

# Bibliotecas Científicas

Técnicas Experimentales  
Práctica de Laboratorio #13

7 de mayo de 2014

## Resumen

El objetivo de esta práctica es hacer uso de bibliotecas científicas para realizar representaciones gráficas.

## 1. Motivación y Objetivos

Como se ha indicado Python es un lenguaje de programación de propósito general. Se trata de un lenguaje interpretado y de tipos dinámicos. Es un lenguaje potente para realizar prototipos y también para realizar aplicaciones.

1. NumPy es una extensión del lenguaje que define los tipos de datos `'array'` y `'matrix'` así como operaciones básicas con ellos.
2. SciPy es otra extensión del lenguaje que proporciona herramientas para realizar cálculo numérico, optimización, estadísticas y más cosas.

El desarrollo de SciPy comenzó en el año 2001. Sus orígenes se remontan al paquete con extensiones numéricas para Python denominado Numeric. Posteriormente apareció Numarray, con la intención de construir un paquete más flexible y de código más limpio, aunque resultó ser más lento para cálculos matriciales en pocas dimensiones. En el año 2005, el principal impulsor de SciPy, Travis Oliphant, reunificó ambos en un único paquete que integrase las ventajas de ambos, y que se denominó Numpy, que se puede considerar como el núcleo de SciPy. SciPy se concibe actualmente como una extensión de las funcionalidades de Numpy.

SciPy posee módulos para optimización de funciones, integración, funciones especiales, resolución de ecuaciones diferenciales ordinarias y otros muchos aspectos. Puede ser usado con Linux, Windows y ha sido también compilado para Mac.

La organización de SciPy se estructura en subpaquetes, que se pueden considerar especializados en dominios científicos determinados:

- a) stats (*Statistical Functions*)
- b) sparse (*Sparse matrix*)
- c) lib (*Python wrappers to external libraries*)
- d) linalg (*Linear algebra routines*)
- e) signal (*Signal Processing Tools*)
- f) misc (*Various utilities that don't have another home*)
- g) interpolate (*Interpolation Tools*)
- h) optimize (*Optimization Tools*)
- i) cluster (*Vector Quantization / Kmeans*)
- j) fftpack (*Discrete Fourier Transform algorithms*)
- k) io (*Data input and output*)
- l) maxentropy (*Routines for fitting maximum entropy models*)
- m) integrate (*Integration routines*)
- n) lib.lapack (*Wrappers to LAPACK library*)
- ñ) special (*Special Functions*)
- o) lib.blas (*Wrappers to BLAS library*)

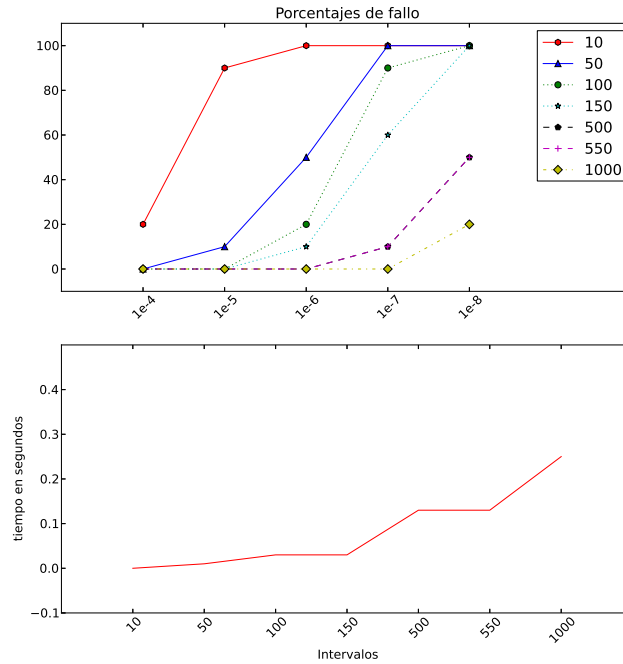


Figura 1: Representación de los experimentos de aproximación de  $\pi$

En versiones anteriores se incluían paquetes para dibujo. En la actualidad se sigue la filosofía de dejar de integrar este tipo de funcionalidades y apoyarse en proyectos que se desarrollan aparte.

SciPy es software libre, que se distribuye con la licencia tipo BSD, en la que tanto el uso comercial como no comercial es libre. Sin embargo, no es GPL por lo que no puede ampliarse con este tipo de software.

La fuente primordial de información es la página web del proyecto: <http://www.scipy.org>

3. Mathplotlib es otra extensión del lenguaje que facilita el trazado de gráficos.

El objetivo de esta práctica de laboratorio es utilizar algunas de las funciones definidas en estas bibliotecas.

## 2. Ejercicios propuestos

Diseña un modulo Python que realice una representación gráfica de los experimentos realizados para calcular las aproximaciones del número  $\pi$ . en las prácticas anteriores. Ha de ser similar al que aparece en la Figura 1.

1. Ha de mostrarse la representación del tiempo invertido en cada experimento.
2. Ha de mostrarse los valores de porcentaje de fallos con los cinco valores de 'umbral' diferentes utilizados.

## 3. Para saber más...

Represente gráficamente la función  $\frac{4}{1+x^2}$  en el intervalo  $[0, 1]$ .

## Referencias

- [1] Tutorial de Python. <http://docs.python.org/2/tutorial/>
- [2] Documentación de SciPy. <http://docs.scipy.org/doc/>