

Assignment #9

Math 1800: Mathematical Programming in Python

Instructions: Choose 3 of the following problems to work on. Submit your responses as Python text files, with the extension `.py`. Each file should include your name and the problem number. This problem set is due Thursday, March 30, by midnight.

- *Problem 9.0:* Consider the following astronomical data:

Name	Mass(kg)	Radius(km)
Sun	1.988e30	695500
Mercury	3.301e23	2440
Venus	4.867e24	6052
Earth	5.972e24	6371
Mars	6.417e23	3390
Jupiter	1.899e27	69911
Saturn	5.685e26	58232
Uranus	8.682e25	25362
Neptune	1.024e26	24622

The density ρ of a sphere of radius r and mass m is $\rho = 3m/(4\pi r^3)$. Store the astronomical information as a Python dict called *planets*, in which “name” is the key, and “planets[name]” is a list of [mass, radius]. This means that the mass of Saturn, for instance is `planets['Saturn'][0]`. Use a `for` loop to read the data for each planet from the dict, and print a table of name, mass, radius, density;

- *Problem 9.1:* Python can split a text into individual words, consider each word, and report the length of that word in characters. Suppose we want to construct a record of the word lengths in a given text. We might try to create a dict, in which the key is the word length, and the value is the number of times that word length is observed.

If we start with an empty dict called `word_length`, and we find a word of length n , we want to say something like

```
word_length[n] = word_length[n] + 1
```

that is, the key n should have its value increased by 1. However, if this is the first time we’ve seen a word of length n , there is no key n . This is fixed by including an initialization that only happens the first time a given value n is observed.

```
text = ...
for word in text.split():
    n = len(word)
    try:
        word_length[n] = word_length[n] + 1
    except KeyError:
        word_length[n] = 1
```

Try this using

```
text = 'four score and seven years ago our fathers brought forth ' \
'on this continent a new nation conceived in liberty and dedicated ' \
'to the proposition that all men are created equal'
```

(You can allow text to run over multiple lines this way, or else just have the text be one very long line.) Then print `word_length` to report the word length frequencies.

- *Problem 9.2:* The file `morse.py`, available at <https://scipython.com/ex/bdb>, defines a dictionary which converts an upper case letter or digit to its Morse code equivalent. Write a program that uses this dict to translate a message into Morse code. Use spaces to separate each Morse code symbol, and a slash / to delimit words. For example, "PYTHON 3" becomes

.--- .-.- - --- -. / ...--

Now translate the message 'END OF SEMESTER IS COMING'

- *Problem 9.3:* The function `x, w = scipy.special.roots_legendre (n)` can be used to approximate the integral of a function $f(x)$ over the interval $-1 \leq x \leq +1$ as follows:

$$Q(f) = \sum_{0 \leq i < n} w_i f[x_i] \approx I(f) = \int_{-1}^{+1} f(x) dx$$

For smooth functions, the accuracy of this approximation increases very fast as n is increased. Consider the function

$$f(x) = \frac{1}{x^4 + x^2 + 0.9}$$

whose integral over $-1 \leq x \leq +1$ is $I(f) = 1.5822329637296729331\dots$. Write a program that estimates this integral, and use it to determine a value for n such that estimate is accurate to 10 digits.

- *Problem 9.4:* Use the function `scipy.integrate.quad(f, xmin, xmax)` to approximate the integral of

$$f(x) = \frac{1}{x^4 + x^2 + 0.9}$$

over the interval $-1 \leq x \leq +1$. Given that the exact integral is $I(f) = 1.5822329637296729331\dots$, report the error in the estimate as computed by the `quad()` function. Note that `quad()` returns several output values, but you are only interested in the first one. So your call might look like

```
estimate, _ = quad ( f, xmin, xmax )
```

where the underscore says *I don't care about the other outputs!*.

- *Problem 9.5:* The function `scipy.optimize.minimize_scalar(f)` can be used to estimate the location of an argument x that returns a minimum value of the given function $f()$. Consider the function

$$f(x) = 2x^4 - 4x^2 + x + 20$$

You might check a plot of this function over the interval $-2 \leq x \leq +2$ to get an idea of its behavior. You should see that there are two local minimizers of the function. Figure out a way, using the optional `bound=[]` input argument, so that you can make two separate calls to `minimize_scalar()` which will give you the value of both of these local minimizers.

- *Problem 9.6:* The function `scipy.optimize.minimize(f, start)` can be used to estimate the location of an argument that returns a minimum value of the given function $f()$. Consider the function

$$f(x, y) = (4 - 2.1 * x^2 + x^4/3) * x^2 + x * y + 4 * (y^2 - 1) * y^2$$

and assume your starting point is $(-1.5, 0.5)$. Write a program which uses the `minimize()` function, and report the minimizer (x, y) and the value of $f(x, y)$. Note that `minimize()` returns a single data structure `res`, but that your minimizer is stored as `res.x`. By default, `minimize()` acts as though you have specified `method='Nelder-Mead'`, a very simple algorithm. After you have computed the minimizer this way, try again, but use `method='Powell'`. You should see that this second method does a better job of minimization for this problem!

- *Problem 9.7:* The function `scipy.optimize.brentq(f,a,b)` can be used to estimate the location of an argument x which is a root of the given function $f()$ in the interval $a \leq x \leq b$. Consider the function

$$f(x) = 5 - x - 2 \sin(x)$$

which is a version of Kepler's equation. In order to use `brentq()`, the function must have opposite signs at the points a and b . Determine appropriate values of a and b , request an estimate for the root x , and print the value of $f(x)$, which should be close to zero.

- *Problem 9.8:* The function `g=scipy.interpolate.interp1d(xdata,ydata,kind=)` can define a function $g(x)$ which interpolates a set of data using a particular estimate determined by the `kind=` argument. Consider the `scipy.special` function `f(x)=jn(5,x)`, which is the Bessel function of integer order $n = 5$. Using `ndata=21` evenly spaced points `xdata` between 0 and 10, get sample values `ydata` of this Bessel function. Use `interp1d()` with `kind = 'cubic'` to find an interpolating function $g(x)$ to this data. Compare your original function (red line) and your interpolant (blue dots) on a plot, using 51 plot points over the interval.
- *Problem 9.9:* The `fft()` and `ifft()` functions in `scipy.fft` perform the Fast Fourier Transform of data, and its inverse. Typically, the data is some kind of time-varying signal $f(t)$. Applying the Fast Fourier Transform to the input data results in frequency information. Various operations can be performed in the frequency domain, after which the inverse transform returns a modified signal $\hat{f}(t)$. However, if we don't make any changes in frequency space, then applying the FFT followed by the inverse FFT simply returns our data.

Consider the signal

$$f(t) = 3 \sin(2\pi t) + \sin(8\pi t) + 0.5 \sin(14\pi t)$$

Create a data vector `x` by evaluating $f(t)$ at 101 points in the interval $0 \leq t \leq 1$. Apply the FFT to this sample data to create a vector `y`

```
y = fft ( x )
```

Then immediately apply the inverse FFT to create a vector `z`:

```
z = ifft ( y )
```

Now use `np.linalg.norm()` to measure and report $\|x - z\|$.

- *Problem 9.8:*
- *Problem 9.9:*