

week01-00-demo-notebook

January 14, 2025

1 [George McNinch](#) Math 87 - Spring 2025

2 § Week 1 - Demo notebook

This notebook is intended to illustrate some features of [jupyter notebooks](#).

For the most part, materials for this course will be presented in the form of notebooks like this one.

You can view notebooks on [colab](#) or you can install some software on your own computer and view/edit notebooks there; there is discussion of installation on the [course web site](#) – in particular, the “[python & jupyter resources information](#)” page.

3 Mathematical typesetting

Jupyter notebooks contain text (like this) but also they can contain *mathematical symbols*; for example:

$$\int_{-\infty}^{\infty} f(x)dx = 0 \quad \text{or} \quad \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$$

If you are interested in what is going on “under the hood”, text is entered using `markdown` syntax which is converted to `html` and displayed in your browser.

4 Markdown and MathJax

You can see the `markdown` underlying what you are reading now by “double clicking”

You can read about [markdown syntax starting from here](#), though there shouldn’t be a need for you to write markdown for our course.

The mathematical typeset appears thanks to an *extension* to markdown/html called [MathJax](#); again, you don’t really need to know details about mathjax. But it is worth knowing that the syntax is the same as `LaTeX`.

5 Under the hood?

If you are *curious*, you can see the markdown/mathjax that was used to create the text you are reading currently.

First, notice that there is a “boxed region” containing this text - in the parlance of `jupyter` notebooks, that region is called a `cell`. If you click with the mouse pointer on this text, that `cell` receives `focus`.

Now that you’ve focussed on this cell, you can get at the underlying `code` in a couple of ways:

- probably the simplest is to just double-click with the mouse inside the cell. In order to return to normal viewing, press `shift-[enter]`
- in `colab`, you can click the right-hand mouse button to get a menu of options - then choose “Edit”
- in `jupyter lab/notebook`, there are some key sequences that enable editing, but I think I won’t go into those details for now. “double-click” should always work...

6 Code!

More importantly, `jupyter` permits you to view and evaluate `code`. For this course, we’ll always use code in the `python` (specifically: `python3`) language, but other possibilities are available.

Remember the `cells` that we mentioned above? Well, there are a few types of cells. One is called a `markdown` cell, and such cells contain text (and mathematics) for reading. Another is called a `code cell`, and it contains (in this case) `python` code.

The next cell is an example of a `code cell`.

```
[5]: from math import sin,cos

def g(x):
    return sin(x)**2 + cos(x)**2

def h(x):
    return sin(x)**2 - cos(x)**2

for i in range(15):
    print(f"{i} - {g(i):.5f} - {h(i):.5f}")
```

```
0 - 1.00000 - -1.00000
1 - 1.00000 - 0.41615
2 - 1.00000 - 0.65364
3 - 1.00000 - -0.96017
4 - 1.00000 - 0.14550
5 - 1.00000 - 0.83907
6 - 1.00000 - -0.84385
7 - 1.00000 - -0.13674
8 - 1.00000 - 0.95766
9 - 1.00000 - -0.66032
10 - 1.00000 - -0.40808
11 - 1.00000 - 0.99996
12 - 1.00000 - -0.42418
```

```
13 - 1.00000 - -0.64692
14 - 1.00000 - 0.96261
```

To execute the contents of a **code cell**, type [shift]-[enter] while that cell has the focus. If the code in the cell produced **output**, code execution will result in a new cell containing that output.

Code in later code-cells will “remember” definitions from earlier cells.

For example:

```
[6]: h(3.5)
```

```
[6]: -0.7539022543433047
```