Final Report Title

Author 1
Undergraduate Student
Dept. of Mathematics
UMass Dartmouth
Dartmouth MA 02747
Email: username@umassd.edu

Author 2
Undergraduate Student
Dept. of Physics
UMass Dartmouth
Dartmouth MA 02747
Email: username@umassd.edu

Abstract

Write abstract here. This is a modified version of Brent W. Woodruff's LaTeX tutorial.

1 Problem Formulation

First we'll start of with some plain text. Pretty plain, right? You'll notice that this paragraph is not indented and sometimes you don't always want that. In fact, there are quite a few times LaTeX will put things everywhere but where you want them. Learn it and love it, because the equation formatting makes it *so* much better than a WYSIWYG word processor.

This new paragraph is indented. I'll try to fill up the rest of this line so it looks more like a paragraph. You can also do subsections as well as sections, like so.

1.1 This is a subsection

Write subsection here.

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{1}$$

2 Methodology

Simply put, LaTeX can format equations in tons of ways using the amsmath and amsfonts package. You can do inline equations like $y = x^2$ and even more complicated ones like $u(x,0) = \sin(x)$ and $f(x) = \frac{\sqrt{12-x}}{x^3-3}$. If you want an equation centered on a line by itself you can do this:

$$c(x)\rho(x)\frac{du}{dt}(x,t) = \frac{d}{dx}[K_0(x)\frac{du}{dx}(x,t)]$$

and if you want it to have a number, use the equation environment

$$\int_0^1 e^x dx = e - 1. \tag{2}$$

Now I can reference the above equation like so: Equation (2) was made up on the spot.

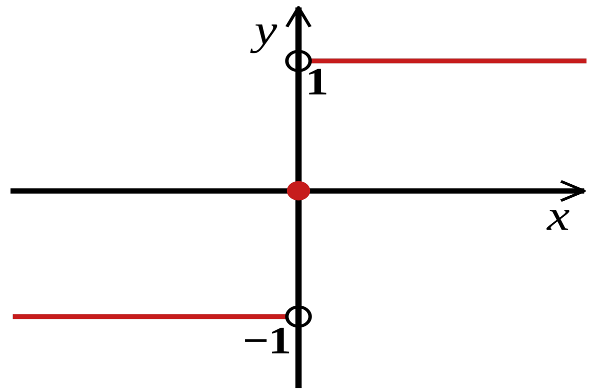


Figure 1: Signum function

3 Numerical Results

LaTeX can process a number of different graphics formats to include in files. The most common format is EPS. It is important to state that the images probably won't come up in the file where you expect them. The idea here is to just put the image directly after the text where it is supposed to show, hope for the best, and then consult LaTeX gurus if you need them somewhere else. Figure placement should probably be the *last* thing you worry about.

And like equations, you can reference figures. Figure 1 is the picture of a signum function.

Table example

		ABC		
		1	2	3
A	4	10	12	14
B	6	12	12	14
C	8	14	14	14

Show your numerical results here. An example of inserting a stand alone graph can be seen in Figure 2.

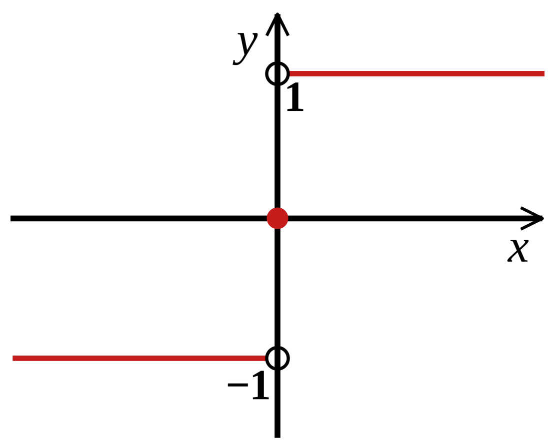
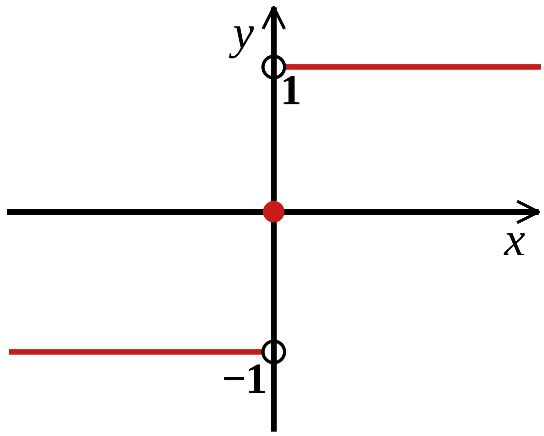


Figure 2: Caption for graphic

An example of inserting two graphs with one caption can be seen in Figure 3.

An example of inserting a stand alone graph with a caption on the left can be seen in Figure 4.

An example of inserting two graphs with two different captions can be seen in Figure 5 and Figure 6.



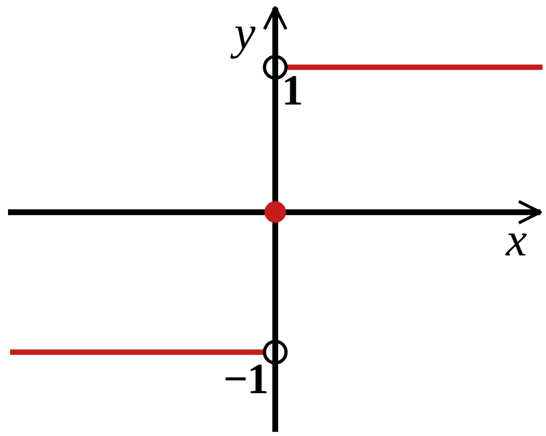
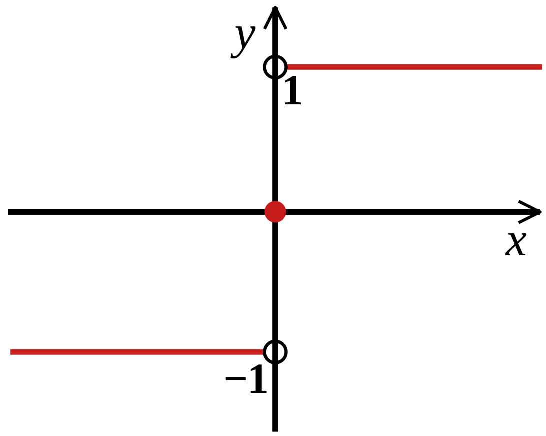


Figure 3: Caption for graphic

Figure 4: Caption for graphic



An example of inserting a graph and a table side by side with two different captions can be seen in Table 1 and Table 7.

n	$\lvert \cdot \rvert_\infty$	$t_\lambda(sec)$
5	1.802765996014689e-01	0.001410
10	3.699315545079440e-02	0.003316
20	3.626362118354209e-03	0.014435
30	4.462425364284428e-04	0.036919
40	5.832149101614448e-05	0.080982
50	8.373242851966722e-06	0.152643
60	1.294515106553540e-06	0.256228

Table 1: Put caption for table here

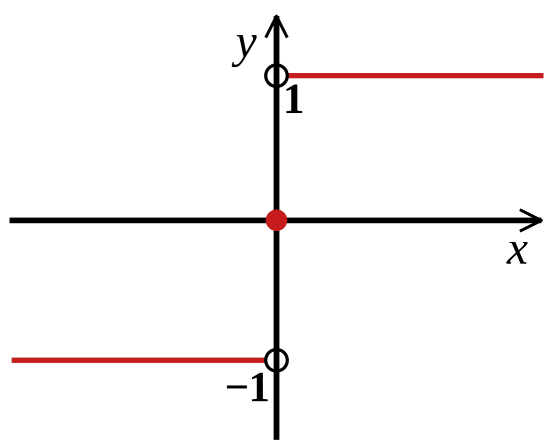


Figure 7: Put caption here

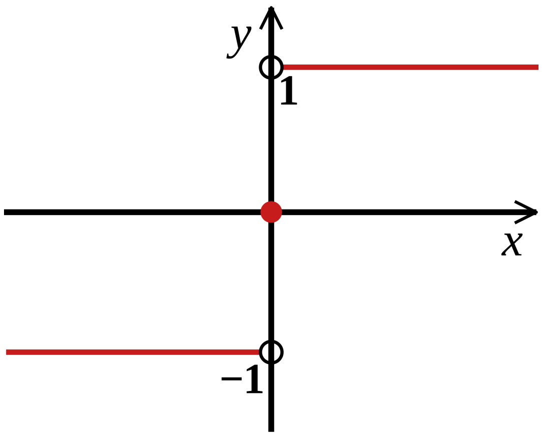


Figure 5: Caption for graphic

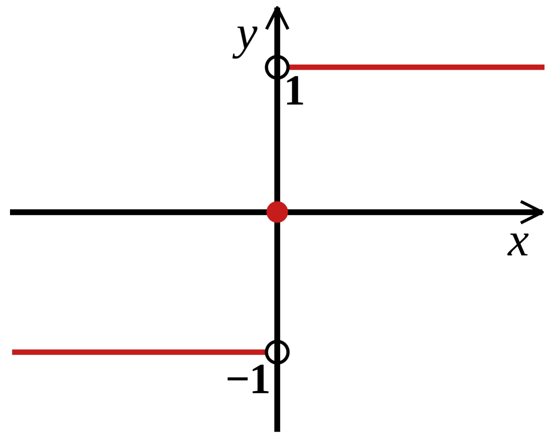


Figure 6: Caption for graphic

4 References using BibTeX

All your references are stored in .bib files. In this example, let say myref.bib. The combination of BibTeX and MathSciNet is very convenient. Once you master them, you can't go back. Now, I cite one of references [1].

5 Appendix

Maybe you want to include MATLAB codes or Python codes here.

```
import numpy as np

def incmatrix(genl1, genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int)  #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

                if M is None:
                    M = np.copy(VT)
                else:
                    M = np.concatenate((M, VT), 1)

                VT = np.zeros((n*m,1), int)

    return M
```

6 Acknoledgements

Thank whoever help you in this project including the funding agency if you are funded.

References

[1] Nicholas J. Higham. *Handbook of writing for the mathematical sciences*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1993.