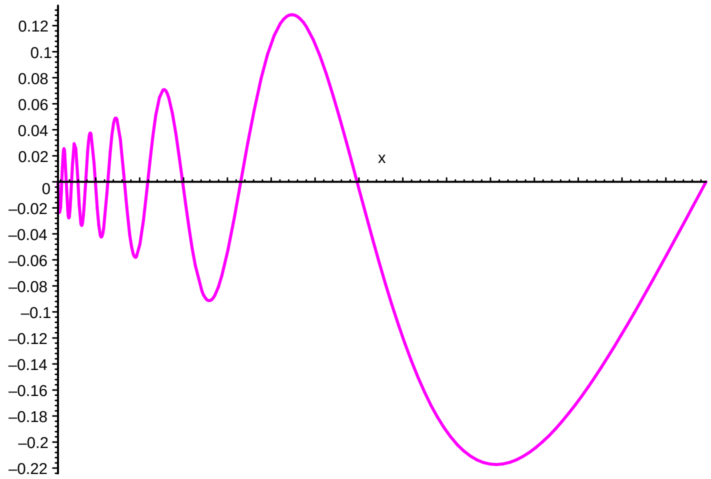
```
#!perl -i.bak
#
$opt = $ARGV[0];
$rel5 = "no"; $i=0;
$comout = "off";
if ($opt =~ /\-h/) {
print "Usage: perl -S clreps files to colorize black and
white\n";
print "       eps files generated when exporting Maple
worksheets\n";
print "       to LaTeX. Also thickens up lines in line
drawing\n";
print "       and removes the border around the
plot.\n";
exit;
}
@x =(0, .11, .179922, .234157, .291451, .3, .30549, .37451, .41,
.42698,
.55498, .59, .592431, .593843, .7, .701882, .752941, .774196,
.82949,.832784, .844706, .89, 1);
@r = (0, 0, .13725, .30980392, .55686, 1, .64706, .80000000, 1,
.55686275, .8000, 0,
.623529, 1, 0, .43922, .75294, .91764706, .55686, .84705882,
.67843137, 1, 1 );
```

```
@g =(0, 0, .13725, .18431373, .13725, 0, .16471, .19607843, 0,
.41960784, .4980, 1,
 .623529, .498, 1, .85882, .75294, .67843137, .13725, .84705882,
.91764706, 1, 1 );
@b = ( 0, 1, .55686, .30980392, .41961, 0, .16471, .19607843, 1,
.13725490, .1961, 0,
 .372549, 0, 1, .57647, .75294, .91764706, .41961, .74901961,
.91764706, 0, 1 );
#
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/;
if (/([0-9\.]+ G)/) {
$t1 = $1;
$t = $t1;
$t =~ s/ G//;
$i=1;
while ($x[$i] <  $t  && $i < 23) {$i
=$i+1;};
$red = $r[$i] -
($x[$i]-$t)*($r[$i]-$r[$i-1])/($x[$i]-$x[$i-1]);
$grn = $g[$i] -
($x[$i]-$t)*($g[$i]-$g[$i-1])/($x[$i]-$x[$i-1]);
$blu = $b[$i] -
($x[$i]-$t)*($b[$i]-$b[$i-1])/($x[$i]-$x[$i-1]);
s/$t1/$red $grn $blu C/;
}
#   This takes off the border for release 4 eps files
if  (/The following draws a box/ && $rel5 eq "no") {$comout =
"on";}
if  ($comout eq "on") {$i=$i+1;s/./\%/; if
($i==8) { $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

5 Make your own Calculus book

There is already a calculus book on the Chisel CD. It was written by Professor Neal Koblitz of the University of Washington for his calculus students. He has generously permitted the inclusion of the LaTeX source documents used to construct the book. These can be copied from the CD over to your hard disk and modified by you to produce a customized book. The customization might simply be exchanging "University of Kentucky" with the name of your school. Or it might be the deletion of a chapter, or permuting the order of some chapters. More substantial changes might include deleting or adding examples to a chapter or adding new topics to a chapter or adding a new chapter altogether. This document will try to tell you how to do these things, assuming you know a bit of LaTeX and have a willingness to experiment.

A word about copyright and history information. The source tex files used to produce the Kentucky Edition of Calculus I and Calculus II were written by Neal Koblitz, (the chapters on series were written by James Wells) and edited by Paul Eakin, Carl Eberhart, and Avinash Sathaye. Permission has been granted to duplicating services and individuals to make copies for student use. If you make minor changes to the book, simply add your name to the list of editors on the coverpage. If you make substantial changes to the book, you may want to add a copyright notice of your own. Please include in that notice a permission to duplicate for student use, so that the book remains an Affordable Text.

5.1 Copy the files to your hard disk and inspect

You need first to copy the directory koblitz\calc1\book (calculus 1) or koblitz\calc2\book (calculus 2) over to your hard drive. This can be done with point and click in the development subdirectory of koblitz\calc1 in Chisel.

We will give details on changing the calculus 1 book. Changing the calculus 2 book would proceed similarly.

First, click on the TexShell Icon on your screen. This opens up a special dos window which knows where everything is. All of the commands given below will work in the TeXShell; they won't work in an ordinary dos window unless you change the PATH.

The directory koblitz\calc1\book contains several files. Some of the important ones are:

- root.tex The file which controls the structure of the book.
- **coverpag.tex** and **preface.tex** These files contain the coverpage and the preface.
- **chapX.tex** and **hmwrkX.tex** where X is 1 to 18. There are the chapter and homework files.
- Lots of **fig*.ps** files. These are the figures that are in the book.
- **root.dvi** This file is generated when the command ***latex root*** is given. At that time lots of .aux and .log files are generated.

There are other files in the directory. You will only change tex files.

To see the .tex files you can change, give the command ***dir *.tex***

The main files to change are the root file *root.tex*, the coverpage file *coverpag.tex*, the preface file *preface.tex*, the chapter files *chap1.tex* throught *chap18.tex*, the homework files *hmwrk1.tex* through *hmwrk18.tex*, the lab files *instr.tex*, *bounce.tex*, and *pistonlb.tex*, the exam review files *midtrm1.tex*, *midtrm2.tex*, and *finlprac.tex*, and the answer file *hmwrkans.tex*.

5.2 To change the root file

Use your text editor to open root.tex. (edit or vi works fine. If you use wordpad which comes on the Chisel disk, be sure save your files as text documents.)

Say we wanted to remove chapter 18 and its homework from the book. Go to the bottom of root.tex, locate and remove the lines

\include{chap18}

\include{hmwrk18}

then save root.tex and exit your editor. Now, in the TexShell, issue the command

latex root (this trys to create a file root.dvi from root.tex)

If this command is successful, and it should be, you can inspect the new book with the dvi-viewer yap . Issue the command (in the TeXShell)

yap root.dvi

and inspect the file in the window that opens up. You can page through the file with pageup and pagedown keys. yap does not print the pictures, but you can create a postscript file root.ps which can then be viewed with the postscript viewer gsview32

dvips root; (this creates a postscript file root.ps from the dvi file root.dvi)

Now issue the command

gsview32 root.ps

and inspect the file in the window that opens up. Pageup and pagedown keys do not work in this viewer, but you can use the space bar and the minus '-' key to page down and up through the document. . Notice that the pictures are there for viewing.

The rootfile *root.tex* controls the formatting and contents of the book, so make all changes pertaining to form and content in it.

5.3 To change one of the other .tex files and inspect the result

Say you want to change the name University of Kentucky in the book to your school's name. Here is how do it. The name University of Kentucky occurs in the files coverpag.tex and preface.tex So to make the change to coverpag.tex, use your text editor to open coverpag.tex and replace, for example, University of Kentucky, Fall 1997 with your school name and a date of your choice in the line

flushleft Edited for the University of Kentucky, Fall 1997

Then save the file and exit the editor. *(Make sure the file you save has the original name coverpag.tex)*

Now remake the book by issuing the commands.

latex root This command generates a lot of messages and takes a while to complete. If many changes have been made, you need to do this twice.

dvips root This command generates a postscript file of the book called root.ps

gsview32 root.ps ***This command uses a postscript viewer to look at the file root.ps You can select pages and print them to your printer from this viewer.***

5.4 Add a new chapter to the book

Suppose you have written a chapter on the differential that fits naturally right after chapter 7. Convert it into a latex document and save it as chap7h.tex, then open the rootfile root.tex and insert the line

\include{chap7h}

between the lines

\include{chap7}

and

\include{chap8}

Now you can insert the changes into the whole book by issuing the commands

latex root you need to do this twice to get the references right

makeindex root.idx This command makes a new index for the book, including any entries that you may have put in. It generates a file called root.ind

move root.ind rootind.tex This command changes the name of root.ind

latex root Latex the book one more time to get the new index inserted

dvips root Create the postscript file root.ps

gsview32 root.ps Look at the new book.

5.5 Make other books too.

You can use the ideas and skills gleaned from making your own calculus book starting with Koblitz's book to making other books starting from scratch or your own set of notes.

Index