

Using Latex in Pressbooks

Using Latex in Pressbooks

lgreenstreet



Using *Latex in Pressbooks by lgreenstreet* is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/), except where otherwise noted.

Contents

Introduction	vi
About Latex	
Why Use Latex?	2
Motivating Examples	3
Equations and Symbols	
Equations	6
Equation Practice Exercises	8
Equation Solutions	9
Environments	10
Accents and Symbols	12
Packages	
More Symbols and Languages	16
Plots	17
Plot Practice Exercises	22
Plot Solutions	24
Resources	28
Latex Outside of Pressbooks	29

Introduction

The goal of the book is to provide an introduction to incorporation Latex into a Pressbooks document. The guide does not assume any prior knowledge of Latex. In fact, the first section provides some background information to help readers decide if Latex is right for their project. The second section shows how to get the QuickLatex plugin running in Pressbooks and provides some worked examples to help new users get started typesetting mathematical equations, accents, and symbols. The third section explains how to use packages with the QuickLatex plugin and provides worked examples of using packages to typeset non-Latin languages and create plots. The final section provides some information on the features of Latex outside of Pressbooks and includes a list of resources. The sections are independent, so feel free to jump to a specific section based on your background and interests.

About Latex

Latex is a typesetting program that was created by computer scientist and mathematician Donald Knuth in 1983. Donald Knuth has many achievements in computer science, including creating the methodology of ‘literate programming’ which encourages programmers to think of their code as a work of literature, not just a program to be run on a machine. The creation of Tex influenced and was influenced by literate programming.

Latex is best known for typesetting mathematical equations but also offers precise control over the typesetting of the entire document. Latex main competitors are “What You See is What You Get” (WYSIWYG) editors such as Microsoft Word and Adobe InDesign. In contrast to WYSIWYG editors, Latex allows you typeset equations quickly using plain text. Latex is used widely in academia in fields including math, computer science, physics, chemistry, linguistics, political science, and music. Latex has been integrated into many other systems including WordPress and Gmail, usually to facilitate math typesetting.

People are often unsure how to pronounce Latex. The name is a shortening of Lamport Tex and can be pronounced as LAY-tekh or LAH-tekh.

Why Use Latex?

Latex is the standard program for typesetting mathematical equations and is a very popular choice for typesetting multi-lingual works and music. Its plain text commands allow authors to write a variety of symbols quickly without clicking through many menus and allow authors control over formatting and alignment. Additionally, Latex is free, has good documentation, and has a large active community which have developed a range of freely available modifications. Thanks to the community, it is easy to find answers to common questions through Google and the Latex community actively answers new questions.

As a stand-alone program, Latex has a steep learning curve because you need to relearn how to do basic document formatting that is done using the graphical interface of WYSIWYG editors. Using Latex with Pressbooks makes the process easier, as you can use HTML for most of the document, only typesetting elements that are difficult to create without Latex. Latex can be incorporated into Pressbooks using the QuickLaTeX plugin, which allows shortcode to be used to include small snippets of Latex as well as provides a mode to interpret Latex syntax site wide.

Motivating Examples

Latex is best known for typesetting math equations. However, many packages have been developed to create other difficult to typeset elements including plots, chemical equations, multi-lingual documents, and music. Below are examples of some of the applications of Latex.

Mathematics

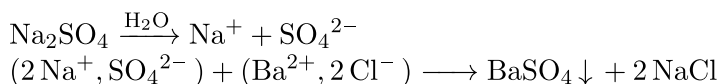
Suppose we want to solve the general quadratic equation $ax^2 + bx + c = 0$. If you isolate for x , you derive the well-known quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

The default distribution of Latex offers [over 10,000 symbols](#). So, you can easily typeset most equations:

$$\begin{aligned} |L_\beta(f) - L_\beta(g)| &= \int_a^b f(x) - g(x) d\beta \\ &= \sup_{P,T} \left\{ \sum_{i=1}^n f(t_i) - g(t_i) \Delta\beta \right\} \\ &= \sup_{P,T} \left\{ \sum_{i=1}^n \frac{\epsilon}{2(\beta(b) - \beta(a))} \Delta\beta \right\} \\ &= \frac{\epsilon}{2(\beta(b) - \beta(a))} (\beta(b) - \beta(a)) \\ &= \frac{\epsilon}{2} \end{aligned}$$

Chemical Equations:

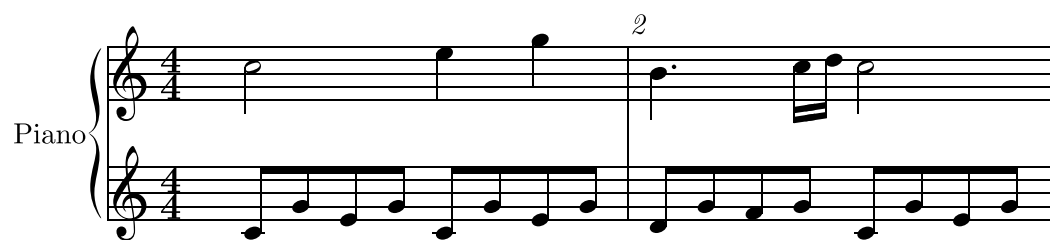


Credit: [Stack Exchange](#)

Phonetic Transcription:

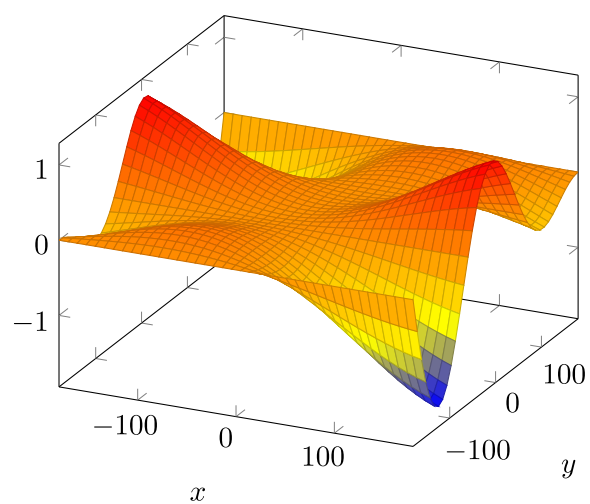
Gloss	‘innate’	‘intolerant’	‘impolite’	‘inconsistent’
UR	\ inert \	\ intalrnt \	\ impaʊlart \	\ inkənsistnt \
Assimilation	NA	NA	ɪmpaʊlart	ɪŋkənsistnt
SR	[ɪnert]	[ɪntalrnt]	[ɪmpaʊlart]	[ɪŋkənsistnt]

Music:



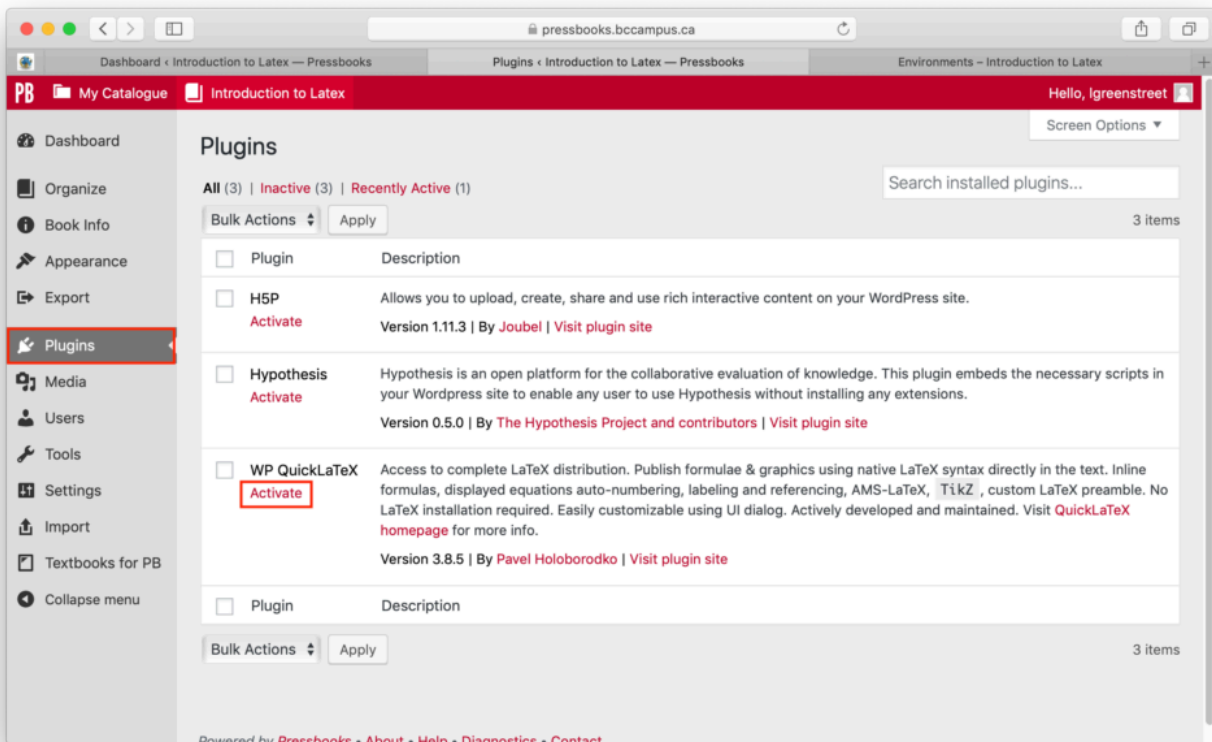
Credit: [Musixtex Documentation](#)

Plots:



Equations and Symbols

Pressbooks has a built-in Latex plug-in which allows you to create a Latex environment using shortcode. Before you start writing code, you'll need to select the plug-in by choosing Plugins from the lefthand sidebar and then choosing 'Activate' under the WP QuickLaTeX plugin.



Once you've activated the plug-in, you can start inserting code anywhere in your document using latex shortcode tags:

```
[latex] Some LaTeX code [/latex]
```

Equations

Commands

Latex allows you to quickly write mathematical equations by specifying characters using commands you can type using the keyboard. For example, we can write $ax^2 + bx + c = 0$ using the code

```
[latex] ax^2 + bx + c = 0 [/latex]
```

While some common math symbols like +/- and exponentiation have a single symbol on the keyboard, most commands start with “\” and then describe the command. An example of an equation that requires \ commands is $\cos^2 \theta + \sin^2 \theta = 1$, which requires commands for the sine, cosine, and angle:

```
[latex] \cos^2 \theta + \sin^2 \theta = 1 [/latex]
```

The table below shows some common “\” commands for symbols, in addition to the trigonometric functions and Greek letters used in the previous example.

	Algebra		Calculus		Logic
×	\times	lim	\lim	¬	\neg
÷	\div	∫	\int	→	\rightarrow
±	\pm	∇	\nabla	∃	\exists
≈	\approx	∂	\partial	∀	\forall
≡	\equiv	∑	\sum	∈	\in

Commands can also take arguments using curly brackets. For example, the fraction and square root in the quadratic formula both take arguments.

```
[latex] x = \frac{-b \pm \sqrt{b^2 + 4ac}}{2a} [/latex]
```

$$x = \frac{-b \pm \sqrt{b^2 + 4ac}}{2a}$$

Superscripts and subscripts are also considered arguments. Use `^{\}` to create a superscript. If, no brackets are used, only the first character will be part of the superscript.

```
[latex] e^{i \pi} = -1 [/latex]
```

$$e^{i\pi} = -1$$

Use `_{\}` to create a subscript. Again, if no brackets are used only the first character will be part of the subscript.

```
[latex] x_1 + x_2 = 2x_3 [/latex]
```

$$x_1 + x_2 = 2x_3$$

You can add a subscript and a superscript to the same item. The order of the subscript and superscript does not matter. For example:

```
[latex]\int_{a}^{b} u dv = u(x)v(x)|_{a}^{b} - \int_{a}^{b} v du [/latex]
```

$$\int_a^b u dv = u(x)v(x)|_a^b - \int_a^b v du$$

Latex is generally not sensitive to white space when typesetting math. Outside of Pressbooks, the commands `\;`, `\:`, `\,` explicitly add large, medium, and small spaces respectively. However, these commands do not seem to be compatible with the QuickLaTeX plugin. As a workaround, you can use the command `\hspace{ }` to add a specific amount of horizontal white space. For example, we can add a space between the functions and the differential variable:

```
[latex]\int_{a}^{b} u \hspace{2pt} dv = u(x)v(x)|_{a}^{b} - \int_{a}^{b} v \hspace{2pt} du [/latex]
```

$$\int_a^b u dv = u(x)v(x)|_a^b - \int_a^b v du$$

Errors

If a statement is not entered correctly, the equation may not display as expected or QuickLaTeX may display an error. Errors take practice to read. For example, if I were to try to write the quadratic formula as

```
[latex] x = \frac{-b \pm \sqrt{b^2 + 4ac}}{2a} [/latex]
```

I would get the error

```
*** QuickLaTeX cannot compile formula:
```

```
  x = \frac{-b \pm \sqrt{b^2 + 4ac}}{2a}
```

```
*** Error message:
```

```
File ended while scanning use of \frac .
```

```
Emergency stop.
```

Try to spot what is causing the error. The answer is given in the solutions to the exercises for this section.

Equation Practice Exercises

Below are a few commands to practice typesetting. To get started, create a book or make a private chapter in an existing book for practice and make sure that the QuickLaTeX plugin is activated (see the start of the section for instructions). Solutions are in the following section.

1. A circle of radius 1:

$$x^2 + y^2 = 1$$

2. Period of a simple harmonic oscillator:

$$T = 2\pi\sqrt{\frac{m}{k}}$$

3. The gravitational force between two masses:

$$F_g = \frac{Gm_1m_2}{r^2}$$

4. The solution to Basel problem:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

Hint: The symbols \sum is written `\sum` and ∞ is written `\infty`.

5. Determine what the error was in the equation for the quadratic formula shown at the end of the last section.

Equation Solutions

Below are solutions to the practice equations. There are multiple ways to express some of the equations, so your solutions do not need to exactly match the answers given.

1. $x^2 + y^2 = 1$

`[latex]x^2 + y^2 = 1[/latex]`

2. $T = 2\pi\sqrt{\frac{m}{k}}$

`[latex]T = 2 \pi \sqrt{\frac{m}{k}}[/latex]`

3. $F_g = \frac{Gm_1m_2}{r^2}$

`[latex]F_g = \frac{Gm_1 m_2}{r^2}[/latex]`

4. $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$

`[latex]\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}[/latex]`

If you want the sum to have the limits above and below, like you see in most textbooks, you can use the command `\displaystyle`.

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

`[latex]\displaystyle \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}[/latex]`

5. The equation is missing a bracket to close the numerator of the fraction, shown in bold below.

`[latex]x = \frac{-b \pm \sqrt{b^2 + 4ac}}{2a}[/latex]`

Environments

When you use the Latex shortcode in Pressbooks, you start in math mode. When writing Latex outside of WordPress, you need to put ‘\$’s around your equation to use math mode. It is helpful to remember this if you are looking at other tutorials or examples online, as their code might not work if you copy and paste it into Pressbooks without removing the ‘\$’s.

Latex supports a variety of environments to create equations, tables, figures, and more. Equations are particularly helpful in Pressbooks, as they keep track of numbering.

For example, we can typeset the double-angle trigonometric identities as:

(1) $\sin 2\theta = 2 \cos \theta \sin \theta$

(2) $\cos 2\theta = \cos^2 \theta - \sin^2 \theta$

```
[latex] \begin{equation} \sin 2 \theta = 2 \cos \theta \sin \theta \end{equation} [/latex]
[latex] \begin{equation} \cos 2 \theta = \cos^2 \theta - \sin^2 \theta \end{equation} [/latex]
```

If you type text in math mode, the spacing will look odd as Latex will try to interpret the words as an equation. For example:

We cantypesetthedouble – angletrigonometricidentitiesas :

```
[latex]We can typeset the double-angle trigonometric identities as:[/latex]
```

You can have Latex format text normally in math mode using the command `\text{}`.

$$\text{fraction} = \frac{\text{numerator}}{\text{denominator}}$$

```
[latex]\text{fraction} = \frac{\text{numerator}}{\text{denominator}} [/latex]
```

Latex also distinguishes between a math mode for typesetting text inline and a display format for stand-alone equations. In the solutions to the practice for commands, we saw that using `\displaystyle` switched Latex to using the mode for display, which looks nicer for large elements. Using the stand-alone mode will also make our fraction look nicer.

$$\text{fraction} = \frac{\text{numerator}}{\text{denominator}}$$

```
[latex]\displaystyle \text{fraction} = \frac{\text{numerator}}{\text{denominator}} [/latex]
```


In addition to `\displaystyle` and the `equation` environment, you will often see people surround an equation with ‘`$$`’ to begin and end a display environment.

You don’t need to use many of the other Latex environments in Pressbooks, as tables and figures can be added using the WYSIWYG editor. However, if you decide to write stand-alone Latex documents you will use other environments frequently.

Accents and Symbols

Latex's plain text commands are also convenient for typesetting accents. For example:

Celui qui ne connaît pas les langues étrangères ne sait rien sur lui-même.
– Goethe

```
[latex]\text{Celui qui ne connaît pas les langues \texttt{'etrang}\texttt{'eres ne  
sait rien sur lui-m}\texttt{^eme}. – Goethe} [/latex]
```

Note that in Pressbooks, you will need to surround the sentence by `\text{ }` for the text to format properly. The table below shows the Latex commands for some common accents.

Grave:	á	<code>\'a</code>	Acute:	à	<code>\`a</code>	Circumflex:	â	<code>\^a</code>
Umlaut:	ä	<code>\"a</code>	Tilde:	ã	<code>\~a</code>	Macron:	ā	<code>\=a</code>
Breve:	ă	<code>\u{a}</code>	Cedilla:	ç	<code>\c{a}</code>	Dot:	ȧ	<code>\.a</code>

Common symbols can also be typeset. Note that some symbols on the keyboard have a “`\`” command since the regular symbol has a special meaning in Latex. The symbols in the table below can be used in math and text mode.

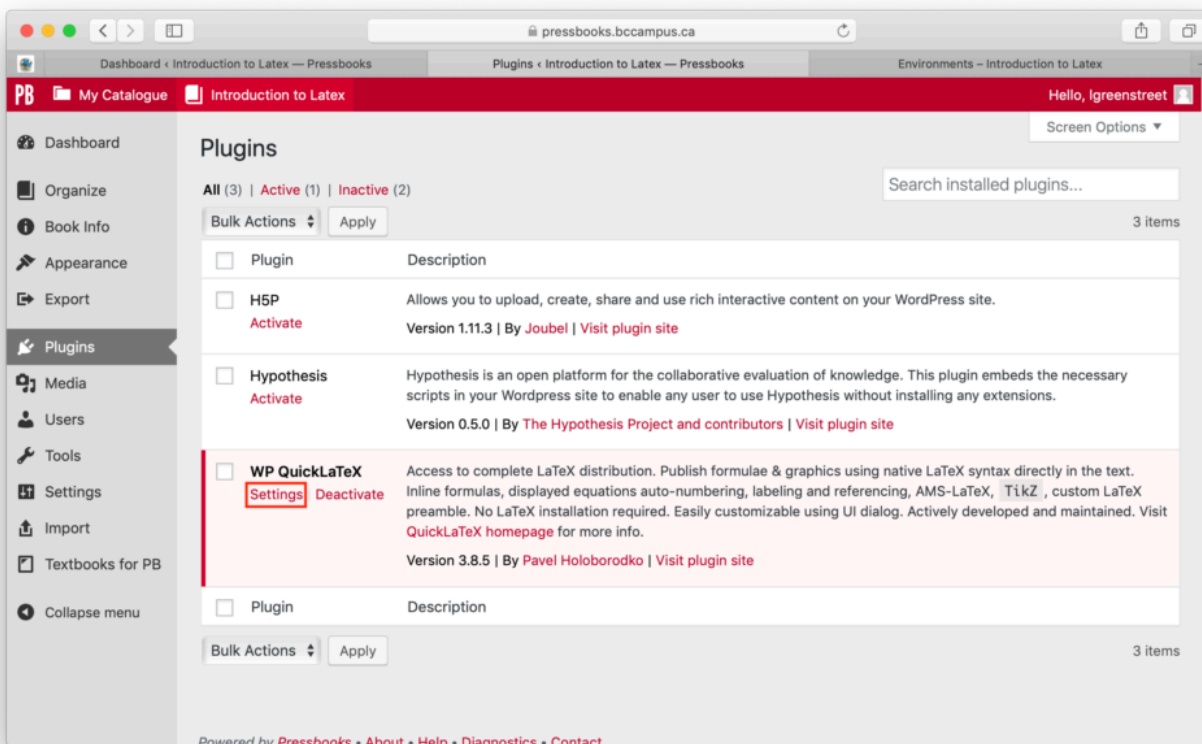
\$	<code>\\$</code>	†	<code>\dag</code>	©	<code>\C</code>
%	<code>\%</code>	‡	<code>\ddag</code>	™	<code>\texttt{trademark}</code>
#	<code>\#</code>	¶	<code>\P</code>	§	<code>\S</code>
{ }	<code>\{ \}</code>	...	<code>\dots</code>	✓	<code>\checkmark</code>

Additional symbols and alphabets can be typeset by using packages, which are covered in the next section.

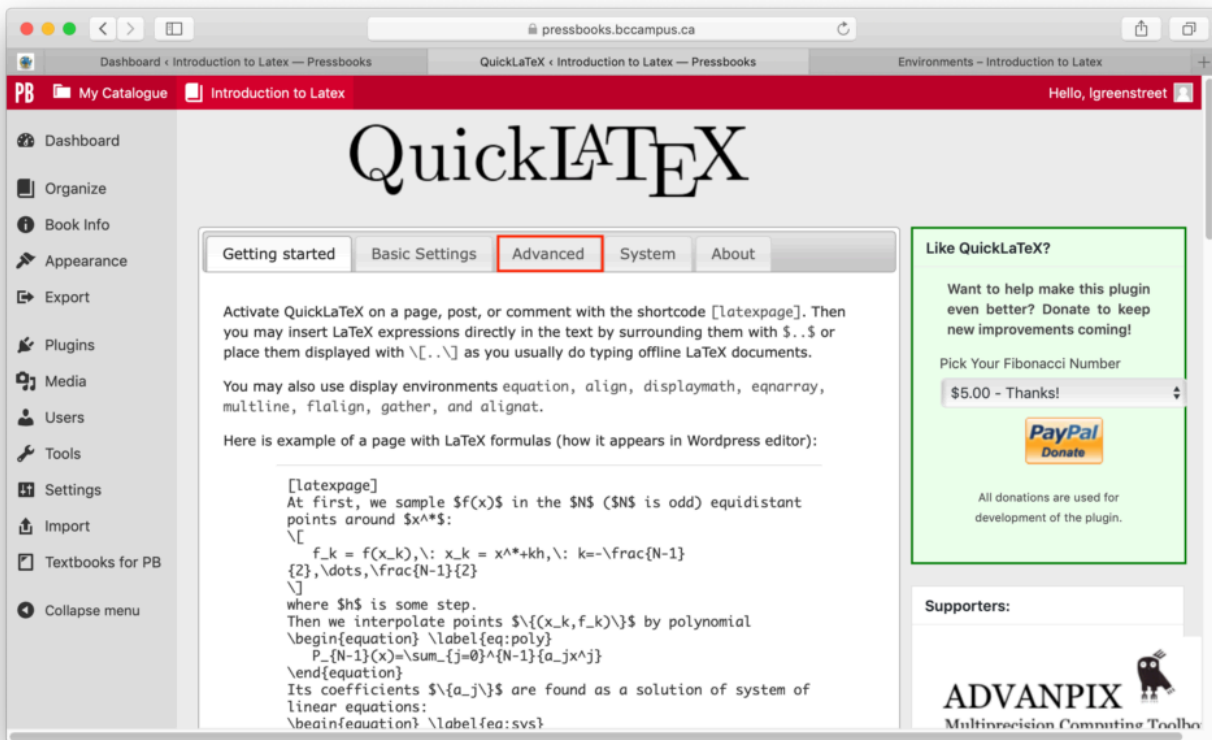
Packages

When using Latex in Pressbooks, you automatically have access to the basic Latex library and the American Mathematical Society (AMS) math package. If you are only using Latex to typeset equations, this will provide the needed functionality. However, Latex also allows users to develop and share their own packages which implement additional functionalities like typesetting non-latin languages or creating the plots and figures.

To be able to use a package when using Latex in Pressbooks, you will need to add the package in the settings for the Plugin. First, click on Plugins and choose ‘Settings’ under the WP QuickLaTeX editor.



Choose the ‘Advanced’ tab under settings and scroll down to the box where you can add commands.



Here you can configure your Latex setup. For now, we'll just add packages that will allow us to write phonetic transcriptions and create plots.

```
\usepackage{tipa}
\usepackage{tikz}
\usepackage{pgfplots}
```

pressbooks.bccampus.ca

Edit Chapter < Introduction to Latex — Pressbooks QuickLaTeX < Introduction to Latex — Pressbooks How do you insert an invisible character in HTML? - Quora

PB My Catalogue Introduction to Latex Hello, Igreenstreet

Dashboard

Organize

Book Info

Appearance

Export

Plugins

Media

Users

Tools

Settings

Import

Textbooks for PB

Collapse menu

This option activates LaTeX interpretation on all pages, regardless of the use of `[latexpage]`. This is useful for a site that is authored exclusively by LaTeX users. If this is turned on, all authors must be made aware that $\$$ is a special symbol, and that the standard dollar symbol must be typed $\$$.

Exclude \$.. \$ ☐ OFF

The use of the traditional TeX $\$$ conflicts with the non-LaTeX user's expectation of its meaning. This option allows us to turn its interpretation off, so standard dollar signs need no special treatment and will not confuse the non-mathematical user.

Please setup LaTeX preamble for the whole* website below.
You can define new commands and include additional packages as usual:

```
\usepackage{tikz}
\usepackage{pgfplots}
\usepackage{tipa}
```

*Global preamble can be overridden by `[preamble]` tag for particular equation.

[Update QuickLaTeX Settings »](#)

All donations are used for development of the plugin.










Supporters:

ADVANPIX
Multiprecision Computing Toolbox

More Symbols and Languages

Symbols

Latex can typeset over 14,000 symbols which can be found in their [comprehensive symbol list](#). If a package is needed to use a symbol, the package is specified in the guide. Below is a table of some symbols from the package fontawesome, including logos of popular software and websites. Note that the fontawesome symbols must be used inside of `\text{}`.

	<code>\faCreativeCommons</code>		<code>\faFilePDFO</code>		<code>\faBookmark</code>
	<code>\faCalendar</code>		<code>\faShareAlt</code>		<code>\faInbox</code>
	<code>\faGithub</code>		<code>\faFileWordO</code>		<code>\faFilePdfO</code>

Languages

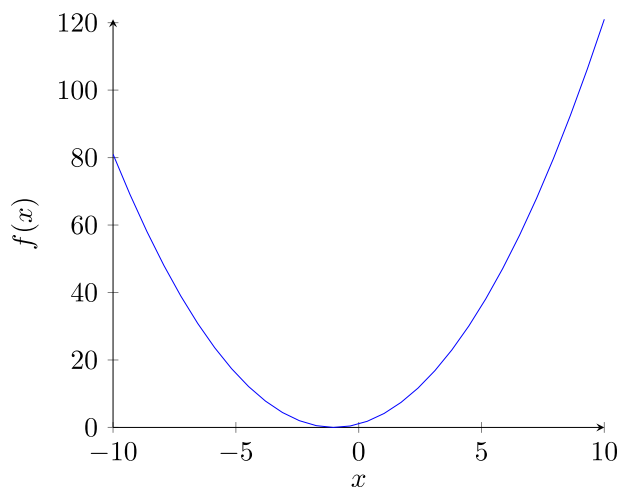
Latex was developed to typeset American English and makes a variety of choices in spacing, hyphenation, and formatting to conform to American English conventions. If you are typesetting a different language, you can use the [Babel](#) package to match typesetting conventions for that language. Babel can handle languages that use the Latin alphabet as well as other systems of characters including Arabic, traditional and simplified Chinese, and Japanese kanji. [Here](#) is an example of a document that uses both English and Japanese from the babel-japanese documentation.

Plots

When including equations, it is often helpful to include plots and figures as a visual learning aid. The Tikz package provides a variety of tools for creating figures. The PGFplots package can be combined with Tikz to create simple to code yet elegant plots.

2D Plots

To create a plot, invoke the plot environment and then specify a few parameters including the axes and function to plot.



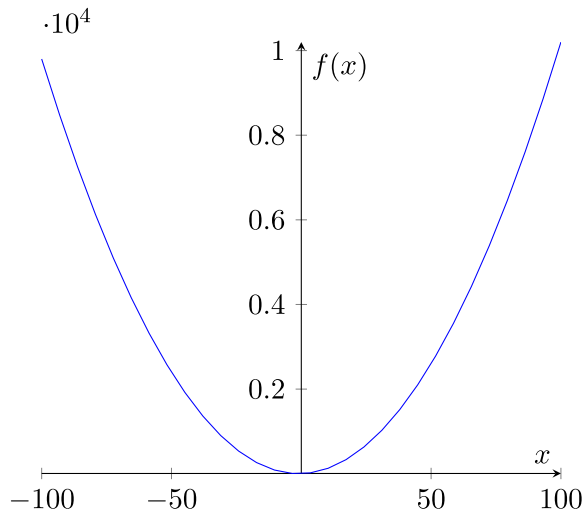
```
[latex]
\begin{tikzpicture}
\begin{axis}[
axis lines = left,
xlabel = $x$, ylabel = {$f(x)$}
\addplot [ domain=-10:10,
samples=30, color=blue]
{x^2 + 2*x + 1};
\end{axis}
\end{tikzpicture}[/latex]
```

Code like this can seem overwhelming, but it becomes easier to read if you parse it one line at a time.

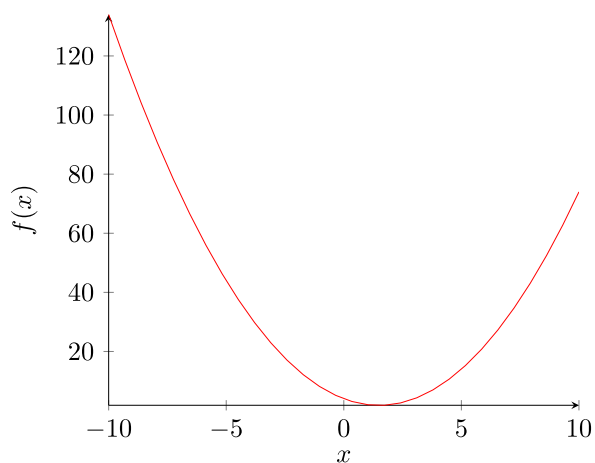
- `\begin{tikzpicture}` starts the tikzpicture environment. Tikzpicture is an environment often used for drawing figures with geometric shapes or plots.
- `\begin{axis}` starts configuring the axes. The '[' at the end of the line indicates that we are going to enter some parameters for the axes.
- `axis lines = left`, means that the axes will be along the bottom and on the lefthand side.
- `xlabel = x` and `ylabel = {$f(x)$}` specify that the x-axis should be labeled x and the y-axis should be labelled $f(x)$. The brackets around $f(x)$ ensure that the whole group is made into the label. The '\$'s means that the text will formatted like it is in math mode.
- The ']' ends the specifications for the axes.

- `\addplot` adds a new plot to the figure.
- `domain=-10:10, samples=30, color=blue` configures the line on the figure, specifying it should range from $x=-10$ to $x=10$, that the plot should use 30 line segments, and that the color of the line should be blue. As you increase the number of line segments, the line becomes smoother. However, using more line segments uses more memory and results in the page loading slower.
- `{x^2 + 2*x + 1};` is the equation that will be plotted.
- `\end{axis}` and `\end{tikzpicture}` end the environments that we started earlier.

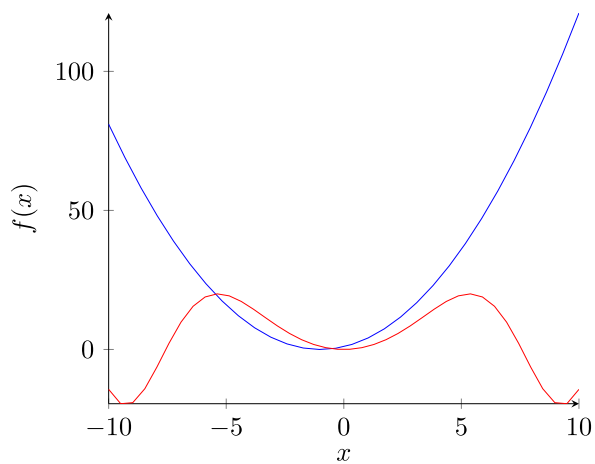
Understanding the code, we can modify the plot to show a wider range of values, modify the function, or plot an additional function on the same figure.



```
[latex]
\begin{tikzpicture}
\begin{axis}[axis lines=middle,
xlabel =  $x$ , ylabel =  $\{f(x)\}$ ]
\addplot [domain=-100:100,
samples=30, color=blue]
{x^2 + 2*x + 1};
\end{axis}
\end{tikzpicture}[/latex]
```



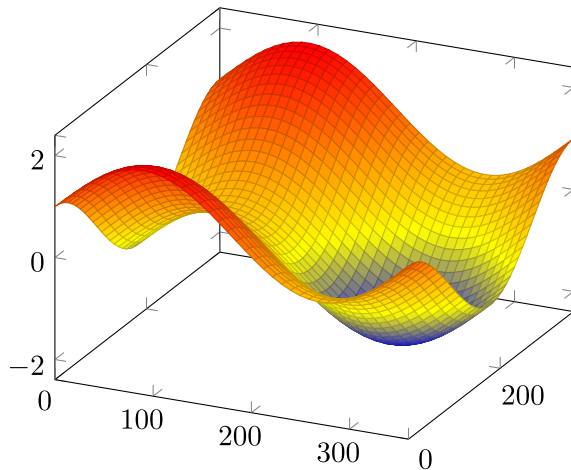
```
[latex]
\begin{tikzpicture}<
\begin{axis}[
axis lines = left,
xlabel =  $x$ ,
ylabel =  $\{f(x)\}$ ,]
\addplot [ domain=-10:10,
samples=30,
color=red]
{x^2 - 3*x + 4};
\end{axis}
\end{tikzpicture}[/latex]
```



```
[latex]
\begin{tikzpicture}
\begin{axis}[
axis lines = left,
xlabel =  $x$ ,
ylabel =  $\{f(x)\}$ ]
\addplot [ domain=-10:10,
samples=30,
color=blue]
{x^2 + 2*x + 1};
\addplot [ domain=-10:10,
samples=40,
color=red]
{20*sin(pi*x^2)};
\end{axis}
\end{tikzpicture}[/latex]
```

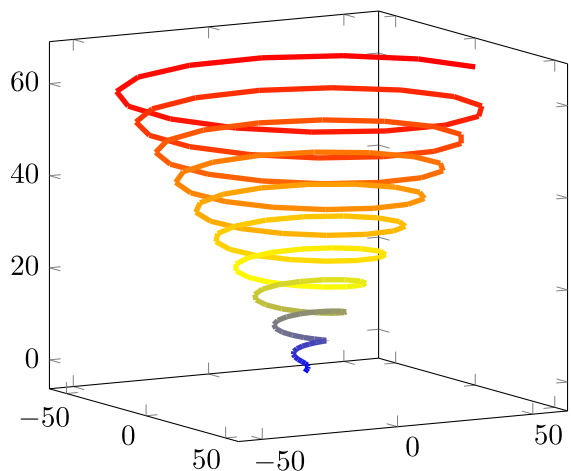
3D Plots

PGFPlots can also create 3D plots. The code for a 3D plot is similar to the 2D plot. Note that in this case we are setting the bounds for the axes in the parameters for `addplot3` with the argument ‘`domain=0:360`’.



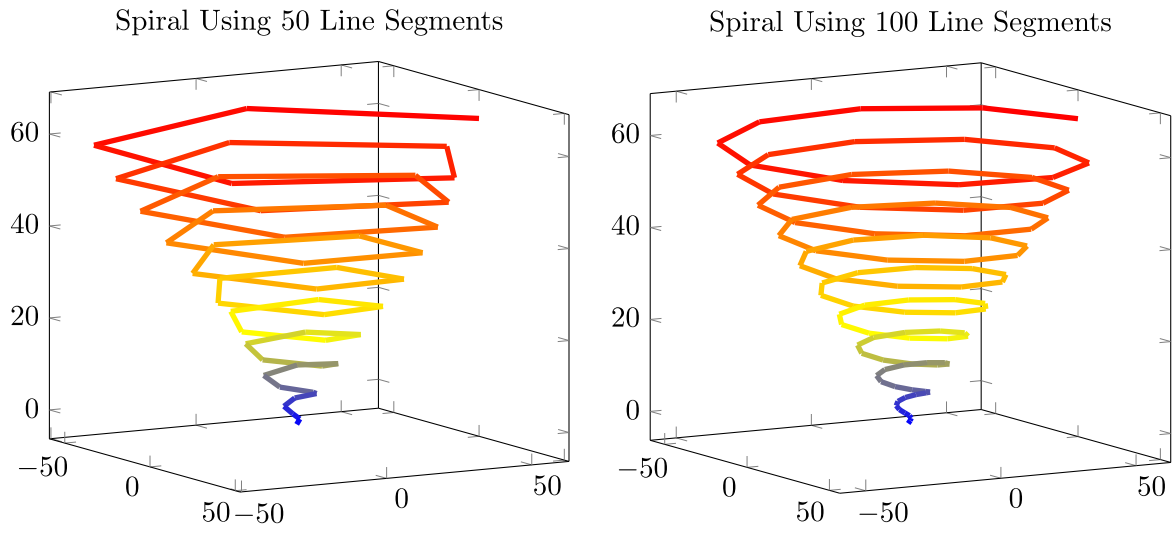
```
[latex]
\begin{tikzpicture}
\begin{axis}
\addplot3[surf,domain=0:360,
samples=40] {\cos(y) +
sin(x)};
\end{axis}
\end{tikzpicture}
[/latex]
```

Here is another example of a 3D plot where we use all three dimensions to create a spiral.



```
[latex]
\begin{tikzpicture}
\begin{axis}[view={60}{10}]
\addplot3[mesh,
domain=0:20*pi,
samples=150,
samples y=0, line
width=2pt]
({x*sin(deg(x))},
{x*cos(deg(x))},
{x});
\end{axis}
\end{tikzpicture}
[/latex]
```

- The two numbers given as parameters to the axes specify what angle you will see that plot at. The first argument specifies the rotation of the x-y plane and the second argument specifies the rotation in z.
- The `mesh` argument in `addplot` creates the color gradient.
- The `samples = 200` argument says tells the plot how many line segments to generate. Note that in this case we had to use 150 samples in comparison to the 30 samples used for the other examples. The plot below shows what the spiral looks like with 50 and 100 samples.



- The function specifies the spiral with a function for each coordinate in terms of a single variable. This type of function is called a parametric function.

The `y samples = 0` specifies that we are defining the line in terms of a single parameter. If you do not include this, Latex will generate an error. It is not obvious that you need this parameter, and the error doesn't make it obvious what to change:

*** QuickLaTeX cannot compile formula:

```
\begin{tikzpicture}
\begin{axis}[view={60}{10}]
\addplot3[mesh, domain=0:20*pi, samples=200, line width=2pt]
({(ln(x)/(ln(20*pi)))*sin(deg(x))},
{(ln(x)/(ln(20*pi)))*cos(deg(x))},
{x/(20*pi)});
\end{axis}
\end{tikzpicture}
```

When doing something new with a package, such as creating a new type of plot, it is often easiest to learn from examples. The package documentation is often a good source of examples and will also help you figure out what arguments can be used. [Here is the package documentation for PGFPlot](#). You can usually find the documentation of a package just by Googling the name of that package and documentation.

Plot Practice Exercises

Before starting these exercises, make sure you have added the `tikz` and `pgfplots` packages by following the instructions at the beginning of the section. Solutions are included after the exercises.

1. First, we'll make some modifications to the plot from the previous section to get more familiar with the code.
 1. Change the domain to between -2 and +2.
 2. Change the label on the y-axis to 'y'.
 3. Change the color of the line to green.
 4. Add a second quadratic function. You may need to adjust the domain to see both functions.

Here is the code for the original plot:

```
[latex]
\begin{tikzpicture}
\begin{axis}[axis lines = left, xlabel =  $x$ , ylabel =  $f(x)$ ]
\addplot [domain=-4:4, samples=100, color=blue]
{x^2 + 2*x + 1};
\end{axis}
\end{tikzpicture}
[/latex]
```

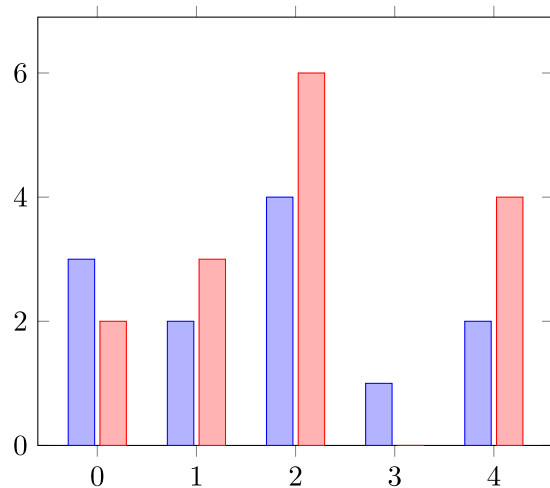
2. We'll now make the plot more informative by adding a title and legend. Add a title to your plot by adding `title=Your Title` to the axes parameters. Add a legend by adding `\legend{function 1, function 2}` after the `addplot` commands.
Hint: If your legend covers up an important part of your plot, add `legend pos=north/south east/west` to the axis parameters to move it. For example, to move the legend to the upper left use `legend pos=north west`.
3. Change the color of the plot to a color mapping using by adding the parameters `colorbar`, `point meta rel=axes wide`, `point meta=y` to the axis parameters, changing the parameters for each `addplot` to `mesh`, `marks=none`, and changing the `addplot` command to `addplot+`. What is the difference in color if you set `point meta res=per plot`?
Note: You need to change `addplot` to `addplot+` because there are parameters for the plot in the axis parameters. Without the `+`, the style changes would override the axis parameters.
4. Change the function used in the 3D surface plot from the last section. Try the function $1/x + 1/y$ or come up with your own. Note that in `pgfplots` you use `a/b` for division and `a*b` for multiplication rather than the regular LaTeX functions. Depending on the function you choose, you might want to change the domain to see an interesting range for the function. For

example, $\frac{1}{x} + \frac{1}{y}$ is most interesting near 0, so try a domain like -2:2.

Here is the code for the original plot:

```
[latex]
\begin{tikzpicture}
\begin{axis}
\addplot3[surf, domain=0:360, samples=40] {\cos(y) + \sin(x)};
\end{axis}
\end{tikzpicture}
[/latex]
```

5. PGFPlots can also create bar graphs. [Look at page 45 of the PGFplots documentation for some examples.](#) Copy the first example from the top of page 45 and use a second instance of addplot coordinates to add red bars beside the blue, like in the plot below.



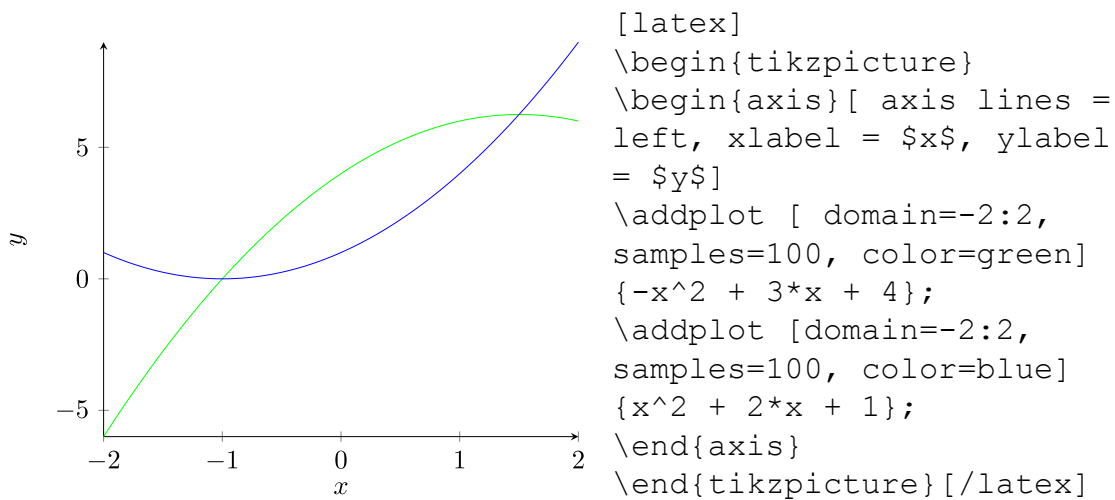
Hint: If you are having trouble having the bars start exactly on the x-axis, set the limits individually using `enlarge y limits={0.15,upper},`
`enlarge x limits=0.15` instead of `enlargelimits=0.15`.

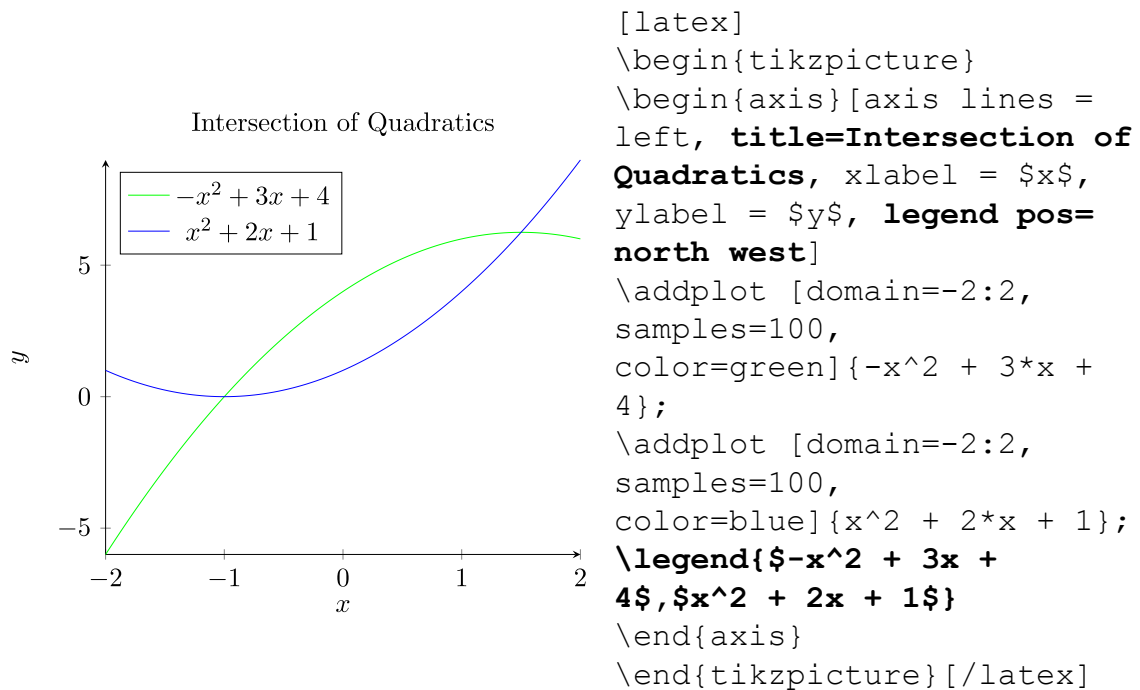
Plot Solutions

These answers show one possible solution to each problem. Other ways of solving the problems are also correct.

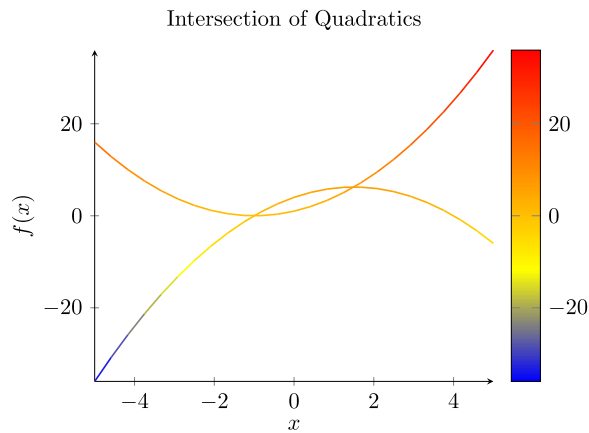
1. The answers below show what portion of the code to change.

1. `\addplot [domain=-2:2, samples=100, color=blue]`
2. `\begin{axis}[axis lines = left, xlabel = x , ylabel = y]`
3. `\addplot [domain=-2:2, samples=100, color=green]`
4. `\addplot [domain=-2:2, samples=100, color=blue]`
`{ $x^2 + 2x + 1$ };`

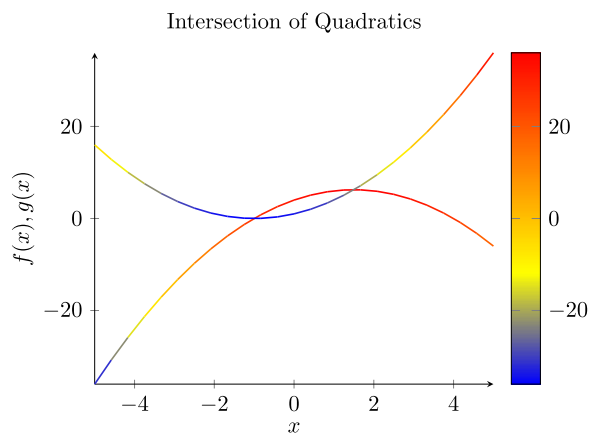




2. _____
3. The first plot below uses a colorbar with point meta rel=axes wide, which maps the colorbar to the whole y axis. The second plot uses point meta rel=per plot which maps the colorbar to each function on the plot.

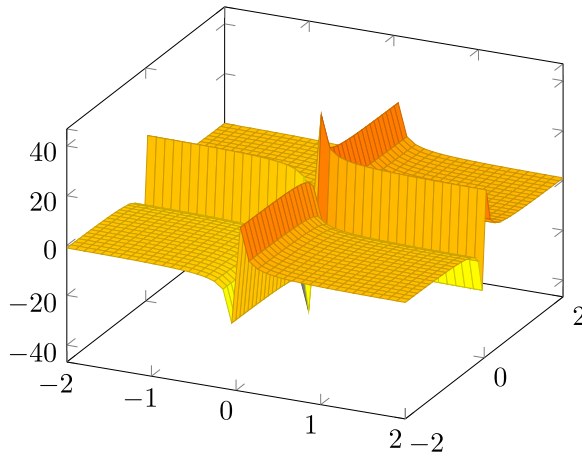


```
[latex]
\begin{tikzpicture}
\begin{axis}[axis
lines=left,
title=Intersection of
Quadratics, xlabel=$x$,
ylabel=$f(x)$, colorbar,
point meta rel=axes wide,
point meta=y]
\addplot+[mesh, thick,
mark=none]
{-x^2 + 3*x + 4};
\addplot+[mesh, thick,
mark=none]
{x^2 + 2*x + 1};
\end{axis}
\end{tikzpicture}[/latex]
```

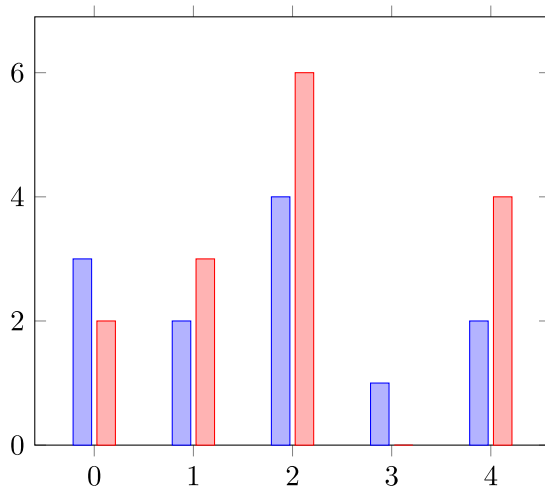


```
[latex]
\begin{tikzpicture}
\begin{axis}[axis
lines=left,
title=Intersection of
Quadratics, xlabel=$x$,
ylabel={$f(x), g(x)$},
colorbar, point meta
rel=per plot, point meta=y]
\addplot+[mesh, thick,
mark=none]
{-x^2 + 3*x + 4};
\addplot+[mesh, thick,
mark=none]
{x^2 + 2*x + 1};
\end{axis}
\end{tikzpicture}[/latex]
```

4. The plot below shows the behavior of $\frac{1}{x} + \frac{1}{y}$ near 0. Note that if you increase the number of samples from 50 to 100, you will exceed the memory allotted for QuickLatex. In general, increasing the number of samples will reduce the responsiveness of your page. So, it is not always desirable to enter larger sample numbers.



```
[latex]
\begin{tikzpicture}
\begin{axis}
\addplot3[surf,
domain=-2:2, samples=40]
{1/x + 1/y};
\end{axis}
\end{tikzpicture}
[/latex]
```



```
[latex]
\begin{tikzpicture}
\begin{axis}[ybar, enlarge y
limits={0.15,upper},
enlarge x limits=0.15]
\addplot plot coordinates
{(0,3) (1,2) (2,4) (3,1)
(4,2)};
\addplot plot coordinates
{(0,2) (1,3) (2,6) (3,0)
(4,4)};
\end{axis}
\end{tikzpicture}
[/latex]
```

5.

Resources

With the commands we've covered in the previous sections, you can now typeset mathematical equations, multi-lingual documents, and plots. The other examples shown in the introduction, such as music and chemical equations can all be typeset using the same principles. If you want to learn more, there are many freely available resources to help users learn Latex from basic tutorials to troubleshooting complex issues.

Below are some resources that are helpful for new Latex users. If you are having difficulty with a specific problem, I suggest Googling for it, as you can often find a detailed answer to your specific question.

Resources

- Rice University has a helpful guide for [mathematical symbols and functions](#).
- The [CTAN LaTeX Comprehensive Symbol List](#) specifies the 14,000+ symbols available in Latex.
- Dr. David Woods of Trinity College London has written a helpful [summary sheet for document formatting commands](#).
- [Detexify](#) is a website that lets you draw a symbol and suggests Latex commands, which is helpful when you don't know the name of a symbol.
- Some helpful packages include:
 - [Font Awesome: Web Icons/Symbols](#)
 - [Babel: Typesetting for Non-English Languages](#)
 - [TIPA](#): IPA for Linguistic Transcription
 - [PGFPlots](#): Plotting and Charts
 - [MusiXTeX](#): Music Typesetting
 - [mhChem](#): Chemical Equations
- There are a variety of Latex communities on the web that are happy to answer questions. Usually you can find the answer to a question just by Googling. Usually you'll find the answer on [LaTeX Stack Exchange](#) or [Latex.org](#).

Latex Outside of Pressbooks

Outside of Pressbooks, Latex is commonly used for journal papers, theses, and dissertations. In addition to handling equations, plots, and multi-lingual documents, Latex is popular for academic writing because Latex elegantly integrates sectioning documents and creating tables of contents, Latex supports BibTex a free reference manager to create easy and flexible bibliographies, and Latex allows you to precisely specify almost any aspect of the document format.

Choosing an Editor

If you are interested in typesetting stand-alone Latex documents, there are a variety of free Latex editors. They range from providing a word-processor style interface to programs which look more like an editor for code. [Here](#) is a helpful Quora post which covers many popular editors and their tradeoffs. While the title says the editors are for Windows, most are also available for Mac. In particular, I'll highlight Overleaf, which is an online editor. Overleaf is free and [offers free templates](#), [well-written tutorials](#), and options for online collaboration.