Example (pandoc) Markdown to Latex PDF Template

Derek Harter
Texas A&M University - Commerce

December 11, 2021

Abstract

This is the abstract. This is only an example abstract. It consists of two paragraphs.

In this template we demonstrate how to use pweave and pandoc in order to create a publication ready PDF file. This template support LaTeX markup for BibTex bibliographies and mathematical notation. In addition the use of pweave allows us to include literate programming concepts, such as running python code in document and getting the results.

Contents

List of Figures

List of Tables

1 Introduction

This document template serves as an example of creating a PDF publication ready output document from a pandoc markdown plain text file. The pandoc markdown is weaved into a LaTeX markup file, then we use latex or pdflatex to create the final camera ready pdf file. This template demonstrates using LaTeX bibliography, math markup, figures, tables and a table of contents. We also demonstrate some literate programming examples with embedded python code chunks displaying output and figures.

2 Citations

Example of using BibTex bibliographies and citation system. The bibliography is included/generated at the end of the document. We show an example of formatting using an APA like style.

If you want to do a normal parenthetical citation of a book or journal article, (the most common kind) you would just cite the article keys (Cullity, 1972; Gupta & Senturia, 1997; Zhang et al., 1999). If you are referencing the author names as part of the discussion, for example, in the work of Zhang et al. (1999), the authors show that an ultrathin low-temperature channel is possible using Poly-Si. Another somewhat common variation is to reference a particular page like (Cullity, 1972, p. 99). Or also when you are talking about an example (e.g. Gupta & Senturia, 1997, p. 22).

The bibliography included in this template is from a standard IEEE publication latex template. There are many examples of more atypical citations like book with editions and and editor (Metev & Veiko, 1998), a chapter in a book collection (Rose, 1988), or a electronic source with a url (Valloppillil & Ross, 1998).

One suggestion, now adays I often simply search for an article I need to reference in Google Scholar. If you find the reference, Google Scholar has a link that will give you a BibTex bibliography citation entry, that you can simply copy and paste into your .bib file, saving a lot of time and almost always ensuring you get the most accuracte citation information.

3 Figures and Tables

We can insert an inline image that will automatically be given a figure number and provide a caption using markdown.

Later on in the document we show using pweave to generate and insert figures from code.

We can also create simple tables using the markdown table formats like this.

Table 1: An example simple markdown table

Right	Left	Center	Default
12	12	12	12
123	123	123	123
1	1	1	1

One reference for the different kinds of pandoc markdown tables is the original Pandoc manual

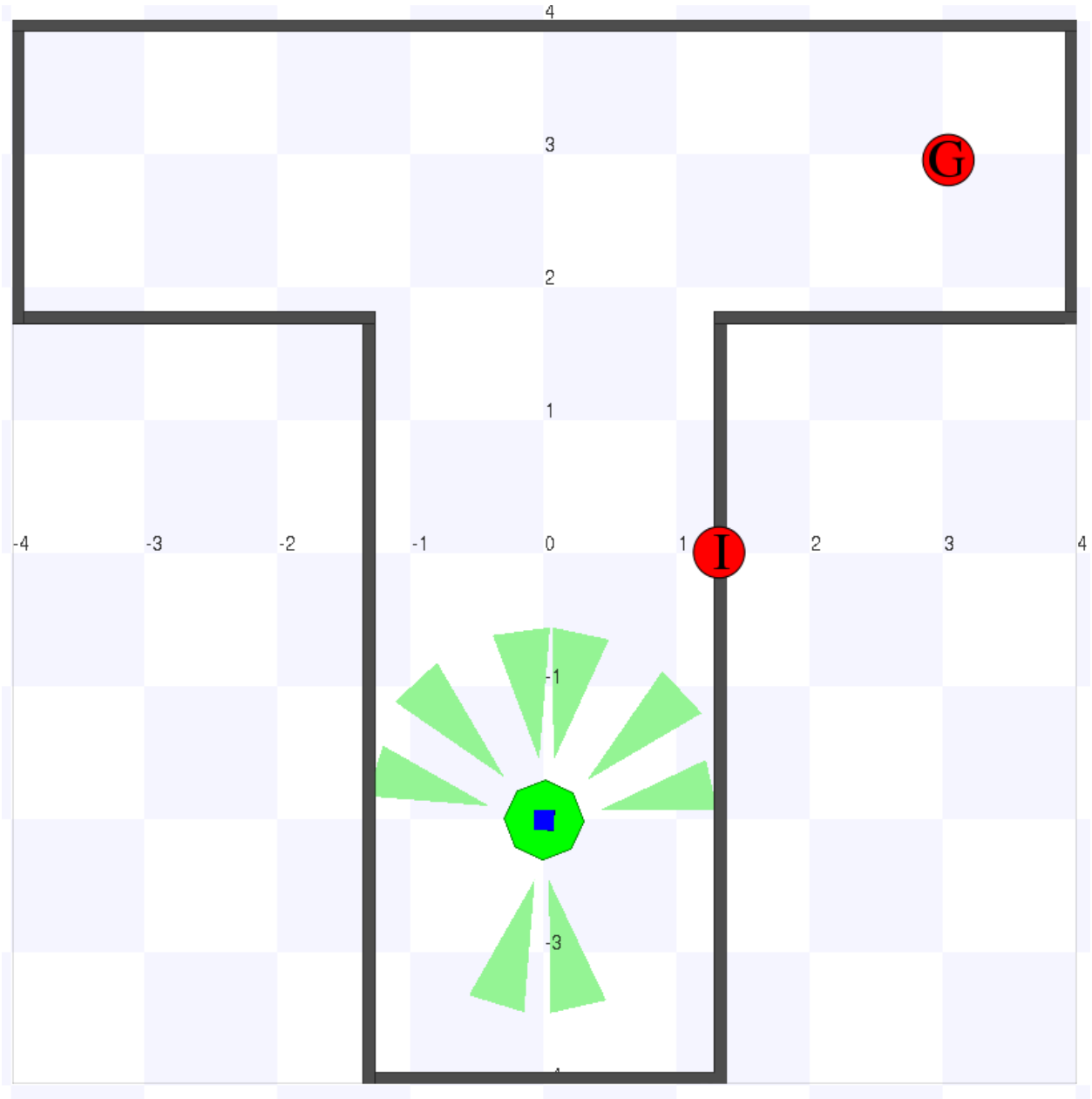

Figure 1: This is an example figure. All figures are in the figures subdirectory

If more complex tables are needed, you can fall back to using LaTeX markup to get more features for needed tables.

4 Example Document Content

This an example of a script that can be published using Pweave. The script can be executed normally using Python or published to HTML with Pweave Text is written in markdown in lines starting with "#" and code is executed and results are included in the published document. The concept is similar to publishing documents with MATLAB or using stitch with Knitr.

Notice that you don't need to define chunk options (see Pweave docs), but you do need one line of whitespace between text and code. If you want to define options you can do it on using a line starting with #+. just before code e.g. #+ term=True, caption=Fancy plots. If you're viewing the HTML version have a look at the source to see the markup.

The code and text below comes mostly from my blog post FIR design with SciPy, but I've updated it to reflect new features in SciPy.

5 FIR Filter Design

We'll implement lowpass, highpass and bandpass FIR filters. If you want to read more about DSP I highly recommend The Scientist and Engineer's Guide to Digital Signal Processing which is freely available online.

5.1 Functions for frequency, phase, impulse and step response

Let's first define functions to plot filter properties.

[mathescape, fontsize=, xleftmargin=0.5em]python import matplotlib.pyplot as plt import numpy as np import scipy.signal as signal

Plot frequency and phase response def mfreqz(b,a=1): w,h = signal.freqz(b,a) h$_dB$ = 20 $*$ $np.log10(abs(h))plt.subplot(211)plt.plot(w/max(w), h_dB)plt.ylim(-150, 5)plt.ylabel('Magnitude(db)')plt.xlabel(r'Normaliz$
plt.title(r'Frequency response') plt.subplot(212) h$_Phase$ = $np.unwrap(np.arctan2(np.imag(h), np.real(h)))plt.plot(w/max(w)$
plt.title(r'Phase response') $_=plt.subplots_adjust(hspace = 1)$

Plot step and impulse response def impz(b,a=1): l = len(b) impulse = np.repeat(0.,l); impulse[0] =1. x = np.arange(0,l) response = signal.lfilter(b,a,impulse) plt.subplot(211) plt.stem(x, response, use$_linecollection$ = $True)plt.ylabel('Amplitude')plt.xlabel(r'n(samples)')plt.title(r'Impulseresponse')plt.subplot(212)step$ = $np.cumsum(response)plt.stem(x, step, use_linecollection = True)plt.ylabel('Amplitude')plt.xlabel(r'n(samples)')plt.title(r'S$
1)

5.2 Lowpass FIR filter

Designing a lowpass FIR filter is very simple to do with SciPy, all you need to do is to define the window length, cut off frequency and the window.

The Hamming window is defined as: $w(n) = \alpha - \beta \cos \frac{2\pi n}{N-1}$, where $\alpha = 0.54$ and $\beta = 0.46$

The next code chunk is executed in term mode, see the Python script for syntax. Notice also that Pweave can now catch multiple figures/code chunk.

[mathescape, fontsize=, xleftmargin=0.5em]python n = 61 a = signal.firwin(n, cutoff = 0.3, window = "hamming") Frequency and phase response mfreqz(a)

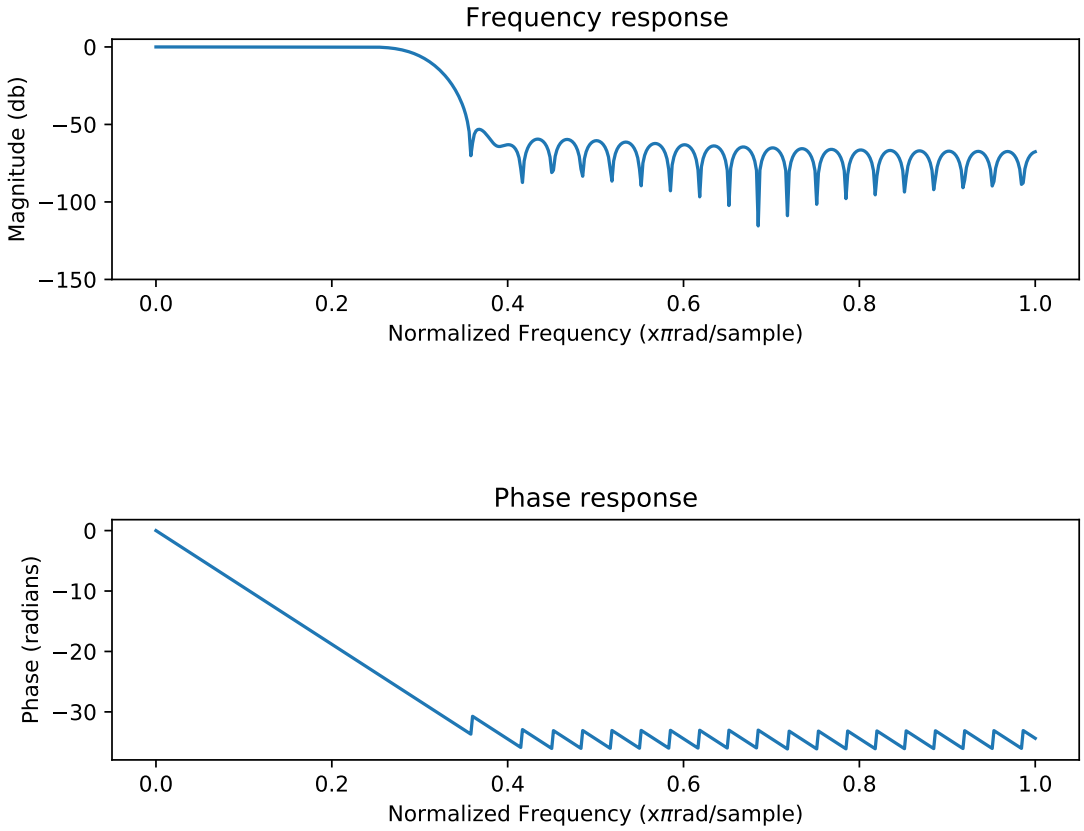

Figure 2: Frequency and Phase Response

[mathescape, fontsize=, xleftmargin=0.5em]python n = 61 a = signal.firwin(n, cutoff = 0.3, window = "hamming") Impulse and step response impz(a)

5.3 Highpass FIR Filter

Let's define a highpass FIR filter, if you compare to original blog post you'll notice that it has become easier since 2009. You don't need to do ' spectral inversion "manually" anymore!

[mathescape, fontsize=, xleftmargin=0.5em]python n = 101 a = signal.firwin(n, cutoff = 0.3, window = "hanning", pass$_zero = False)mfreqz(a)$

5.4 Bandpass FIR filter

Notice that the plot has a caption defined in code chunk options.

[mathescape, fontsize=, xleftmargin=0.5em]python n = 1001 a = signal.firwin(n, cutoff = [0.2, 0.5], window = 'blackmanharris', pass$_zero = False)mfreqz(a)$

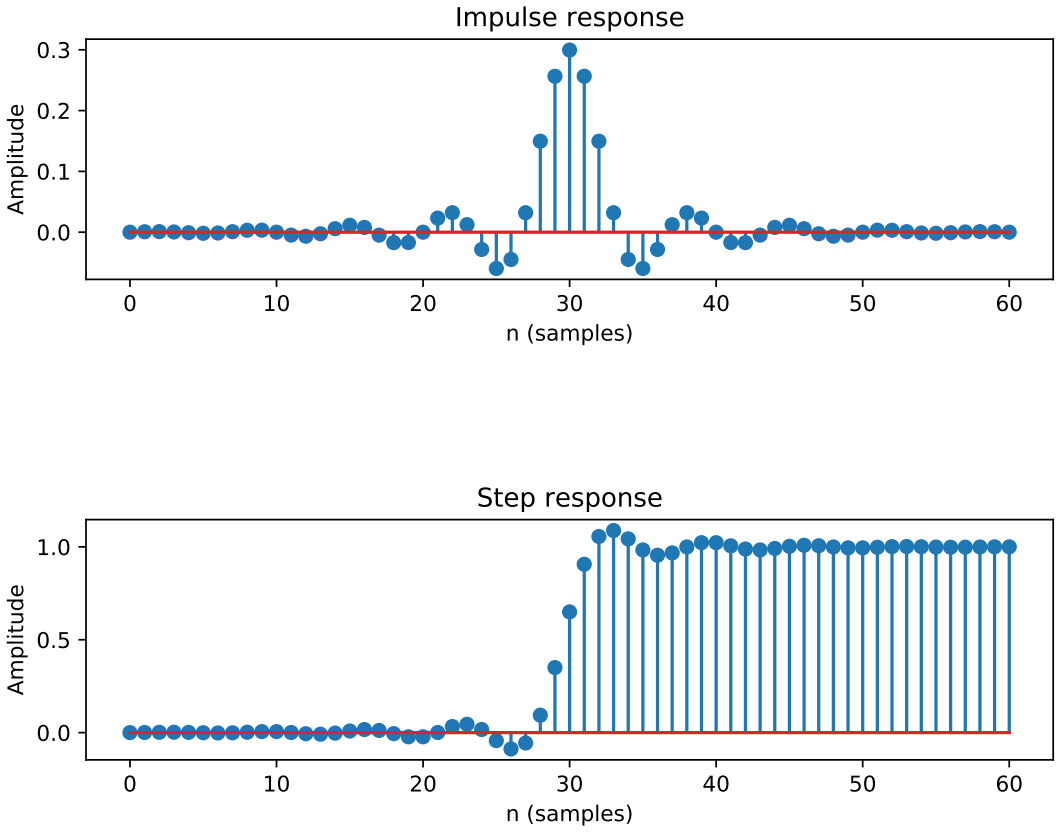

Figure 3: Impulse and Step Response

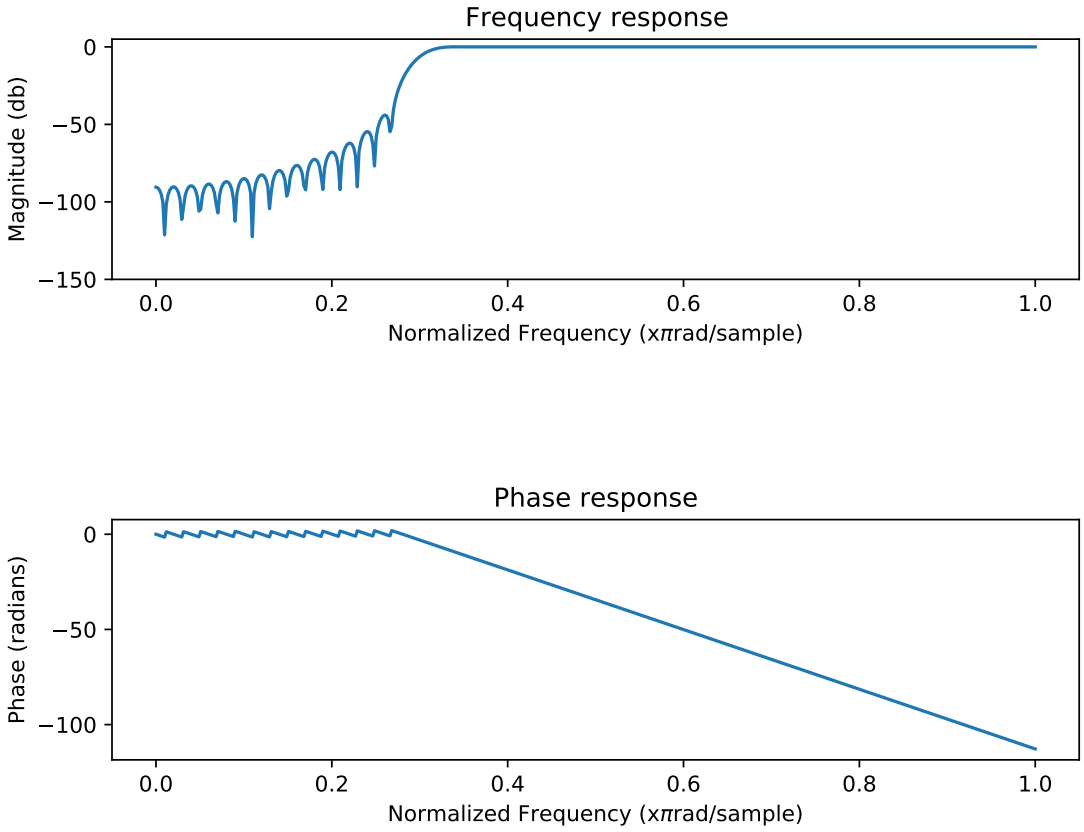

Figure 4: Highpass FIR filter.

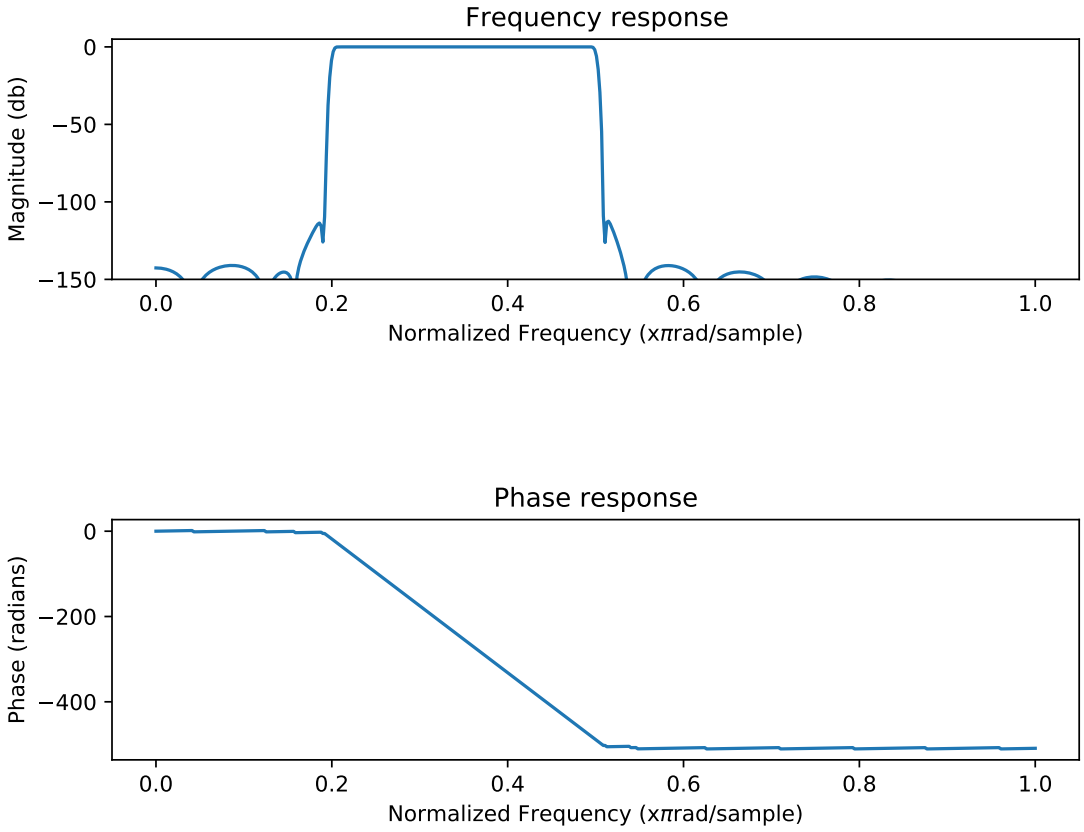

Figure 5: Bandpass FIR filter

6 Test C Code Block

Code blocks other than Python cannot be executed, but will be displayed and syntax highlighted.

```
int main()
{
  int x = 5;
  int y = 7;

  printf("x + y = %d", x + y);
}

void my_other_function(int x, char c, float f)
{
  float f = 0.1234;
  return Null;

  if (x == 5)
  {
    cout << "Hello world" << endl;
    f = 3.14159;
  }
}
```

7 Test Python Code Block and Execution

[mathescape, fontsize=, xleftmargin=0.5em]python import numpy as np import matplotlib.pyplot as plt

def f(N): x = np.linspace(-2.0 * np.pi, 2.0 * np.pi, N) y = np.cos(x) + 2 * np.sin(0.5 * x)

return (x, y)

x,y = f(100) title = "This is my plot" plt.plot(x,y, label=r'cos(x) sin($\frac{x}{2}$)') plt.plot(x, y) plt.xlabel('x') plt.ylabel(r'$y = \cos(x)$ sin($\frac{x}{2}$)') plt.title(title) plt.legend()

Lets test seaborn plots.

[mathescape, fontsize=, xleftmargin=0.5em]python import seaborn as sns import pandas as pd

mean, cov = [0, 1], [(1, .5), (.5, 1)] data = np.random.multivariate$_n ormal(mean, cov, 200) df = pd.DataFrame(data, columns = [$"$x$"$, $"$y$"$])$

sns.jointplot(x="x", y="y", data=df, kind="kde");

8 Test Functions and labeling

My first function

$$f(x) = \int_{-\infty}^{\infty} \bar{x} - x^2 + 3y\beta \; dx \tag{1}$$

My second great formula

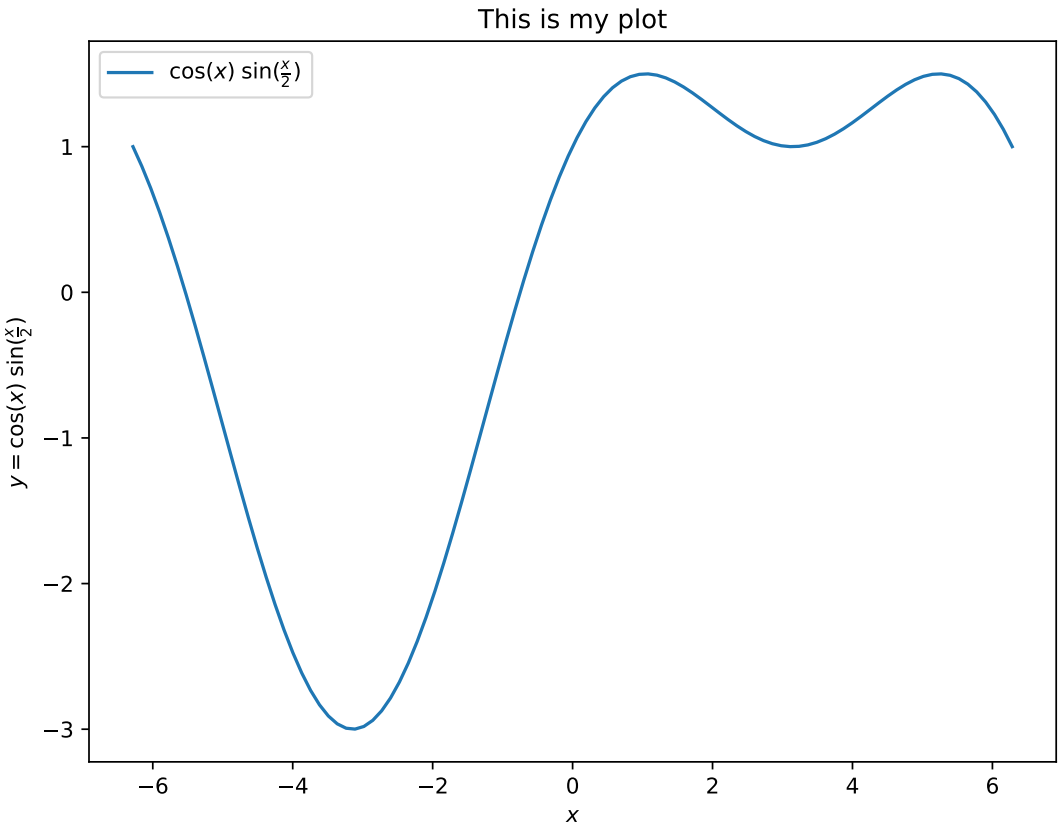

Figure 6: Matplotlib plot of sin and cos

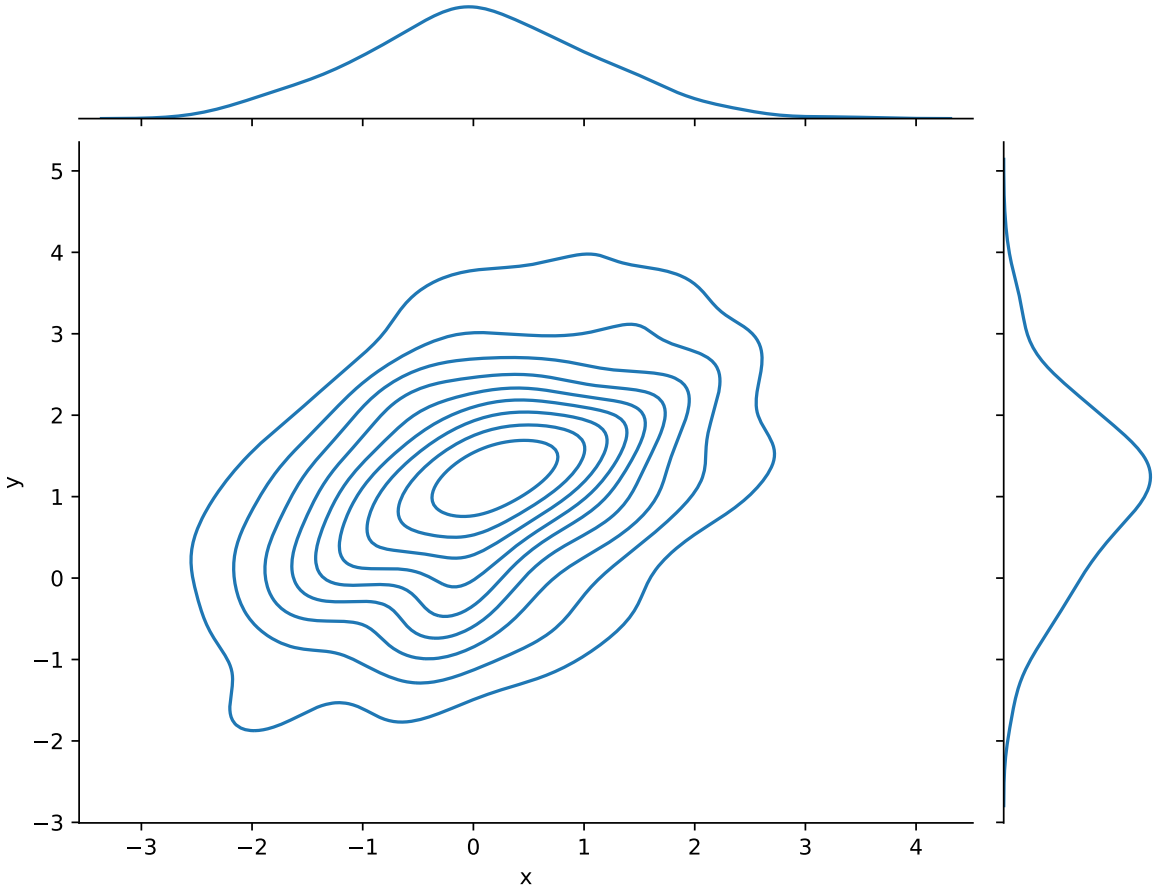

Figure 7: Seaborn joint contour plot

$$g(x) = \sum_{x=1}^{\infty} \frac{x^2}{x!} 2x \tag{2}$$

References

Cullity, B. D. (1972). *Introduction to magnetic materials*. Addison-Wesley.

Gupta, R. K., & Senturia, S. D. (1997). Pull-in time dynamics as a measure of absolute pressure. *Proc. IEEE International Workshop on Microelectromechanical Systems (MEMS'97)*, 290–294.

Metev, S. M., & Veiko, V. P. (1998). *Laser assisted microtechnology* (R. M. Osgood Jr., Ed.; Second). Springer-Verlag.

Rose, H. E. (1988). A course in number theory. Oxford Univ. Press.

Valloppillil, V., & Ross, K. W. (1998). *Cache array routing protocol v1.1*. http://ds1.internic.net/internet-drafts/draft-vinod-carp-v1-03.txt

Zhang, S., Zhu, C., Sin, J. K. O., & Mok, P. K. T. (1999). A novel ultrathin elevated channel low-temperature poly-Si TFT. *Phys. Rev. B.*, *20*, 569–571.