



Universidad Americana

Facultad de Ingeniería y Arquitectura

Cálculo Científico en Python

Autores:

Andrés Ernesto Castillo Rothschuh

Fátima Marie Zogaib Gradiz

Managua, Nicaragua

18 de Marzo 2025

Introducción

El cálculo científico en Python nos permite resolver problemas matemáticos y computacionales mediante el uso de herramientas especializadas. Gracias a su sintaxis sencilla y su potente ecosistema de bibliotecas, Python se ha convertido en uno de los lenguajes más utilizados para el análisis numérico, la simulación de sistemas complejos y la manipulación de grandes volúmenes de datos. **Con bibliotecas optimizadas como NumPy, SciPy y SymPy, Python facilita la resolución de ecuaciones matemáticas, la optimización de procesos y el análisis estadístico, permitiendo a investigadores y desarrolladores abordar problemas de alta complejidad de manera eficiente.**

Uno de los casos más notables del uso de Python en el cálculo científico es la generación de la primera imagen de un agujero negro por el Event Horizon Telescope (EHT). NumPy desempeñó un papel clave en el procesamiento de enormes cantidades de datos astronómicos, permitiendo la manipulación eficiente de matrices y el desarrollo de algoritmos avanzados para la reconstrucción de imágenes.

Importancia

Python se ha convertido en un estándar para el cálculo científico debido a su facilidad de uso, sintaxis clara y la gran cantidad de bibliotecas especializadas disponibles.

Su aplicación permite:

- Resolver problemas complejos de manera eficiente.
- Automatizar cálculos repetitivos.
- Facilitar el análisis de grandes volúmenes de datos.
- Integrarse con otras herramientas y lenguajes de programación.

Librerías Comunes y Ejemplos

Algunas de las bibliotecas más utilizadas para el cálculo científico en Python incluyen **NumPy** y **SciPy**. Estas bibliotecas son ampliamente utilizadas en ciencia de datos y otras disciplinas científicas debido a su eficiencia en el manejo de grandes conjuntos de datos numéricos y matrices.

NumPy

- Manejo de arreglos multidimensionales.
- Operaciones matemáticas rápidas.
- Manipulación de datos numéricos de manera eficiente.

Scipy

Amplía las funcionalidades de NumPy al ofrecer una amplia gama de algoritmos numéricos.

- Cálculos matemáticos avanzados.
- Herramientas para aplicaciones científicas (optimización, álgebra lineal, integración, interpolación, entre otros).

La combinación de NumPy y SciPy permite a los desarrolladores y científicos abordar problemas complejos de manera eficiente, desde el procesamiento de imágenes hasta el análisis de datos biomédicos. Además, al ser de código abierto y contar con una amplia comunidad de desarrolladores, estas bibliotecas reciben actualizaciones y mejoras constantes, asegurando su relevancia y eficacia en el ámbito científico.

Ejemplo de uso:

Medidas de dispersión

Las medidas de dispersión son estadísticas que indican cuánto se alejan los datos de un conjunto de valores respecto a su valor central o promedio. Estas medidas se utilizan para analizar la variabilidad o heterogeneidad de los datos.

```

import numpy as np
from scipy import stats

import numpy as np

# Datos
datos = np.array([4, 49, 19, 4, 14, 20, 17, 50, 4, 20])

# Media
media = np.mean(datos)
print("Media:", media)

# Mediana
mediana = np.median(datos)
print("Mediana:", mediana)

# Desviación estándar
desviacion_estandar = np.std(datos)
print("Desviación estándar:", desviacion_estandar)

# Moda
moda = stats.mode(datos)
#00P objeto de la clase ModeResult, tiene atributos mode y count
print("Moda:", moda.mode)

Media: 20.1
Mediana: 18.0
Desviación estándar: 15.984054554461457
Moda: 4

```

```

#Resultado
Media: 20.1
Mediana: 18.0
Desviación estándar: 15.984054554461457
Moda: 4

```

Aquí se define una lista de datos llamada "data". La media, mediana y desviación estándar de los datos se calculan utilizando las funciones "mean", "median" y "std" de NumPy. Nótese que para el caso de la moda recurrimos a SciPy.

La media es la suma de todos los valores dividida por el número de valores, la mediana es el valor que se encuentra en el centro de un conjunto de datos ordenados, y la desviación estándar mide la cantidad de variabilidad o dispersión en los datos. La moda por su parte, es el valor que más se repite en la serie.

Generación de la imagen del agujero negro de M87

El Event Horizon Telescope (EHT) capturó más de **350 terabytes de datos diarios**, almacenados en discos duros con helio, lo que representó un desafío monumental en el procesamiento de datos astronómicos. **NumPy fue fundamental** en esta tarea, sirviendo como la base del paquete **eht-imaging**, utilizado para la simulación y reconstrucción de imágenes a partir de datos de interferometría de base muy larga (VLBI).

Así con la ayuda de NumPy, los científicos pudieron **procesar y analizar grandes volúmenes de datos** con eficiencia, optimizando operaciones numéricas esenciales para la reconstrucción de la imagen. Además, facilitó la implementación de algoritmos avanzados en múltiples equipos independientes, permitiendo **verificar la consistencia de los resultados** antes de generar la imagen final. Este enfoque garantizó la precisión y fiabilidad de la primera imagen de un agujero negro, un logro histórico en la astronomía.

Conclusión

Python es esencial en el cálculo científico por su versatilidad y potentes bibliotecas. La imagen del agujero negro de M87 demuestra su impacto en la ciencia, con NumPy liderando el procesamiento de datos y visualización. Este avance, logrado por más de 200 científicos y tecnologías innovadoras, reafirma el papel de Python en la investigación y el descubrimiento científico.

Referencias

Torres, A. (2023, Marzo). Guía de cálculo científico en Python. Matemáticas en SciPy y NumPy explicadas con ejemplos. FreeCodeCamp.org.

<https://www.freecodecamp.org/espanol/news/funciones-en-python-introduccion/>

IBM Technology. (2023, July). NumPy vs SciPy. YouTube.

<https://www.youtube.com/watch?v=l3s-8uTBVA>

NumPy - Case Study: First Image of a Black Hole. (n.d.). Numpy.org.

<https://numpy.org/case-studies/blackhole-image/>