

Diseño de software para cómputo científico.

Curso de postgrado para doctorado

Juan B. Cabral (jbcabral@unc.edu.ar)

La ciencia tiene una dependencia de grandes infraestructuras computacionales como super-computadoras e infraestructuras de redes.

Con esto las tareas del científico moderno también abarcan desarrollar herramientas confiables, optimizando el uso del cómputo, así como su tiempo en tareas de desarrollo.

Aclaraciones previas.

- **Lenguajes de alto nivel:**

Python, R, Julia, etc.

- **Lenguajes de bajo nivel:**

C, C++, FORTRAN, etc.

Objetivos

- Explicar el **correcto** uso de lenguajes de alto y bajo nivel. (utilizar la simpleza de los primeros **cuando se pueda** y la eficiencia de los segundos **cuando se necesite**).
- Introducir técnicas y tecnologías modernas para la creación de software **confiables**.

En palabras simples: Exprimir un lenguaje de alto nivel hasta el punto que sea necesario el esfuerzo de utilizar bajo nivel. (en nuestro caso el alto nivel es **Python**)

Aclaraciones posteriores.

No se va enseñar HPC (Eso es la materia de Nicolás Wolovick).

No se va enseñar grandes volúmenes de datos (Esto es la materia de Damián Barsotti)

Si se va enseñar **cuándo** es útil alguna de esas dos técnicas.

Aclaraciones Finales.

- La ingeniería de software es un área imposible de barrer extensivamente en su totalidad en una materia.
- La idea es preparar al alumno en el uso eficiente de herramientas de alto nivel así como prácticas básicas para la mejora de la calidad de sus proyectos resultantes.
- Osea: como se trabaja en la industria.

Contenido

lo que se ve.

1. Lenguajes de alto nivel.
 2. Estimaciones.
 3. Calidad de software.
 4. Persistencia de datos.
 5. Optimización, paralelismo, concurrencia y cómputo distribuido en alto nivel.
 6. Integración con lenguajes de alto nivel con bajo nivel.
 7. Utilidades.
-

Cerrando

- Se va a evaluar con trabajos prácticos y un seminario final.
- Está orientado a los doctorados de todas las licenciaturas.
- La unidad de utilidades (unidad 6) se va a ver en paralelo con las demás (vendría a ser algo así como el práctico).

Del último examen rendido hay un paquete Python publicado y un paper en revisión.

GriSPy: A Python package for Fixed-Radius Nearest Neighbors Search

Martin Chalela^{a,b,c}, Emanuel Sillero^{a,b,c}, Luis Pereyra^{a,b,c}, Mario Alejandro Garcia^d, Juan B. Cabral^{e,a}, Marcelo Lares^a, Manuel Merchn^{a,b}

^a Instituto De Astronomía Teórica Y Experimental - Observatorio Astronómico Córdoba (IATE, UNC-CONICET), Córdoba, Argentina.

^b Observatorio Astronómico de Córdoba, Universidad Nacional de Córdoba, Laprida 854, X5000BGR, Córdoba, Argentina

^c Facultad de Matemática, Astronomía y Física Universidad Nacional de Córdoba (FaMAF-UNC) Bvd. Medina Allende s/n, Ciudad Universitaria, X5000BGR, Córdoba, Argentina

^d Universidad Tecnológica Nacional, Facultad Regional Córdoba (UTN-FRC), Maestro M. Lopez, esq. Cruz Roja Argentina, Ciudad Universitaria - Córdoba Capital

^e Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS, CONICET-UNR), Ocampo y Esmeralda, S2000E2P, Rosario, Argentina.

Abstract

We present a new regular grid search algorithm for quick fixed-radius nearest-neighbor lookup developed in Python. This module indexes a set of k-dimensional points in a regular grid, with optional periodic conditions, providing a fast approach for nearest neighbors queries. In this first installment we provide three types of queries: *bubble*, *shell* and the *nth-nearest*; as well as three different metrics of interest in astronomy: the *euclidean* and two distance functions in spherical coordinates of varying precision, *haversine* and *Vincenty*; and the possibility of providing a custom distance function. This package results particularly useful for large datasets where a brute-force search turns impractical.

Keywords: Data mining; Nearest-neighbor search; Methods: Data analysis; Astroinformatics; Python Package

GriSPy

latest

Search docs

CONTENTS:

Requirements
Installation
Licence
Tutorial
API Module

Docs » GriSPy documentation

[Edit on GitHub](#)

GriSPy documentation



pypi package 0.0.2 build passing docs passing License MIT Python 3.6+

GriSPy (Grid Search in Python) is a regular grid search algorithm for quick nearest-neighbor lookup.