# Homework 6

Programming for Scientists
Due on: Mar 3

## §1 QUESTIONS

**Question I:** What is under-flow (when talking about numerical precision)?

**Question II:** What is the smallest value (i.e., smallest absolute value) that a 32 bit IEEE-754 float can represent? (Hint: Wikipedia has the answer to this question).

**Question III:** You will sometimes see the following programming idiom:

```
import numpy as np
mystery = np.uint32(-1)
```

What is the value of mystery? Why would we be interested in this particular value? (Hint: think of its bit representation).

**Question IV:** With numpy, we saw that we could easily create an array of 32 bit numbers:

```
import numpy as np
A = np.array([1,2,3,4,5],np.int32)
```

However, it is also possible to simply use a traditional Python list:

```
A = [1,2,3,4,5]
```

This has the advantage of using Python numbers, which are of infinite precision, instead of being limited to 32 bits. Given this obvious disadvantage, why would anyone use the 32 bit array? (There is more than one reason, but you only need to give one).

**Question V:** Consider the following code

```
import numpy as np
...
M = A.ptp(0)
```

If $A$ is of shape $(120, 1000)$, what is the resulting shape of $M$? What does the ptp function return?

## §2 PROGRAMMING ASSIGNMENT

Consider the following approximation to compute an integral:

$$\int_0^1 f(x)dx \approx \sum_{i=0}^{999} \frac{f(i/1000)}{1000}. \tag{1}$$

1. Implement two versions of this for integrating $f(x) = x^2$. One should use "pure Python" (i.e., you should not require any imports). Another should be based on numpy (where you compute all the values $f(i/1000)$ in a single step).

2. Determine which version is faster. (Hint: You might need to run each version a many times to be able to get a meaningful answer.)