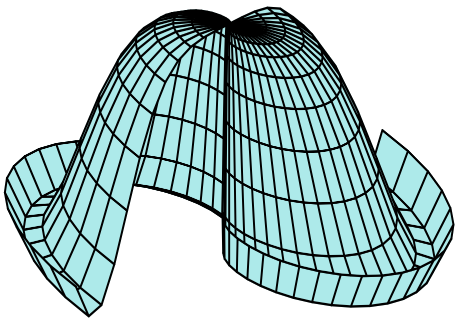
Making Large Documents with Maple

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



March 22, 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. One can also insert ***bookmarks*** to paragraphs in a worksheet and ***hyperlinks*** (bookmarks in) other 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 worksheet perls.mws if it can be found. 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 students, 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

At the start, one can simply print off a worksheet that one has written in 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 set of worksheets.*

This worksheet describes a perl5 script (mknotes.pl) which I wrote to massage the .tex files exported from MapleVr4 worksheets, before latexing them. In order to use it, you need to have perl5 installed on your machine. It is on the Chisel disk.

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.

2.1 mknotes.pl

Usage:

1. **perl -S mknotes -h** to get a brief 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. Change mknotes to mknotes -i in 2 or 3 and mknotes -t to mknotes -it to suppress the input cells in the LaTeX output. This is helpful when preparing papers or exams where you might not want the input to appear.

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.

Specifically, mknotes

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.

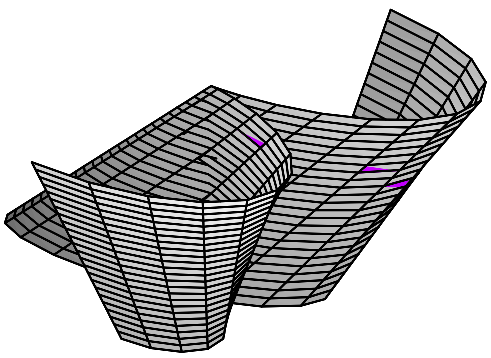
Sample:

```
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);
### WARNING: linalg[stack] has been renamed linalg[stackmatrix] to
avoid collision with mainline stack
A := stack(zrow,A);
### WARNING: linalg[stack] has been renamed linalg[stackmatrix] to
avoid collision with mainline stack
A := stack([op(vars),Z],A);
end:
```

Enables inclusion of externally drawn eps files. If you have a bitmap drawn with paint shop pro pasted into your worksheet, and you want this to appear in the latex output, do the following: 1. export your drawing to eps from paint shop pro. 2. Underneath the pasted in picture in the worksheet, open a new paragraph and type in the name you gave to the eps file (leave off the eps suffix). 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.). When you export this, mknotes makes the substitutions needed, even if there is no eps file.

Sample:



Handles user defined character style bookmark. Create a character style called bookmark, with green font say. Then when you make a bookmark, also highlight a word or phrase in that paragraph and convert it to the 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. As you may know, there is no passive union, intersection, or setminus in MapleVr4. mknotes fixes this, 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 could add your own latex macros and modify mknotes to recognize these.

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

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

$$\mathrm{union}(C,\,B,\,A)$$

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

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

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

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

$$(A\ \cup(B\ \cup C))$$

Or you can just make a plain ol proc.

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

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

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

$$(A\ \cup(B\ \cup C))$$

One can set up some things using '*'.

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

$$R\,Intersect\,(\overrightarrow{\mathrm{CD}}\ \cup\overrightarrow{\mathrm{AB}})$$

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

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

$$\mathit{angle}\ RVD = \mathit{ray}\ VR\,\mathrm{Union}(\mathit{ray}\ VD)$$

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 clreps.pl

This script massages the eps plot files that are generated when a maple worksheet is exported to LaTeX. As generated in release 4, the eps plot files are monochrome, the lines are too thin, and there is a black border around each picture. clreps.pl restores the standard 16 colors, thickens up the lines, and removes the border.

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 eps files in the current directory.

2.4 mknotes5.pl

This version of mknotes is tailored to work better with release 5 of Maple. It also has the color commands built into it.

Unlike the three scripts mknotes.pl, mkindex.pl, and mkcontents.pl, mknotes5.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.

3 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.

3.1 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*.

3.2 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.

3.3 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.*

3.4 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.

3.5 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.