Numtest Documentation

Release 1.1.0

Sébastien Boisgérault

July 04, 2016

Contents

Numtest is a doctest extension that simplifies the test of numerical results.

It provides a new doctest directive `NUMBER` to use with numerical tests:

```
>>> import math
>>> math.pi # doctest: +NUMBER
3.14
```

To enable it, import the module `numtest` before you run the tests.

CHAPTER 1

How does it work ?

A major issue in numerical tests is the control of the precision of the results.

Numtest infers the precision that you want from the number of digits used in the expected result: the string `"3.14"` used in the above example means that the best three-digit approximation of `math.pi` should be `3.14`.

For example, a `math` module that would declare `pi` as any of the numbers

```
3.141592653589793, 3.14, 3
```

would pass the above test while the values

```
100, 3, 3.1, 3.149
```

would fail the same test.

CHAPTER 2

Additional Features

Numtest supports comparison of list of numbers and NumPy arrays. For example, the test below passes:

```
>>> import numpy
>>> x = numpy.linspace(0.0, 1.0, 4)
>>> x # doctest: +NUMBER
[0.00, 0.333, 0.667, 1.00]
```

Only the shape and values of lists or arrays are tested: lists may be successfully compared to arrays, arrays of integers with arrays of floats, etc.

Gotchas

Be aware that NumPy displays arrays with only a 8-digits precision by default. Hence the test below would fail:

```
>>> x # doctest: +NUMBER
[0.00, 0.333333333333, 0.667, 1.00]
```

However, the display precision used by NumPy is configurable. The prior invocation of

```
>>> numpy.set_printoptions(precision=17)
```

will make the test pass as expected.