

# Math 248

Jack Bullen

November 2, 2023

## 1 `jupyter/notebook/`

Jupyter Notebook is a web application for creating and sharing documents that contain live code, equations, visualizations and explanatory text.

## 2 `jupyter/markdown/`

Markdown is a simple text-to-HTML markup language written in plain text. Jupyter notebook recognizes markdown and renders markdown code as HTML. See [Markdown](#) (by John Gruber) and [GitHub Markdown Help](#) for more information.

## 3 `jupyter/latex/`

LaTeX is a programming environment for producing scientific documents. Jupyter notebook recognizes LaTeX code written in markdown cells and renders the mathematical symbols in the browser using the [MathJax](#) JavaScript library.

## 4 python/numbers/

The main numeric types in Python are integers, floating point numbers (or floats) and complex numbers. The syntax for arithmetic operators are: addition `+`, subtraction `-`, multiplication `*`, division `/` and exponentiation `**`.

## 5 python/variables/

Just like the familiar variables  $x$  and  $y$  in mathematics, we use variables in programming to easily manipulate values. The assignment operator `=` assigns values to variables in Python. Choose descriptive variable names and avoid special Python reserved words.

## 6 python/sequences/

The main sequence types in Python are lists, tuples and range objects. The main differences between these sequence objects are:

## 7 python/functions/

A function takes input parameters, executes a series of computations with those inputs and then returns a final output value. Functions give us an efficient way to save and reuse a block of code over and over again with different input values. There are built-in functions in the standard Python library and we can define our own functions.

## 8 python/logic/

The boolean type has only two values: True and False. Let's assign a boolean value to a variable and verify the type using the built-in function `type()`:

## 9 python/loops/

A for loop allows us to execute a block of code multiple times with some parameters updated each time through the loop. A for loop begins with the for statement:

## 10 scipy/numpy/

NumPy is the core Python package for numerical computing. The main features of NumPy are:

## 11 scipy/matplotlib/

Matplotlib is a Python package for 2D plotting and the `matplotlib.pyplot` sub-module contains many plotting functions to create various kinds of plots. Let's get started by importing `matplotlib.pyplot`.

## 12 root-finding/root-finding/

Root finding refers to the general problem of searching for a solution of an equation  $F(x) = 0$  for some function  $F(x)$ . This is a very general problem and it comes up a lot in mathematics! For example, if we want to optimize

a function  $f(x)$  then we need to find critical points and therefore solve the equation  $f'(x) = 0$ .

## 13 root-finding/bisection/

The simplest root finding algorithm is the bisection method. The algorithm applies to any continuous function  $f(x)$  on an interval  $[a, b]$  where the value of the function  $f(x)$  changes sign from  $a$  to  $b$ . The idea is simple: divide the interval in two, a solution must exist within one subinterval, select the subinterval where the sign of  $f(x)$  changes and repeat.

## 14 root-finding/secant/

The secant method is very similar to the bisection method except instead of dividing each interval by choosing the midpoint the secant method divides each interval by the secant line connecting the endpoints. The secant method always converges to a root of  $f(x) = 0$  provided that  $f(x)$  is continuous on  $[a, b]$  and  $f(a)f(b) < 0$ .

## 15 root-finding/newton/

Newton's method is a root finding method that uses linear approximation. In particular, we guess a solution  $x_0$  of the equation  $f(x) = 0$ , compute the linear approximation of  $f(x)$  at  $x_0$  and then find the  $x$ -intercept of the linear approximation.

## 16 integration/integrals/

The definite integral of a function  $f(x)$  over an interval  $[a, b]$  is the limit

## 17 integration/riemann-sums/

A Riemann sum of a function  $f(x)$  over a partition

## 18 integration/trapezoid-rule/

The definite integral of  $f(x)$  is equal to the (net) area under the curve  $y = f(x)$  over the interval  $[a, b]$ . Riemann sums approximate definite integrals by using sums of rectangles to approximate the area.

## 19 integration/simpsons-rule/

Simpson's rule uses a quadratic polynomial on each subinterval of a partition to approximate the function  $f(x)$  and to compute the definite integral. This is an improvement over the trapezoid rule which approximates  $f(x)$  by a straight line on each subinterval of a partition.

## 20 differentiation/differentiation/

The derivative of a function  $f(x)$  at  $x = a$  is the limit

## 21 differential-equations/first-order/

A differential equation is an equation involving an unknown function  $y(t)$  and its derivatives  $y', y'', \dots$ , and the order of a differential equation is the highest order derivative of  $y(t)$  appearing in the equation. There are methods to solve first order equations which are separable and/or linear however most differential equations cannot be solved explicitly with elementary functions. We can always use graphical methods and numerical methods to approximate solutions of any first order differential equation.

## 22 differential-equations/numerical-methods/

Most differential equations cannot be solved explicitly using elementary functions however we can always approximate solutions using numerical methods. The order of a numerical method describes how much the error decreases as the step size decreases. Higher order methods are more accurate however they require more computations to implement. Euler's method is the simplest method however the Runge-Kutta method (RK4) is the most commonly used in practice.

## 23 differential-equations/systems/

A system of differential equations is a collection of equations involving unknown functions  $u_0, \dots, u_{N-1}$  and their derivatives. The dimension of a system is the number  $N$  of unknown functions. The order of the system is the highest order derivative appearing in the collection of equations. Every system of differential equations is equivalent to a first order system in a higher dimension.

## 24 linear-algebra/linear-algebra-scipy/

The main Python package for linear algebra is the SciPy subpackage `scipy.linalg` which builds on NumPy. Let's import both packages:

## 25 linear-algebra/solving-linear-systems/

A linear system of equations is a collection of linear equations

## 26 linear-algebra/eigenvalues-eigenvectors/

Let  $A$  be a square matrix. A non-zero vector  $\mathbf{v}$  is an eigenvector for  $A$  with eigenvalue  $\lambda$  if

## 27 linear-algebra/applications/

Polynomial interpolation finds the unique polynomial of degree  $n$  which passes through  $n + 1$  points in the  $xy$ -plane. For example, two points in the  $xy$ -plane determine a line and three points determine a parabola.

## 28 problems/problems/

Write a function called `newton` which takes input parameters  $f$ ,  $x_0$ ,  $h$  (with default value 0.001), tolerance (with default value 0.001) and max iter (with default value 100). The function implements Newton's method to approximate a solution of  $f(x) = 0$ . In other words, compute the values of the recursive sequence starting at  $x_0$  and defined by

**28.1**

$$\sum_{k=0}^{\infty} \frac{(-1)^k E_{2*k} z^{2k}}{(2k)!}$$

**28.2**

$$\sum_{k=0}^{\infty} \frac{E_{2*k} z^{2k}}{(2k)!}$$

**28.3**

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} z^{2k}}{(2k)!}$$

**28.4**

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} z^{2k}}{2 (2k)!}$$



28.5

$$\sum_{k=0}^{\infty} \frac{z^{2k+1} (2k)!}{2^{2k} (2k+1) k!^2}$$

28.6

$$\sum_{k=0}^{\infty} \frac{(-1)^k z^{2k+1} (2k)!}{2^{2k} (2k+1) k!^2}$$

28.7

$$\sum_{k=0}^{\infty} \frac{(-1)^k z^{2k+1}}{2k+1}$$

28.8

$$\sum_{k=0}^{\infty} \frac{z^{2k+1}}{2k+1}$$

**28.9**

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1} z^{2k} (2k)!}{2^{2k+1} k^2 (k!)} + \log(2)$$

**28.10**

$$\sum_{k=0}^{\infty} z^k \frac{(4k)!}{2^{4k} \sqrt{2} (2k)! (2k+1)!} = \sqrt{\frac{1 - \sqrt{1-z}}{z}}$$

**28.11**

$$\sum_{k=0}^{\infty} z^{2k+2} \frac{2^{2k} k!^2}{(k+1) (2k+1)!}$$

**28.12**

$$\sum_{n=0}^{\infty} z^{2n} \frac{\prod_{k=0}^{n-1} (\alpha^2 + 4k^2)}{(2n)!}$$

**28.13**

$$(z+1)^\alpha$$

**28.14**

$$\sum_{k=0}^{\infty} z^k \binom{\alpha+k-1}{k} = \frac{1}{(1-z)^\alpha}$$

**28.15**

$$\sum_{k=0}^{\infty} \frac{z^k \binom{2k}{k}}{k+1} = \frac{1 - \sqrt{1-4z}}{2z}$$

**28.16**

$$\sum_{k=0}^{\infty} z^k \binom{2k}{k} = \frac{1}{\sqrt{1-4z}}$$

**28.17**

$$\sum_{k=0}^{\infty} z^k (\alpha choose k + 2k) = \frac{\frac{1}{2z} (1 - \sqrt{1 - 4z})}{\sqrt{1 - 4z}}$$

**28.18**

$$H(x)$$

**28.19**

$$\sum_{k=1}^{\infty} H_k z^k = \frac{(-1) \log (1 - z)}{1 - z}$$

**28.20**

$$\sum_{k=1}^{\infty} z^{k+1} \frac{H_k}{k + 1} = \frac{\log (1 - z)^2}{2}$$

**28.21**

$$\sum_{k=1}^{\infty} z^{2k+1} \frac{(-1)^{k-1} H_{2*k}}{2k+1} = \frac{\log(z^2+1) \operatorname{atan}(z)}{2}$$

**28.22**

$$\sum_{\substack{0 \leq k \leq 2n \\ 0 \leq n \leq \infty}} \frac{(-1)^k}{2k+1} \frac{z^{4n+2}}{4n+2} = \frac{\operatorname{atan}(z)}{4}$$

**28.23**

$$\sum_{n=0}^{\infty} \frac{x^2}{n^2(n+x)} = x \frac{\pi^2}{6} - H(x)$$

**28.24**

$$\sum_{k=0}^n nchoosek = 2^n$$

**28.25**

$$\sum_{k=0}^n (-1)^k n \text{choose} k = 0$$

**28.26**

$$\sum_{k=0}^n k \text{choose} m = (1 \text{choose} m + n) + 1$$

**28.27**

$$\sum_{k=0}^n (-1 \text{choose} k + (k + m)) = m \text{choose} n + n$$

**28.28**

$$\sum_{k=0}^n \alpha \text{choose} k (\beta \text{choose} n - k) = \alpha + \beta \text{choose} n$$

**28.29**

$$\sum_{k=1}^{\infty} \frac{\sin(k\theta)}{k} = \frac{\pi - \theta}{2}$$

**28.30**

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} \cos(k\theta) = \frac{\log(2 \cos(\theta) + 2)}{2}$$

**28.31**

$$\sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} \sin(k\theta) = \frac{\theta}{2}$$

**28.32**

$$\sum_{k=1}^{\infty} \frac{\cos(2k\theta)}{2k} = -\frac{\log(2 \sin(\theta))}{2}$$

**28.33**

$$\sum_{k=1}^{\infty} \frac{\sin(2k\theta)}{2k} = \frac{\pi - 2\theta}{4}$$

**28.34**

$$\sum_{k=0}^{\infty} \frac{\cos(\theta(2k+1))}{2k+1} = \frac{\log(\cot(\frac{\theta}{2}))}{2}$$

**28.35**

$$\sum_{k=0}^{\infty} \frac{\sin(\theta(2k+1))}{2k+1} = \frac{\pi}{4}$$

**28.36**

$$\sum_{k=1}^{\infty} \frac{\sin(2\pi kx)}{k}$$



**28.37**

$$B_n(x) = -\frac{n!}{2^{n-1}\pi^n} \sum_{k=1}^{\infty} \frac{1}{k^n}$$

**28.38**

$$\sum_{k=1}^{n-1} \sin\left(\frac{k\pi}{n}\right)$$

**28.39**

$$\sum_{k=1}^{n-1} \sin\left(\frac{2k\pi}{n}\right) = 0$$

**28.40**

$$\sum_{k=0}^{n-1} \csc^2\left(\theta + \frac{k\pi}{n}\right) = n^2 \csc^2(n\theta)$$

**28.41**

$$\sum_{k=1}^{n-1} \csc^2 \left( \frac{k\pi}{n} \right) = \frac{n^2 - 1}{3}$$

**28.42**

$$\sum_{k=1}^{n-1} \csc^4 \left( \frac{k\pi}{n} \right) = \frac{(n^4 + 10n^2) - 11}{45}$$

**28.43**

$$\sum_{n=a+1}^{\infty} \frac{a}{-a^2 + n^2} = \frac{H_{2*a}}{2}$$

**28.44**

$$\sum_{n=0}^{\infty} \frac{1}{a^2 + n^2} = \frac{a\pi \coth(a\pi) + 1}{2a^2}$$

**28.45**

$$\displaystyle \sum_{n=0}^{\infty} \frac{1}{4a^4 + n^4} = \frac{1}{8a^4}$$

**28.46**

$$n$$

**28.47**

$$\displaystyle \sum_{n=-\infty}^{\infty} e^{-n^2\pi} = \frac{\sqrt[4]{\pi}}{\Gamma\left(\frac{3}{4}\right)}$$

**28.48**

$$\sum_{n=-\infty}^{\infty} e^{-n^2\pi} = \frac{\sqrt[4]{\pi}}{\Gamma\left(\frac{3}{4}\right)}$$

**28.49**

$$\sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} = \text{cdots} + \left( -\frac{1}{4} + \left( \frac{1}{3} + \left( -\frac{1}{2} + 1^{-1} \right) \right) \right)$$

**28.50**

False

**28.51**

$$\sum_{k=0}^{\infty} \frac{1}{k!} = c + \left( \frac{1}{4!} + \left( \frac{1}{3!} + \left( \frac{1}{2!} + \left( \frac{1}{0!} + \frac{1}{1!} \right) \right) \right) \right)$$

**28.52**

False

**28.53**

False

**28.54**

False

**28.55**

False

**28.56**

False

**28.57**

$$\sum_{k=1}^{\infty} \frac{(-1)^k}{k^2 + 1} = \cdots + \left( \left( \left( -\frac{1}{2} + \frac{1}{5} \right) - \frac{1}{10} \right) + \frac{1}{17} \right)$$

**28.58**

$$\frac{4}{8 \cdot 6 \cdot 7} - \left( -3 - \frac{1}{6} + \frac{1}{30} \right)$$

**28.59**

$$\sum_{k=1}^{\infty} \frac{1}{T_k} = c + \left( \frac{1}{15} + \left( \frac{1}{10} + \left( \frac{1}{6} + \left( \frac{1}{3} + 1^{-1} \right) \right) \right) \right)$$

**28.60**

$$T_n$$

**28.61**

$$\sum_{k=1}^{\infty} \frac{1}{Te_k} = \frac{1}{35} + \left( \frac{1}{20} + \left( \frac{1}{10} + \left( \frac{1}{4} + 1^{-1} \right) \right) \right)$$

**28.62**

$$Te_n$$

**28.63**

False

**28.64**

False

**28.65**

$$f(x) = \frac{\pi}{4}$$

**28.66**

$$\frac{\pi}{4}$$

**28.67**

$$\frac{\pi}{4}$$