

Galaxy-Merger Reconstruction

A Model-Observation Comparison

Hans-Martin Adorf, 2006-06-02

Summary: This document outlines a project which will allow a comparison between observations and theory. The observation in our case consists of a 2D-image of a galaxy. Theory is used to peek into the past in order to find out how the galaxy looked like then. The theoretical counterpart to the observation consists of a series of 3D simulations of a merger of two hypothesized parent galaxies. Each simulation produces a series of snapshots, and for each of these snapshots an optimum 2D projection/transformation is being sought. The resulting predicted 2D-image is compared with the given observed image, and a similarity measure is computed. The simulation process continues until a satisfactory prediction is found and the corresponding parameters are returned. The process described here is an example for a general inverse problem, where, based on an observation, a set of parameters is to be estimated. In our case the parameter set consists of about 10 physical parameters needed for characterizing the galaxies and the merger process, plus the time of the best matching snapshot, plus up to 6 projection/transformation parameters.

Introduction

We assume that an astronomer has observed a monochromatic image of a galaxy, and that he/she would like to understand the galaxy's plausible history, i.e. how the galaxy has probably evolved into its observed state. This is an inverse problem, for which a direct inversion is not possible. That is, it is not possible to take the observed image and, by some direct manipulations of the data, derive the parameters that would describe the merger of two hypothetical parent galaxies. Instead, a simulation framework needs to be used, which will eventually produce a series of theoretically predicted images that can directly be compared with the given observation. The best of these images is retained along with the parameters used for its production.

Processes

The framework for the inverse problem needs to perform the following processes:

- Generate multiple sets of physical parameters (using educated guesswork).
- Use each parameter set for starting a simulation, each of which will generate a number (say 100) of 3D snapshots.
- For each 3D snapshot compute an optimum 2D projection/transformation, and return a (dis-) similarity measure.
- Feed the results back to the generator of parameter sets which will launch new simulations with subsequent comparisons.
- A termination criterion is being used for terminating the simulation loop.

Observed image

In the following we assume that the observation consists of a single monochromatic image of the galaxy in question.

Input parameters

At the very beginning of the inversion process, a number (say 10) of sets of physical parameters need to be generated somehow. During later iterations, a process of “informed guessing” will be used to generate variations of the parameter sets.

Whatever is known about the physical parameters (e.g. ranges, or prior probabilities) should be captured and used in the process.

The parameter sets will probably be generated using evolutionary algorithms or Monte Carlo Markov Chain methods. In evolutionary computing, each parameter set will be represented by an individual in the population. In a recombination and mutation process offspring are generated from the current population.

Galaxy-merger simulations

In the following the term “simulation run” is used to describe an individual in the population above. The set of simulations corresponding to a set of parameter sets is called a “generation of simulations”.

Each parameter set, comprising 10 to 15 physical parameters each, will be used for starting a simulation run. These simulation runs will, if possible, run in parallel. Each simulation run generates a series of 3D snapshots.

Snapshots

Typically 50 to 100 snapshots are being produced within each simulation run. A snapshot consists of a model of the merged galaxies. This model cannot be directly viewed.

Snapshot projection and transformation

For each 3D snapshot a theoretically predicted 2D image needs to be generated that can directly be compared with the observed image. The process of generating a 2D-image from a 3D-model can be decomposed into a projection that already generates a 2D-image followed by a transformation that properly registers this image with the observed one.

There are six parameters describing a particular projection:

- Two angles describe the direction into which the simulated 3D galaxy is being projected. The result is a 2D-image.
- This 2D image needs to be registered with the given observation, i.e. scaled, shifted, and rotated in order to optimally match the observed image. The scale parameter(s) may be known, in which case the registration process only needs to estimate the relative shift and rotation. Two parameters are needed for describing the shift, two for the scale, and one for the rotation of the theoretical image, giving rise to four transformation parameters.

There is an existing C-routine that generates a 2D-projection from a 3D-snapshot.

While there are established methods (e.g. via point-pattern matching) for computing the shift, rotation and scale parameters that are needed for optimally matching two given images, it is currently unclear which method(s) to best use for selecting the two directional angles required for the projection of the 3D-model onto the 2D-plane.

It might be advantageous to use a single method for generating all the projection and transformation parameters simultaneously.

It is conceivable that the same optimization method (evolutionary algorithm or MCMC) is being used for generating sets of projection parameters that is being used for generating sets of physical merger parameters.

A promising avenue consists in using first and second image moments in order to estimate the shift, scale and rotation parameters.

Image comparison

Each predicted 2D image needs to be compared with the given 2D observation. Each image can be considered to be a point in a high-dimensional space with as many dimensions as there are pixels in the image. As objective function for the parameter set associated with a predicted 2D image, we can initially take a simple metric consisting of the (squared) Euclidean distance between the observed and the predicted image. This distance has to be minimized.

Snapshot evaluation

The optimum value of the objective function across all the investigated projections and transformations is preserved and becomes the objective value for the snapshot. The associated parameters are also preserved.

Simulation run evaluation

The optimum value of the objective function across all snapshots from a given simulation is preserved and becomes the objective value for the whole simulation. The physical parameters that belong to the optimum objective value are also preserved.

Evaluation of each generation of simulations

For a given generation of simulations the optimum value of the objective function is preserved along with the physical and projection/transformation parameters that were used.

Data management requirements

Input data

The requirements for data management are not very demanding. There is a single input datum, the observed image, whose size typically is of the order of a few Megabyte, only.

Intermediary data

Each simulation produces a series of 50 to 100 3D snapshots. These need to be stored locally until they have been compared with the 2D observation. Once the value of the objective function has been estimated, the snapshot data can in principle be deleted. The only data that need to be preserved are

- the physical parameters,
- the snapshot time (number),
- the projection and transformation parameters, and
- the value of the objective function.

From this information the simulation can be re-run and all simulation results regenerated.

However, usually the current best snapshot and the optimum theoretical 2D image will be cached for later use.

It should be noted that in principle a simulation is independent of the observed image. Only the objective function changes with the given observed 2D image.

If the physical parameters are generated by an evolutionary algorithm, it might happen that (almost) the objective value is requested more than once for the same parameter set. Also, when more than one observed image has to be „analyzed“, the same parameter set may be revisited. In these situations it might be advantageous, to cache the snapshot data, and only run the projection and transformation optimizer on the snapshots.

If millions of particles are used, the data volume may be 10 Gigabyte per snapshot, i.e. 1 Terabyte in total for 100 snapshots. For very large simulations where 10 to 100 individual simulations are running in parallel, the demands for intermediate data storage space can be considerable, i.e. up to 10 to 100 Terabyte.

Typically, though, the storage demands will be considerably lower.

Output data

The whole inverse problem process will produce the estimated parameter set along with the value of the objective function. The amount of data is negligible.

For visualization the optimum predicted image will also be part of the output. This single image will probably only amount to a few Megabyte of data.

If some or all of the snapshots of the best simulation shall be saved and returned, the output data volume may increase considerably, and final output data transfer via the network may become a problem.

Workflow and job management

The inverse problem process represents a workflow (I-WF) that can be decomposed into a number of sub-workflows.

As described, the parameter generator generates a set of parameters (corresponding to the genotypes of an evolutionary algorithm). Each parameter set triggers a full-blown simulation workflow (S-WF). These S-WFs can run in parallel. Since only the parameter set is required, these simulations can in principle be run in a *multi-site mode*, without requiring major data traffic. Each simulation requires MPI.

For the comparison with the observation, which is best done at the site where the simulation is running, only the observed image is required, which can be transmitted once at the beginning.

Each simulation produces a series of 3D snapshots, each of which needs to be analyzed by a projection-transformation-evaluation workflow (PTE-WF). These workflows can again all run in parallel. The question is whether the S-WF produces snapshots faster than the PTE-WF can analyze them. If the PTE-WF takes more time to analyze a snapshot than it takes the S-WF takes to produce one, a backlog of pending PTE-WFs will occur.

Open questions

Observations

Are there multiple observations of the same galaxy in different bandpaths? If yes, we need a more complicated process for parameter generation and comparisons.

Theory-Observation_comparison.odt

Shall only individual observed images be analyzed, or are we expecting several simultaneous analyses?

Parallelization opportunities

Currently there is no control element in the ProC that could be used for executing the workflows in parallel each of which represents a simulation run. Could a simple specific control element be designed and implemented that serves the purpose? Or should we use a simple ProC loop element?

Snapshot selection

An algorithm will be required for selection of optimal snapshot. In addition we may consider adapting the times when snapshots are going to be saved for each generation.

Scale-, shift- and rotation-invariant matching of images

Could we import an existing algorithm, rather than implementing one ourselves?

Do we have signal-to-noise information, or can we generate some in order to use a chi-squared metric as objective function, instead of the Euclidean distance?

Can we assume that the landscape of the objective function is unimodal with respect to the projection parameters?

Should other representation of the images (e.g. wavelets) be used for theory-observation matching?

Does the *intensity scale* also have to be estimated along with the six projection and transformation parameters? [Torsten says: no. The intensities are known.]

Different evolutionary algorithms

Do we have (or know of) an evolutionary algorithm that could be used for generating the sets of physical parameters, each giving rise to a simulation run? Can we import one or more algorithms? Can we use the same algorithm for generating physical parameter sets and for generating projection parameter sets?

Multiple modes

The projections and transformations have symmetries which will lead to multiple modes (peaks) of the objective function. Will there be additional modes?

Single or multiple objective function

The problem has so far been described as a single-objective optimization (SOO) problem. Do we want to eventually settle on a single objective function (Euclidean, chi-squared, other), or do we envisage computing more than one objective function simultaneously, i.e. try to solve a multi-objective optimization (MOO) problem?

Elitism – saving optimum objective values and corresponding parameter sets

We should probably save the second and third best values of the objective function along with the corresponding parameter sets.