Exercises for

Scientific Computing using Python
—
the Python Programming Language

Thomas Arildsen, Tobias Lindstrøm Jensen

The exercises are based on:

Hans Peter Langtangen, *“A Primer on Scientific Programming with Python,”* Third Edition, 2012

Upload one of the solutions at the Moodle page before tomorrow.

Exercise 1 — Integration

Consider the approximation of definite integral using the mid point integration method:

$$\int_a^b f(x)dx \approx h \sum_{i=0}^{n-1} f(a + \frac{1}{2}h + ih) \tag{1}$$

with $h = \frac{b-a}{n}$.

- Make a Python function `midpointint(f, a, b, n):` that applies the mid point integration method where `f` is a scalar function that can be evaluated as `f(x)`.
- Compute closed form solutions of $\int_a^b f(x)dx$ for your favorite f, e.g. $\exp, \sin, \cos$.
- Compare the result of you implementation with closed form solutions.

Exercise 2 — Sorting

The file `stars.list` contains a list of stars.

```
data = [
(’Alpha Centauri A’,    4.3,  0.26,      1.56),
(’Alpha Centauri B’,    4.3,  0.077,     0.45),
(’Alpha Centauri C’,    4.2,  0.00001,   0.00006),
("Barnard’s Star",      6.0,  0.00004,   0.0005),
(’Wolf 359’,            7.7,  0.000001,  0.00002),
(’BD +36 degrees 2147’, 8.2,  0.0003,    0.006),
(’Luyten 726-8 A’,      8.4,  0.000003,  0.00006),
(’Luyten 726-8 B’,      8.4,  0.000002,  0.00004),
(’Sirius A’,            8.6,  1.00,      23.6),
(’Sirius B’,            8.6,  0.001,     0.003),
(’Ross 154’,            9.4,  0.00002,   0.0005),
]
```

Each entry of the list is a tuple containing (in order):

- The name of the star.
- The distance from the sun in light years.
- The apparent brightness.
- The luminosity.

The task is to sort the list according to either distance, brightness or luminosity, that is the 2nd, 3rd and 4th element of the 4-tuple.

A method to do so is to use the built-in function `sorted(iterable, key=None, reverse=False) -> new sorted list`. We will use the keyword argument `key` to sort according to other keys. Specifically, the function has the form `key(a)` where `a` is an objects in the list/iterable input to `sorted`. The function should return a corresponding value to which the sorting should be applied.

- Read the task carefully and make sure you understand the task.
- Make a function that can be called as `sorted_stars(data, ’distance’)` and returns the list of stars in ascending (increasing) order of distance.
- Print the list of stars sorted according to distance, brightness or luminosity.

Exercise 3 — Finding primes

The algorithm Sieve of Eratosthenes is used for finding all the prime numbers less than or equal to a number N.

- Read and understand the algorithm. You can check it out on e.g. Wikipedia.org
- Sketch the algorithm using pen and paper and decide for data types/structures.
- Implement the algorithm in a seperate module. Remember to add a descriptive doc-string.