

CR-Analyze

Scripts para el análisis de componentes contrarotantes en IllustrisTNG

Fecha de realización: 15 de diciembre de 2023
Fecha de entrega: 15 de diciembre de 2023
La Serena

1. Introducción y Raciocinio

El análisis de simulaciones y grandes bases de datos en la astronomía, es de especial importancia a día de hoy, en donde nos encontramos claramente en un contexto social y científico, en donde el problema se centra en el gran volumen de datos que poseemos como comunidad científica. Siendo entonces imposible de realizar un análisis factible en un tiempo realista de estos mismos utilizando métodos tradicionales. Es aquí en donde las herramientas computacionales (desarrollo de algoritmos, software, paquetes, códigos, etc...) otorgan las capacidades necesarias para el desarrollo de las grandes cantidades de datos. Claramente en astronomía se poseen y se generan grandes volúmenes de datos de la observación e integración de fenómenos astrofísicos, es en este último que el trabajo presente planea dar un enfoque, ya que el análisis de datos de simulaciones hidrodinámicas no es trivial cuando se trata del análisis dinámico de las estructuras presentes en estas mismas, razón por la cual es un requerimiento de más alta importancia el desarrollo de herramientas para el análisis de estas simulaciones. Y es de las simulaciones que uno obtiene predicciones de los fenómenos que debiésemos o se espera presenciar en las observaciones, considerando claramente un modelo de universo determinado.

El trabajo aquí presente planea mejorar, limpiar y documentar los algoritmos, métodos y análisis realizados anteriormente durante el desarrollo de un taller de investigación de pregrado enfocado en componentes contrarotantes en TNG50. Estos algoritmos si bien suficientes para un análisis inicial, se quedan limitados por su escritura confusa y poco documentada de los métodos utilizados, causando así complicaciones en la eficiencia a la hora de replicar los trabajos y análisis ya realizados.

El documento se enfocará primero en describir el problema científico, luego en declarar las herramientas del curso que se tomaron como enfoque, para después describir la manera específica como se aplicaron estas herramientas, y finalmente se finalizará con unas conclusiones finales sobre las herramientas aprendidas, y la proyección a futuro del proyecto.

2. Problema

Científicamente se tiene un interés por las componentes contrarotantes, debido a su posible conexión con la evolución de las galaxias en las que se encuentran ([1]) al igual que detecciones de estas estructuras en galaxias externas ([2]).

En términos algorítmicos sin embargo, el código que se realizó en su momento para el análisis de las componentes descritas en el párrafo anterior, carece de planteamiento previo a la escritura del mismo, descendiendo así en limitaciones como fruto de ello, particularmente la replicación del análisis, la modularidad de los métodos (siendo hardcoded varias partes de los mismos), siendo entonces obligatorio en el caso de querer cambiar algún parámetro en particular, entrar en el código en sí y modificar de manera acorde.

El código escrito previamente para el análisis entonces carece de modularidad, planteamiento y transparencia en sus métodos, haciendo entonces difícil y innecesariamente complicado el análisis continuado de los datos.

3. Herramientas

Entonces para abordar este problema, el trabajo aquí presenta el concepto de "QR-Analyze", un paquete de python, que será esencialmente una mejora/modularización de las funciones y métodos que fueron utilizadas durante el trabajo del taller de investigación. Se utilizan principalmente los conceptos de programación orientada a objetos, documentación, y diseño de software, al igual que el concepto de guardado en HDF5.

3.1. Programacion Orientada a Objetos

En el proyecto presentado es la menor pero no menos importante parte del mismo, en términos sencillos la programación orientada a objetos consiste en programación que utiliza objetos y clases para organizar el código, por el hecho de utilizar python como lenguaje, es claramente necesario abordar los problemas algorítmicos con objetos y clases en mente. En el caso particular del proyecto, no existe un interés por las clases de manera particular, pero si en los objetos que salen de los métodos desarrollados.

Por lo tanto si bien no fue necesario realizar un diseño de un nuevo tipo de objeto o de la implementación de una clase, si que se tuvo que tener en cuenta las salidas y entradas de los métodos utilizados, y sobre como estos objetos y sus propiedades podían ser aprovechados.

Principalmente el proyecto hace uso de la abstracción para el diseño de los algoritmos y métodos.

3.2. Documentacion

En cuanto a la documentación se hace uso de docstrings dentro de todos los métodos realizados para poder dejar claro las limitaciones y capacidades de los mismos, declarando entonces los parámetros que acepta, sus valores por defecto, y los objetos que devuelve.

Y debido a que escribir en IPython cualquier función con la forma "paquete.funcion?.entrega el docstring, sera sencillo consultar sobre el funcionamiento de las funciones y métodos del paquete, se considero el uso de herramientas como Doxygen o Sphinx, sin embargo ambas requieren de un formato específico del docstring que se interpone a la funcionalidad mencionada al comienzo del párrafo.

3.3. Diseño de software, UML, Diagramas de flujo

Una de las mayores mejoras en comparación a los métodos por mejorar, en el proyecto se realizo una reestructuración completa de los métodos construidos en el taller, en donde se crearon y se revisaron las relaciones entre los métodos y funciones, los documentos de diseño son mostrados en la sección de avances.



Figura 1: Abstracción del método a utilizar para la rotación, del taller de investigación de licenciatura.

3.3.1. Diagramas de flujo

Los diagramas de flujo se utilizan en dos partes del diseño de este proyecto, inicialmente en el diseño general(el cual es mostrado en la sección de avances en la figura 4) para ver el flujo del análisis de los datos a lo largo de todo el proceso esperado.

Luego se utiliza en el diseño específico de los métodos mostrados en el UML, para diseñar una forma inicial del flujo de los datos para la solución de un problema en particular. La simbología utilizada si bien es similar a la que se utiliza generalmente en los diagramas de flujo con decisiones, I/O, bucles, y

asignaciones, no es completamente equivalente, aun así, se realizan en la fase previa a la escritura del pseudo-código.

3.3.2. Unified Modeling Language(UML)

Los UML son especialmente útiles para representar sistemas completos con interacciones entre métodos, transferencia de parámetros entre otros, permitiendo incluso representar bases de datos.

En el caso específico del proyecto estos UML fueron diseñados para todos los scripts ya escritos en fase de release, ya que era y es necesario para completar cualquier script en particular. En estos UML se describen inputs y outputs, además de importaciones.

3.4. Aplicación de las herramientas

En cuanto a la aplicación, primero se piensa en el problema general a abordar, la medición y detección de discos contrarotantes en galaxias. Es claro que para ello se requiere calcular la circularidad, la cual requiere de una dada rotación de las galaxias a procesar, de las cuales es necesario los datos de las partículas individuales que las conforman, finalmente llevamos el problema a la abstracción y se obtiene un diagrama de flujo general de los datos, el diagrama representativo del flujo general es mostrado en la sección de avances(Figura 2).

Se comienza por un proceso de sobreabstracción(solución simplificada), luego un diagrama de flujo general, para después un diseño del UML a utilizar(solución general), y con este se realiza el diseño de cada método individual mencionado o representado en el UML, se diseña un diagrama de flujo particular para el mismo(solución particular representada), y finalmente tomando estos diagramas de flujo de cada método se escribe el pseudocódigo(solución particular) para finalmente escribir el código final(solución particular aplicable y funcional).

4. Avances

En esta sección se muestran de manera directa los diagramas, diseños, de los métodos, al igual que se mencionan los problemas particulares que surgieron en la creación de los mismos.

Esta sección es la más particular de todas ya que mencionara el análisis y sus fases de manera directa, por lo que las estrategias y metodologías mencionadas aquí, aplican al proyecto en particular.

4.1. Limpieza de códigos y métodos

Se realizó una revisión de los scripts realizados durante el taller, y se diseñó el flujo de los datos general a través de los mismos.

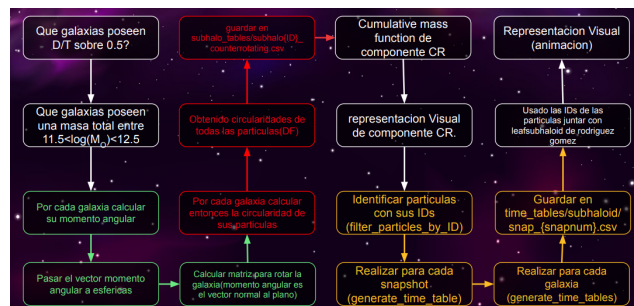
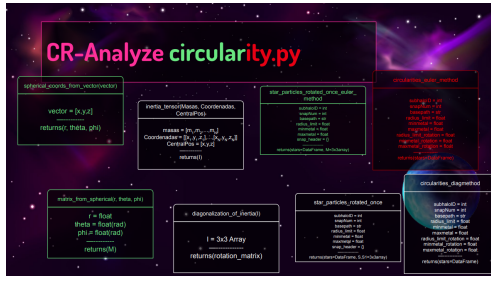
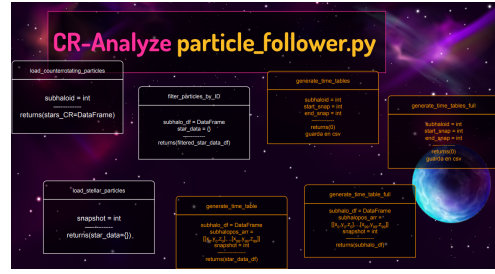


Figura 2: Flujo de datos a través de las manipulaciones en los scripts del taller de investigación de licenciatura.

Se identifican las entradas y salidas de los métodos que se planean mejorar y documentar.



(a) script circularity.py, encargado de rotar y calcular circularidades



(b) script particle_follower.py encargado de seguir y colocar el origen de las partículas

Figura 3: Metodos a mejorar en el proyecto

Teniendo presente el flujo y el diseño de los métodos y el análisis hecho anteriormente, es mas sencillo rediseñar el análisis, y los scripts.

4.2. Descripción de codigos y metodos

Se realizo durante el proceso de análisis de los métodos anteriores y el flujo de los datos, una documentación de los métodos rediseñados en base a los diagramas realizados anteriormente.

Estas docstrings describen los parámetros, el método utilizado descrito de manera superficial, y los retornos que entrega, en ese orden, el diseño de los docstrings no es final y se considera el cambio, en caso de requerir generar documentacion con los mismos.

4.3. Planteamiento y ejecución de métodos

Por ultimo durante la fase de planteamiento se construyo el planteamiento de 4 sets de scripts con métodos que interactuarían entre si, se muestra visualmente también el flujo de datos a través de estos métodos mencionados, se puede ver el diseño general en la figura 4.



Figura 4: Flujo de datos a través de los scripts, diseño general del paquete, el diagrama muestra los métodos de cada script, los outputs finales de los scripts, y las flechas indican importación de métodos

Tomando el primer diseño general como referencia, se diseñan los métodos individuales, de UML de rotator.py el cual esta dentro de los documentos de diseño en el repositorio.

Se realizo principalmente para el calculo de las circularidades un rediseño del flujo de datos dentro del mismo script, para considerar la periodicidad de la caja en el cambio del marco de referencia. Se construyeron nuevos métodos para esto último.

5. El repositorio

El repositorio contiene los scripts, con el formato de paquete en forma ya instalable, las instrucciones de instalacion se encuentran en el Readme en root.

También son entregados los documentos de diseño en versiones visualmente mejoradas de los documentos originales realizados en papel.

Entre estos documentos se incluyen los diagramas de flujo de los scripts realizados, el UML de rotator, y por ultimo el formato postulado para el guardado en HDF5.

Se hace uso de setuptools para la realización del paquete, son requeridos los paquetes de [numpy](#), [pandas](#) y [matplotlib](#), los cuales serán instalados automáticamente en caso de no estar en la maquina a instalar.

También es requerido el paquete de [ilustris-python](#), este mismo debe ser instalado siguiendo las instrucciones del repositorio del mismo. No sera instalado automáticamente a diferencia de los otros paquetes requeridos debido a no estar en pip.

6. Conclusiones/ Conocimientos adquiridos

Brevemente el proyecto logro cumplir no tan solo un rediseño completo de las metodologías usadas para el análisis en el taller de licenciatura, sino que también logro conseguir un 48 % del código ya escrito y utilizable.

Sin embargo a quedado por realizar, el guardado en el nuevo formato postulado en HDF5 y la construcción de los métodos de `stiscian.py` y `movie_maker.py`, donde ambos principalmente consistirían en representar gráficamente información obtenida de los cálculos realizados en `circularity.py`. Por ello el proyecto no se considera como completo, y se seguirá desarrollando hasta que el paquete sea utilizable de comienzo a final para el análisis de estas componentes.

En cuanto al conocimiento obtenido del proyecto, este es de suma importancia ya que la construcción de métodos informáticos para el análisis de datos, es de gran prioridad, especialmente en aquellos académicos que se centran en grandes volúmenes de datos de simulaciones, por ende la utilización de métodos de diseño de software, son esenciales para el futuro del académico.

Referencias

- [1] Bertola, F. y Corsini, E. M., “Counterrotation in Galaxies,” en *Galaxy Interactions at Low and High Redshift* (Barnes, J. E. y Sanders, D. B., eds.), pp. 149–156, Dordrecht: Springer Netherlands, 1999, [doi:10.1007/978-94-011-4665-4_32](#).
- [2] Vergani, D., Pizzella, A., Corsini, E. M., Driel, W. v., Buson, L. M., Dettmar, R.-J., y Bertola, F., “NGC 5719/13: interacting spirals forming a counter-rotating stellar disc,” *Astronomy & Astrophysics*, vol. 463, no. 3, pp. 883–892, 2007, [doi:10.1051/0004-6361:20066413](#).