MATH549 Exercise Sheet 3

Deadline for submission: Monday 20th October

The main business this week is to learn how to include graphics in your LaTeX document. If you're an M.Sc. student, this is one of the elements that must be present in your Maple-LaTeX project: the regulations stipulate that in order to get high marks for the LaTeX component, you need to include each of the following: a table of contents; referencing (i.e. `\ref` and `\cite`); a bibliography; displayed formulae; and at least one inserted graphic. If you're not an M.Sc. student, being able to include graphics is still a useful skill.

Packages

In order to include graphics in your documents, you need to know how to use packages. These are essentially "add-ons" to the basic LaTeX installation: collections of command and environment definitions which enable you to perform specialized tasks. Many packages are available: in the next few steps, you'll learn about some particularly useful ones. To make the commands in a given package available, include the line

`\usepackage{`*packagename*`}`

in your preamble. Traditionally `\usepackage` commands come immediately after the `\documentclass` command, so that someone looking at your file can easily see which packages it uses.

Exercise 3a

Start a new document called `yourname3.tex` with the `article` document style, using 11 point text and the A4 paper option. Increase the width of the displayed text by 1.4 cm, change the horizontal offset so that the text is still centred on the page, and open up the line spacing by 1.2 jots (if you've no idea what I'm talking about, you didn't read the *Aesthetics* section last week). Include the line `\input{yournamedefs}` to input the definitions which you created last week. Use `\maketitle` to give the document a title and advertise your authorship. Start a new (numbered) section called 'Packages'.

We'll start with a simple but useful package called `enumerate`. It changes the definition of the `enumerate` environment so that it takes an additional optional argument which describes how the item labels in the enumerated list should be formatted. If you don't include the optional argument, then the `enumerate` environment behaves exactly as it did before. Here's an example:

```
The following are equivalent:
\begin{enumerate}[a)]
\item $A$ is closed.
\item $X\setminus A$ is open.
\item If $(x_i)$ is a convergent sequence in $X$ with $x_i\in A$ for
all $i$, then its limit is contained in $A$.
\end{enumerate}
```

This produces:

The following are equivalent:

a) A is closed.

b) $X \setminus A$ is open.

c) If (x_i) is a convergent sequence in X with $x_i \in A$ for all i, then its limit is contained in A.

In the optional argument, any occurence of the characters `A`, `a`, `I`, `i`, or `1` is special: they are replaced in successive items by A,B,C,. . . ; a,b,c,. . . , I, II, III,. . . ; i, ii, iii,. . . ; and 1, 2, 3,. . . respectively. Any other characters in the optional argument (such as the `)` in the example above) are simply repeated in each label in the list. If you want to use one of the special characters as a normal character, you have to enclose it in braces. Thus

```
Here are some examples:
\begin{enumerate}[Ex{a}mple i):]
\item $f(x)=|x|$.
\item $f(x)=x^2$.
\item $f(x)=\cos\,x$.
\item Any polynomial with only even powers.
\end{enumerate}
```

produces

Here are some examples:

Example i): $f(x) = |x|$.

Example ii): $f(x) = x^2$.

Example iii): $f(x) = \cos x$.

Example iv): Any polynomial with only even powers.

For your example, produce the following:

Here are some properties of a pseudo-Anosov homeomorphism $f\colon M \to M$:

(pA 1) The stable and unstable foliations of f are unique up to multiplication of the measures by positive constants.

(pA 2) $h(f) = \log \lambda$.

(pA 3) If M has genus g, and n_p is the number of singular points and boundary components at which the invariant foliations of f have p prongs, then

$$\sum_{p=1}^{\infty} n_p(2-p) = 4(1-g).$$

(pA 4) The periodic points of f are dense in M.

Remember to put `\usepackage{enumerate}` in your preamble, to load this particular add-on to standard LaTeX. Do be careful with all aspects of formatting when creating this example, not just the item labels.

Exercise 3b

The American Mathematical Society (AMS) provides a number of useful packages. One which you will almost certainly use is `amssymb`: this gives access to all the symbols listed in Tables 3.12 – 3.19, and also provides the *blackboard bold* symbols $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, $\mathbb{N}$, and so on. These are obtained by `\mathbb{C}`, `\mathbb{R}` etc.: you will probably use these often enough that it's worth defining new commands `\C`, `\R` etc. for them in your `yournamedefs.tex` file. Do this now.

The package `amsmath` also provides a number of useful commands. One of them, `\text`, provides an alternative to `\mbox` for including text in mathematical expressions: it is better than `\mbox` in that it automatically changes the size of the text to make it appropriate to the context. Try reproducing the formula

$$\int_C f(z)dz = 2\pi i \sum_{\text{poles } z} R_z$$

from sheet 1 using this command. `\text` is also useful in constructions such as `$1^\text{st}$` and `$2^\text{nd}$`, which produce 1^st and 2^nd respectively.

Exercise 3c

Another useful AMS package is `amsthm`, which provides tools for defining theorem-like environments. It provides a `\theoremstyle{}` command which takes one argument (either `plain`, `definition` or `remark`) and can be used to change the style of the environments defined by a `\newtheorem` command (see previous sheet). For example, try putting

```
\theoremstyle{plain}
\newtheorem{thm}{Theorem}

\theoremstyle{definition}
\newtheorem{defn}[thm]{Definition]

\theoremstyle{remark}
\theoremstyle{rem}{Remark}[section]
```

in the preamble and then adding

```
\begin{defn}
Definitions are typeset in the same way as the rest of the text,
but with a bold heading.
\end{defn}
\begin{thm}
Theorem's are important, so are typeset in an italic font for
emphasis.
\end{thm}
\begin{proof}
The package also provides a ready-made environment for proofs,
including an empty box at the end.
\end{proof}
\begin{rem}
Remarks are typeset with an italic heading but otherwise in the same
way as the main text.
\end{rem}
```

to produce

Definition 1. Definitions are typeset in the same way as the rest of the text, but with a bold heading.

Theorem 2. *Theorem's are important, so are typeset in an italic font for emphasis.*

Proof. The package also provides a ready-made environment for proofs, including an empty box at the end. □

Remark 3. Remarks are typeset with an italic heading but otherwise in the same way as the main text.

(If you've defined your own environment called 'proof' LaTeX will complain because the environment now has two definitions. If you want to use your own proof environment and still use the amsthm package then replace the `\newenvironment` command in your personal definition by the `\renewenvironment` command.)

The options for numbering theorem-like environments and for naming individual instances are the same as before. The package provides a command `\swapnumbers` which, if placed at the beginning of the list of `\newtheorem` statements causes the numbering to come before the name. To produce un-numbered theorem-like environments use the `\newtheorem*` command. For further documentation see `http://math.ucsd.edu/~jeggers/latex/amsthdoc.pdf`.

Define new theorem-like environments so that you can reproduce the following (the numbering of the corollary may be different in your version):

Corollary 4 (of the Gauss–Bonnet theorem)**.** *The only closed surface with a flat metric is the torus.*

Note. Not every metric on the torus is flat, for instance that induced from the standard embedding in $\mathbb{R}^3$ is not.

Exercise 3d

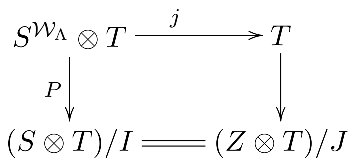
A very powerful package is `xy-pic`, which provides drawing capabilities within LaTeX. To load this package add `\usepackage{xy}` to your preamble. Details of the (extensive) capabilities of this package can be found at

`http://www.tug.org/applications/Xy-pic/`

One simple use is in creating commutative diagrams. (If you don't know what a commutative diagram is, and can't imagine you'd ever want to produce one, feel free to ignore this exercise.) Here's an example:

```
\[
\xymatrix{
S^{{\mathcal{W}}_\Lambda}\otimes T \ar[r]^j \ar[d]_{P} & T \ar[d] \\
(S\otimes T)/I \ar@{=}[r] & (Z\otimes T)/J
}
\]
```

produces the diagram



(Note the use of `\mathcal` to produce *calligraphic* characters). Table 3.14

Inside the `\xymatrix{ }` command the entries of the diagram are laid out on a grid by using `&` to separate entries and `\\` to end a line (just as in the `array` environment). To create an arrow starting at an entry and ending, for example, two entries to the right and one down place `\ar[rrd]` after the starting entry (and before the `&`). Several arrows can start at the same entry, simply add more `\ar` commands with the appropriate string of l,r,u,d in square brackets to specify the relative end point. There are numerous sophisticated options for changing the type of arrow, for instance to get an equals sign as in the above example, making arrows curve, go under or around other entries and so on. Labels can be placed on either side of an arrow by adding them as sub or superscripts to the command defining the arrow.

For your example, produce the following at the end of your document:

$$\begin{array}{ccc} H^p(X)\times H_{p+q}(X,\widehat{A}) & \xrightarrow{\cap} & H_q(X,\widehat{A}) \\ {\scriptstyle \mathrm{id}\times}\Big\downarrow{\scriptstyle \partial} & & \Big\downarrow{\scriptstyle \partial} \\ H^p(X)\times H_{p+q-1}(\widehat{A}) & \xrightarrow[\cap]{} & H_{q-1}(\widehat{A}) \end{array}$$

(Use `\widehat{A}` to obtain $\widehat{A}$. Other similar *math mode accents* are listed in Table 3.1. To get the labels on the horizontal arrows in the correct place you will need to add some spaces.)

When you come to write your own documents, there will almost certainly be tasks you want to do which haven't been covered in these exercise sheets. Whatever the task, there is probably a ready-written LaTeX package to do it.

Try `http://www.tex.ac.uk/tex-archive/help/Catalogue/catalogue.html` (there's a link on the module webpage) if you're looking for a package to perform some particular task. The *Topical index* is a good place to start, though you could also look at the *Alphabetic index* to get an idea of the number of packages available.

Not all of these packages are installed on the machines in room 302. If you find there's a package that you *really* need which isn't installed, then come and talk to me about it: often it's just a matter of downloading relevant files and putting them in the same folder as the LaTeX document you're working on, or adapting the instructions on the module webpage for installing the `pinlabel` package.

Including Graphics

To include graphics in your document you use the `graphicx` package. In these exercises you'll start with a ready-made graphics file which I'll provide you with, and then produce some of your own.

pdfLaTeX has facilities for including graphics files in `pdf`, `jpg`, and `png` formats. If you have images in some other format, it is usually quite straightforward to convert them to one of these.

The initial graphics file used in this worksheet, for reasons which will soon become apparent, is in a different format: it is an *Encapsulated Postscript* (`eps`) file. The first thing to do, then, is to download it and convert it to PDF.

Go to the module webpage, and follow the instructions there for downloading the file `example.eps`, putting it in the same folder as `yourname3.tex`. Carry out the following instructions to convert it to a PDF file `example.pdf` (if you're not used to using the command prompt in Windows, this will seem like a mystical incantation — call me over if you can't get it to work):

1. Open a command prompt window by choosing `Run...` from the `Start` menu, and typing `cmd` in the dialog box which appears.

2. In the command prompt window which appears, type the following three lines (assuming that `example.eps` is in the folder `M:\documents\latex`):

   ```
   m:
   cd \documents\latex
   c:\miktex26\miktex\bin\epstopdf example.eps
   ```

   If you've installed LaTeX on your own computer using my installation CD, you should replace `miktex26` in the last line with `texmf`. Note that you can cut down on typing by hitting the `TAB` key to complete partially typed folder and file names: for example, typing

   ```
   c:\mik TAB \mik TAB \bin\epstopdf exa TAB
   ```

   should produce the final line above. Try opening the file `example.pdf` which should now have been produced to see the image you're going to insert into your document.

Exercise 3e

Add the line
`\usepackage{graphicx}`
to your preamble. Start a new (numbered) section called 'Graphics', and put the following in it:

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=0.8\textwidth]{example}
\caption{A $1$-pruning disk in the horseshoe}
\label{fig:example}
\end{center}
\end{figure}
Figure~\ref{fig:example} is my first example.
```

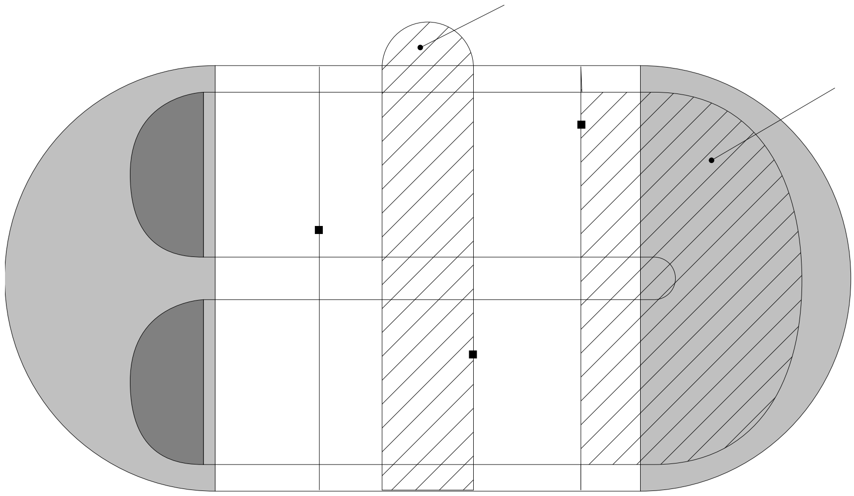


Figure 1: A 1-pruning disk in the horseshoe

Process your file (twice, to get the `\ref` to come out right). If you now view the document, you should see something like Figure 1.

There are several things to explain here.

a) `figure` is the LaTeX environment for displaying figures: it has an optional argument which specifies where in your document you'd like the figure to appear. The argument used above, `htbp`, says that your first preference is for the figure to appear right here (at the point in the text where the `figure` environment occurs); if that fails (probably because there isn't room on the page), your second preference is for it to be at the top of some page; your third is for it to be at the bottom of some page; and if all that fails, you'd like it to go on a page which contains nothing but figures and other *floats*. If you don't include an optional argument, it's taken to be `tbp`. Page 45

b) I usually put my figures in a `center` environment, so they appear centred on the page and not flush against the left margin.

c) `\includegraphics` is the command which actually inserts the graphics. It takes one compulsory argument, which is the name of the graphics file to be included (here `example.pdf`: you mustn't specify the `.pdf`). The optional argument `width=0.8\textwidth` specifies that you want the graphic to be scaled so that its width is 4/5 of the width of the text on the page. There are other possible optional arguments (see Table 4.1 on page 72), but I find this by far the most useful. In fact, I have a command defined by
`\newcommand{\pichere}[2]{\includegraphics[width=#1\textwidth]{#2}}`

so that I can just type
`\pichere{0.8}{example}`.

d) The `\caption` command is self-explanatory.

e) If you want to give a label to a figure, the `\label` must come after the `\caption`. This is because it's possible to include several graphics in a single `figure` environment, each with its own caption, and LaTeX has to know which particular graphic your label is referring to.

Note that there's no need to put your graphics in a `figure` environment: you should only do that if you want a numbered figure with a caption. You can use the `\includegraphics` command anywhere in your document to insert a figure. By the way, remember that in sheet 2 you produced a table using the `tabular` environment. If you want your table to be numbered and have a caption, you can wrap the `tabular` environment inside a `table` environment, which works pretty much the same way as `figure` (see pages 44 – 46).

Exercise 3f

There are no labels on Figure 1. The reason for this is that drawing packages, while they have facilities for putting text on your picture, can't produce labels of the same quality as LaTeX. If you want the labels on your figure to look as good as the rest of your document, you can use the `pinlabel` package. Follow the instructions on the module webpage for installing this package, and include `\usepackage{pinlabel}` in your preamble.

Now add the following lines just after `\begin{figure}`:

```
\labellist
\small
\pinlabel {$J_0$} at 78 48
\pinlabel {$F(D)$} [bl] at 501 240
\endlabellist
```

Process the file and study the result. You should find that two labels, J_0 and $F(D)$, have been added to the figure.

How does this work? The labels to be added come between the `\labellist` and `\endlabellist` commands. The `\small` command says that these labels should be a bit smaller than the normal font size in your document (this generally produces a better appearance for figure labels). The first `\pinlabel` command says that the label J_0 should be placed on the figure at coordinates $(78, 48)$.

How do you know what coordinates to use? Open up the file `example.eps` by double-clicking on it. It should open in a program called *Ghostview*: if it doesn't, then you need to create an association between `eps` files and the Ghostview program, which is `V:\gv48\gsview\gsview32.exe` — call me over if you don't know how to do this. Once the figure is open in Ghostview, observe that the coordinates of the cursor in Ghostview are displayed in the program's status bar. It is these coordinates that you need to give to `pinlabel` (and this is why you need to have the figure in `eps` format as well as in PDF format).

The second `\pinlabel` command has an optional argument `[bl]`. The reason for this is that you want the label $F(D)$ to appear, not centered on the end of the arrow in the figure, but with the bottom left of the label on the end of the arrow. The coordinates of the end of the arrow are exactly $(501, 240)$. `pinlabel` puts the label $F(D)$ so that its bottom left is not exactly at these coordinates, but shifted slightly away from them to leave a space between label and arrow.

Add further `\pinlabel` commands to your figure, so that the end result is like Figure 2. Use `\overline{101}` to obtain $\overline{101}$. If you're running short of time, just do enough of the labels to be sure you've got the idea.

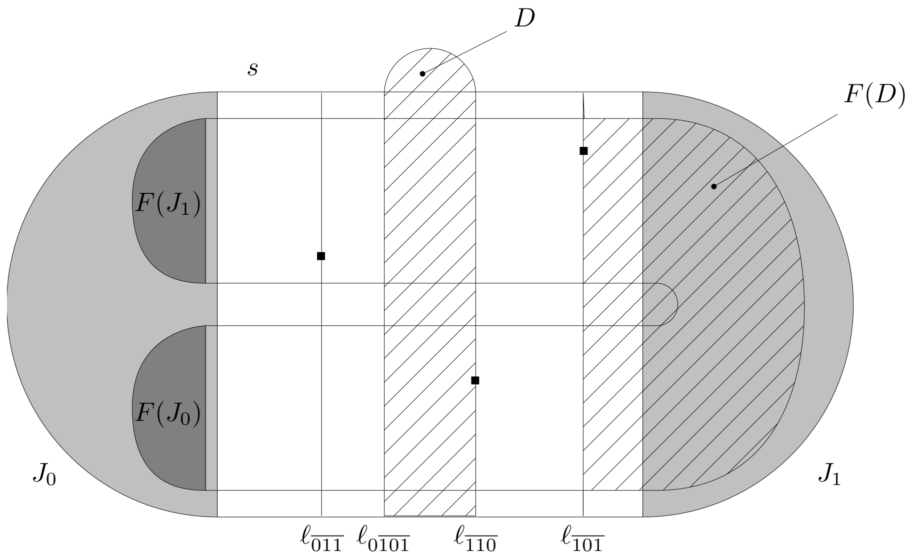


Figure 2: A 1-pruning disk in the horseshoe

Creating Encapsulated PostScript Files

You now know how to include graphics in your documents: creating the graphics in the first place can be a much more complicated business, and one which doesn't really come under the remit of this module. Here are some of the options. There'll be specific exercises using a) and b) in just a moment.

(Note that if you don't need to put labels on your figure then you can create PDF or `jpg` files directly. If you do want to include labels, then you need to start with an `eps` file (so that you can read off the coordinates) and convert it to PDF.)

a) M.Sc. students will soon be learning Maple. There's a relatively simple way to produce `eps` files from Maple graphical output.

b) There's a simple figure drawing program called *Winfig* installed on the computers in room 302, which produces `eps` output. It isn't much of a program, but is OK for simple figures. This program (at least the trial version of it) can also be downloaded for use on your own computer, from `http://www.schmidt-web-berlin.de/winfig/`

c) In University computer centres, you can use the program *Corel Draw* to produce `eps` files (`Start->Install->Graphics->CorelDrawX4`). However, this is a full-featured professional drawing program which takes a long time to learn, and using it to produce simple line figures is rather like using a sledgehammer to crack a nut.

Exercise 3g

Start up Maple 11 (either double click the **Maple 11 Classic Worksheet** desktop shortcut, if there is one, or find it under `Start->Programs->Maple11`. If you can't get it running, there's some help at `http://www.liv.ac.uk/~jonwoolf/Math549/maple/`: call me over if it still doesn't work). Type the following:

```
plot(sin(x)+cos(2*x),x=-2*Pi..2*Pi);
```

to display a plot of the function $\sin x + \cos 2x$ in the range $-2\pi \leq x \leq 2\pi$.

Suppose you want to include this plot in a LaTeX document.

Right-click on the image and select `Export As` from the pop-up menu, then choose the EPS option. Save the file as `yournamegraph.eps` in the same folder as `yourname3.tex`. Finally convert the file to a PDF file `yournamegraph.pdf`. You can check this file by opening it in Acrobat Reader (it may be on its side). Experiment with the different options (`Style`, `Legend`, `Axes`) on the pop-up menu in Maple. Keep Maple running when you've finished, as you'll be using it again in a bit.

Include this plot in your document as a new figure, taking up 30% of the width of the text in your document, with a suitable caption. Process your LaTeX file and have a look at the output. If the figure's on its side, add `angle=-90` inside the `\includegraphics` commmand to solve the problem.

Exercise 3h

In this exercise you'll create a simple graphic with Winfig: rather than give you precise instructions, I'll provide a few tips, leave you to experiment with the program yourself, and expect you to call me over for help. You can use Corel Draw or any other program you know if you prefer — but don't expect me to give much help in that case!

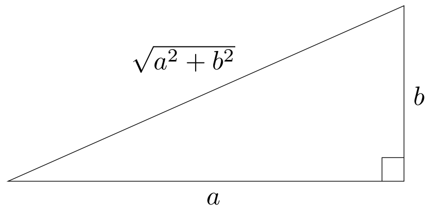


Figure 3: Pythagoras's theorem

Your goal is to include a figure in your document which looks like Figure 3.

Start up Winfig, and try to produce a triangle like the one in the figure. (You'll add the labels later with `pinlabel`.) Here are a few tips:

a) To draw a triangle, you should select the *polygon* tool (not *polyline* or *regular polygon*). As your mouse floats over the buttons on the left of the screen, the names of the tools appear in the status bar at the bottom of the screen.

b) Having selected the *polygon* tool, various options concerning line width, style, etc. appear at the right of the screen.

c) To draw a triangle, left-click at the first point, move the mouse and then left-click at the second point, then move the mouse again and *right*-click at the third point. (This isn't very intuitive at first.)

d) When you've finished, first save your drawing (as a `fig` file: this enables you to read it back into Winfig later). Then select `Export` from the `File` menu, and select the `EPS` tab in the dialog box which appears. You don't need to change any of the options: just click `OK` and then choose an appropriate filename (`yournametriangle.eps`) and folder for your file. You can then convert this to a PDF file (`yournametriangle.pdf`).

e) Yes, I know, it's not much good.

Now write the `figure` environment and the `\pinlabel` commands to include the figure in your document, taking up 10% of the *height* of the page.

Putting two or more graphics side by side

Exercise 3i

Sometimes you want to put two or more graphics side by side in your document, perhaps for comparison purposes. To do anything really elaborate, you could look at the `subfig` package: but here's how to deal with two simple cases.

First, suppose you want two graphics side by side in the same figure (so they share a common figure number and caption). Then you can just put them one after the other in your `figure` environment: the following produces the output of Figure 4. (Here `hallgraph2.pdf` is the graph of sin $2x$ + cos x. The `\qquad` puts a bit of space between the two graphics (see Sheet 2). Be careful not to leave a blank line between the two `\includegraphics` commands.)

```
\begin{figure}[htbp]
\begin{center}
\includegraphics[width=0.3\textwidth,angle=-90]{hallgraph} \qquad
\includegraphics[width=0.3\textwidth,angle=-90]{hallgraph2}
\caption{The graphs of two functions}
\label{fig:twofunctions}
\end{center}
\end{figure}
```

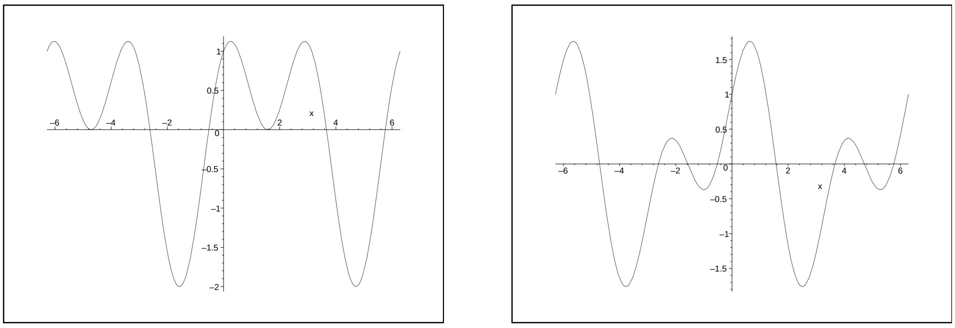


Figure 4: The graphs of two functions

Second, suppose that you want each of the two graphics to have its own figure number and caption. You can do this using the `minipage` environment, which creates a little page within a page. The following produces the output of Figures 5 and 6. It isn't pretty, but it works!

```
\begin{figure}[htbp]
\begin{center}
\begin{minipage}{0.45\textwidth}
\begin{center}
\includegraphics[width=0.5\textwidth,angle=-90]{hallgraph}
\end{center}
\caption{$\sin\,x+\cos\,2x$}
\label{fig:first}
\end{minipage}
\begin{minipage}{0.45\textwidth}
\begin{center}
\includegraphics[width=0.5\textwidth,angle=-90]{hallgraph2}
```

```
\end{center}
\caption{$\sin\,2x+\cos\,x$}
\label{fig:second}
\end{minipage}
\end{center}
\end{figure}
```

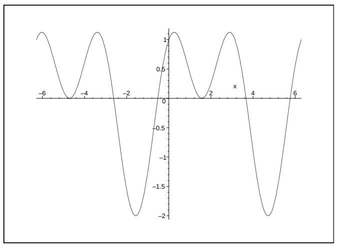


Figure 5: $\sin\, x + \cos\, 2x$

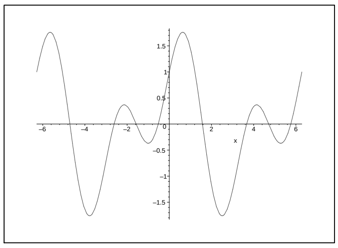


Figure 6: $\sin\, 2x + \cos\, x$

For your example, produce *three* graphs side by side, each with its own caption and figure number. Let the first be the graph of $\sin\, x + \cos\, 2x$, which you've already saved as `yournamegraph.pdf`: save the others, which can be any graphs you like, as `yournamegraph2.pdf` and `yournamegraph3.pdf`.

The final haul

In this final section, you'll practise putting together what you've learned to reproduce a sizeable chunk of mathematical text. You may find this rather tedious, but in my opinion it's much more valuable than just doing a sequence of short exercises. I've tried to pick the text so that it uses many of the techniques you've learned so far.

Exercise 3j

Start a new section in your document called 'The End'. In that section, reproduce the following text.

Theorem 5 (Artin, 1925). *The group $\pi_1 B_{0,n}$ admits a presentation with generators $\sigma_1, \ldots, \sigma_{n-1}$ and defining relations:*

$$\begin{array}{rcll} \sigma_i\sigma_j & = & \sigma_j\sigma_i & \quad |\, i-j\,| \geq 2 \\ \sigma_i\sigma_{i+1}\sigma_i & = & \sigma_{i+1}\sigma_i\sigma_{i+1} & \quad 1 \leq i \leq n-2. \end{array}$$

Proof. (The proof given here is due to Fadell and Van Buskirk, 1962). Let B_n be the abstract group with the presentation of Theorem 5. Until we have established the isomorphism between B_n and $\pi_1 B_{0,n}$ we will use the symbols $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_{n-1}$ for elements of $\pi_1 B_{0,n}$, with $\imath\colon B_n \to \pi_1 B_{0,n}$ defined by $\imath(\sigma_i) = \tilde{\sigma}_i$ for $1 \le i \le n-1$. The elements $\tilde{\sigma}_i$ have already been defined pictorially: we now give an equivalent definition which is more precise. Recall the covering projection $p\colon F_{0,n} \to B_{0,n}$. Choose the point $p\left((1,0),\ldots,(n,0)\right) = \tilde{z}^0$ as base point for the group $\pi_1 B_{0,n}$. Lift loops based at $\tilde{z}^0$ in $B_{0,n}$ to paths in $F_{0,n}$ with initial point $((1,0),\ldots,(n,0))$. Then the generator $\tilde{\sigma}_i \in \pi_1 B_{0,n}$ is represented by the path $\mathcal{F}(t)$ in $F_{0,n}$ given by

$$\mathcal{F}(t) = \left((1,0),\ldots,(i-1,0),\mathcal{F}_i(t),\mathcal{F}_{i+1}(t),(i+2,0),\ldots,(n,0)\right),$$

where $\mathcal{F}_i(t) = (i+t, -\sqrt{t-t^2})$ and $\mathcal{F}_{i+1}(t) = (i+1-t, \sqrt{t-t^2})$. That is, $\mathcal{F}(t)$ is constant on all but the ith and $i+1$st strings, and interchanges those two in a nice way.

The proof of Theorem 5 will be by induction on n, and will exploit the relationship already developed between $\pi_1 B_{0,n}$ and $\pi_1 F_{0,n}$. Let

$$\tilde{\nu}\colon \pi_1\left(B_{0,n}, \tilde{z}^0\right) \to \Sigma_n$$

be defined as follows: let $\tilde{\alpha} \in \pi_1 B_{0,n}$ be represented by a loop

$$\tilde{g}\colon\ (I,\{0,1\}) \to \left(B_{0,n}, \tilde{z}^0\right)$$

and let $g = (g_1,\ldots,g_n)\colon (I,\{0\}) \to (F_{0,n}, z^0)$ be the unique lift of $\tilde{g}$. Define

$$\tilde{\nu}(\alpha) = \begin{pmatrix} g_1(0),\ldots,g_n(0) \\ g_1(1),\ldots,g_n(1) \end{pmatrix} \in \Sigma_n.$$

The kernel of the homomorphism $\tilde{\nu}$ is the pure braid group $\pi_1 F_{0,n}$. Corresponding to the homomorphism $\tilde{\nu}$ is the homomorphism

$$\nu\colon B_n \to \Sigma_n$$

from the abstract group B_n to the symmetric group Σ_n on n letters defined by

$$\nu(\sigma_i) = (i, i+1) \qquad 1 \le i \le n-1.$$

Let $P_n = \ker \nu$. The proof can be completed as a consequence of the following lemma:

Lemma 6. *The homomorphism $\imath\colon B_n \to \pi_1 B_{0,n}$ is an isomorphism onto if $\imath|_{P_n}$ is an isomorphism onto $\pi_1 F_{0,n}$.*

Proof. (Lemma 6). The homomorphism ν is clearly surjective, since the transpositions $\{\nu(\sigma_i)\}$ generate Σ_n. Hence we have a commutative diagram

$$\begin{array}{ccccccccc}
1 & \longrightarrow & P_n & \longrightarrow & B_n & \xrightarrow{\nu} & \Sigma_n & \longrightarrow & 1 \\
\| & & \downarrow{\scriptstyle \imath_n = \imath|_{P_n}} & & \downarrow{\scriptstyle \imath} & & \| & & \| \\
1 & \longrightarrow & \pi_1 F_{0,n} & \longrightarrow & \pi_1 B_{0,n} & \xrightarrow[\tilde{\nu}]{} & \Sigma_n & \longrightarrow & 1
\end{array}$$

with exact rows. Applying the Five Lemma, we obtain the desired result. This completes the proof of Lemma 6. □

Theorem 5 now follows. □

Submission

Submit the assignment to `jonwoolf@liv.ac.uk` by email, making sure you attach `yourname3.tex`, `yournamedefs.tex`, `yournamegraph.pdf`, `yournamegraph2.pdf`, `yournamegraph3.pdf`, and `yournametriangle.pdf` (that's six files in all).

Appendix: where to go from here

You can do almost anything in LaTeX: these sheets have only touched on some of the major features. When you're writing longer documents, you may well find there's something you want to do which I haven't treated. The first things you should do when this happens are:

- See if there's anything relevant in your *Not So Short Introduction*.
- Have a look at the Topical Index in the catalogue of LaTeX packages mentioned earlier in this sheet, and linked from the module webpage.
- Remember that Google is your friend.

Here are some brief details of a few important topics which haven't been covered in the sheets, mostly based on requests for help that I've had from students in previous years.

1. Appendices

To put appendices in your document, include the `\appendix` command just before the first chapter or section which you want to be an appendix. Try the following:

```
\documentclass[11pt,a4paper]{article}

\begin{document}
\section{First section}
Blah blah.
\section{Second section}
Blahdyblahdyblah.
\appendix
\section{First appendix}
Here's my Maple code.
\subsection{Comments on the code}
It all works very well.
```

```
\section{Ideas for further research}
I don't have any.
\end{document}
```

Try changing the documentclass to `report`, and changing each `\section` to `\chapter`, and the `\subsection` to `\section`.

2. Landscape orientation

Sometimes you want one or more pages of your document to be in *landscape*, as opposed to *portrait*, mode. The most common reason for this is that you want to include a very wide figure or table, which doesn't fit on the page in portrait mode.

The way to do this is with the `lscape` package. Try out this example, which shows how it works:

```
\documentclass[11pt,a4paper]{article}
\usepackage{lscape}

\begin{document}
Here's a page in portrait mode.
\begin{landscape}
Here's a page in landscape mode.
\end{landscape}
And another one in portrait.
\end{document}
```

3. Working with large bibliographies

If there are only two or three entries in your document's bibliography, then you can just type the bibliography by hand as explained in sheet 2. For longer bibliographies, however, there's a tool called BibTEX which makes things much easier.

As an example, let me show you how to create the bibliography of sheet 2 using BibTEX.

1. Make a copy of your file `yourname2.tex` with a different filename, such as `test.tex`, and load it up in TEXShell.

2. Go to `http://www.ams.org/mathscinet`. Note that the University has paid for access to MathSciNet: you won't be able to access it from your home computers. Click on *Full Search* on the left of the page. To find the first paper in the bibliography (by Bell and Meyer), change the search fields in the top two drop down boxes to *Author*, and type "Bell" in the edit box next to the first *Author*, and "Meyer" in the edit box next to the second *Author* (or the other way round, it doesn't matter). Click the *Start Search* button, and it should find a review of the relevant paper.

Now in the listbox which reads *Select alternative format*, choose *BibTeX* and click *Go*. In the page which appears, select and copy everything from `@article` down to the final `}`. Open a new document in *Notepad* and paste in what you've copied. Change `MR1348733` in the first line to `BelMey`: this is the label you'll use to refer to this paper from within your LaTeX document. Save the file in the same folder as `test.tex`, with the filename `refs.bib`. (You have to choose `All Files` in the `Save as type` drop down box in the save dialog.)

To find the second reference, go back to the search page of MathSciNet and replace Bell and Meyer with Bestvina and Handel. Since there are several papers authored by these two, you need to select the one you're looking for, which is called *Train tracks for surface homeomorphisms*. Click on the `MR1308491` next to this entry to bring it up, then follow the same procedure as with Bell and Meyer, pasting the BibTeX information into the end of `refs.bib`, and changing `MR1308491` to `BestHan`.

Add a couple more random references to the end of `refs.bib` in the same way.

3. Now return to your file `test.tex`. Go to the end of it, and delete the whole bibliography that you inserted before (from `\begin{thebibliography}` to `\end{thebibliography}`). Replace it with the single line `\bibliography{refs}`.

4. Go to the top of your document, and put the line `\bibliographystyle{plain}` in your preamble.

5. Click the `LaTeX` button, then the `BibTeX` button, then the `LaTeX` button twice more. Study the results.

6. Change `\bibliographystyle{plain}` to `\bibliographystyle{alpha}`, and see what changes. There are many different bibliography styles available: try googling *latex bibliography style*.

7. Note that only those references which are actually cited in your document appear in the bibliography. Thus you can keep one master copy of `refs.bib`, and keep adding to it whenever you find a reference relevant to your work. In each particular LaTeX document you write, you can then just cite the references which are relevant to that document.

4. Creating an index

I've never written a document with an index in my life, but they're certainly useful for some long documents. LaTeX has a support program called `makeindex` to help you to generate an index. You can find details on pages 75 and 76 of your Not So Short Introduction.