MAGIX:

Generic tool for fitting models to astrophysical data

Irina Bernst[1], P. Schilke[1], T. Möller[1], D. Panoglou[1], V. Ossenkopf[1], M. Röllig[1], J. Stutzki[1], D. Muders[2]

[1] 1. Physical Institute, University of Cologne, Zülpicher Str. 77, 50937 Köln, Germany
[2] Max Planck Institute for Radio Astronomy, Auf dem Hügel 69, 53121 Bonn, Germany

SPONSORED BY THE Federal Ministry of Education and Research

About MAGIX

MAGIX (**M**odeling and **A**nalysis **G**eneric **I**nterface for e**X**ternal numerical codes) (http://www.astro.uni-koeln.de/projects/schilke/MAGIX)is a model optimizer developed under the framework of the CATS (**C**oherent set of **A**strophysical **T**ools for **S**pectroscopy) project. It is an ASTRONET funded German-French-Swedish Project that will provide common tools and databases for astrophysical applications.

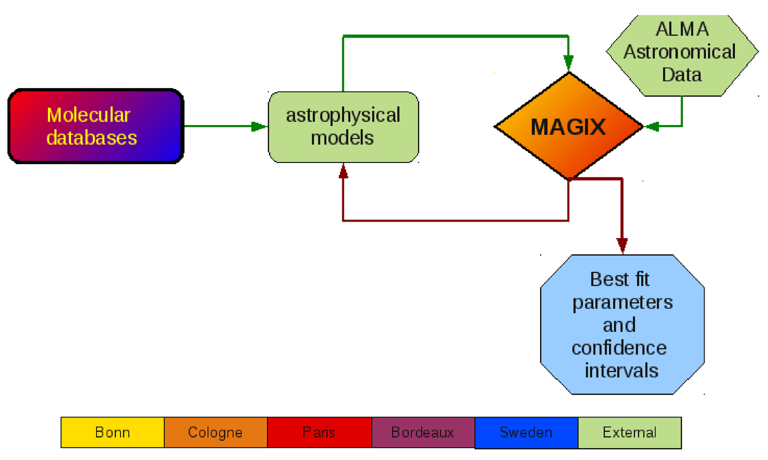

Figure 1: Schematic view of the CATS framework. It consists of a database part (on the left), provided by French and Swedish partners, and by the MAGIX part (right), provided by Germany.

(M)any theoretical models can plug into it, and its goal is to provide the best-fit parameters, within the framework of the model, to a particular data set, including confidence intervals for the parameters values. It consists of

- GUI based frontend to register new models (registration)
- to create model instances, i.e. set up a model with initial conditions (instantiation),
- the fitting engine, and an
- output module.

MAGIX is able to read a variety of model data formats, including FITS, and has a number of algorithms available for finding the best fit:

- Levenberg-Marquardt,
- Simulated Annealing,
- Particle Swarm Optimization,
- Bees algorithm,
- Genetic algorithm,
- Nested Sampling.

It is basically written in Python, whereas various algorithm packages are written in Fortran. MAGIX requires the following packages: python 2.6 (or later), numpy 1.3 (or later), pyfits 1.0 (or later), gfortran 4.3 (or later), matplotlib 0.99 (or later).

In the final version, it is envisioned that MAGIX will have a Heuristics module that will to be able choose the best combination of algorithms based on user-defined priorities.

Parallel computations

Depending on the model, the computational load is heavy (sometimes, a point of the optimization function is calculated more than one hour). For some optimization algorithms, hundreds of function calls are necessary, and improving the speed will be the next long-term goal. The whole system of MAGIX is being parallelized with parallel calculations in the models and in the optimization algorithms, increasing the computational speed.

Pre-registered models

At the moment pre-registered models include: myXCLASS (modeling of the star-forming regions with access to the CDMS and the JPL molecular data bases); SimLine and RATRAN (computing the profiles of molecular lines); myCloud (computing of 3D data cubes of spectral ranges with arbitrary input geometries and calculating the radiative transfer).

Optimization algorithms

Most physical models depend on a set of parameters, and finding the best model means finding the parameter set that most closely reproduces the data by some criteria, e.g. a minimum of the χ^2-distribution. Since most models are nonlinear functions of the input parameters, this means finding the global minimum of a multi-dimensional nonlinear function, which is by no means trivial. However, there are algorithms that can achieve the goals in most cases – examples are the Levenberg-Marquardt (conjugate gradient) method, which is fast, but can get stuck in local minima; Simulated Annealing and Particle Swarm Optimization methods, which are slower, but more robust against local minima. Other, more modern methods, such as Bees, Genetic or Nested Sampling algorithms are included in MAGIX for exploring the solution landscape, checking for the existence of multiple solutions, and giving confidence ranges.

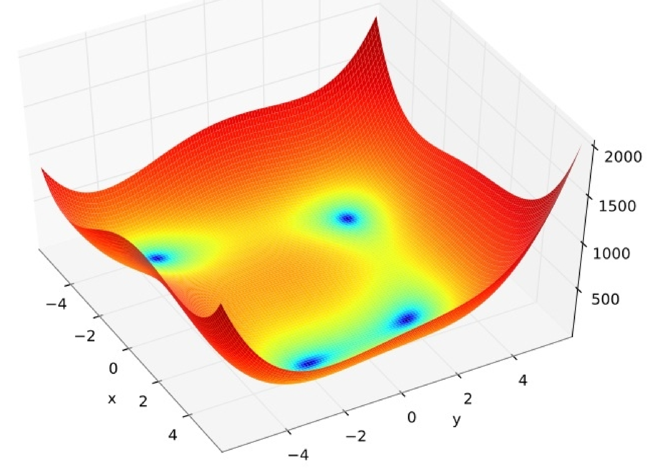

Figure 2: Analytical test function for finding minima: the Himmelblau function has four identical minima.

Bees algorithm

The Bees Algorithm (http://www.bees-algorithm.com) is an optimization algorithm inspired by the natural foraging behavior of honey bees to find the optimal solution. In its basic version, the algorithm performs a kind of neighborhood search combined with random search and can be used for function optimization. The advantages of the algorithm are

- finds multiple minima
- converges quickly
- lends itself to parallelization,

while the main disadvantage is the large computing time, because of many function calls.

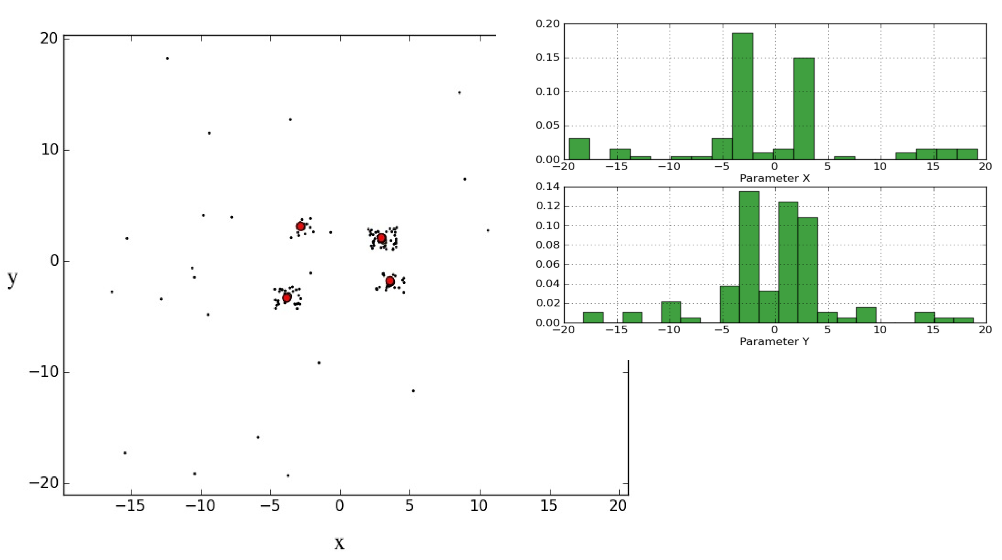

Figure 3: Bees algorithm results on the Himmelblau function after 600 function calls. Red + yellow points are most probable places for minima. The histograms show the final distributions of parameters values (the probabilities obtained by Nested Sampling algorithm). One can see that the minima are found reasonably well.

Genetic algorithm

The Genetic algorithm (http://en.wikipedia.org/wiki/Genetic_algorithm) is a probabilistic search algorithm that iteratively transforms a set (called a population) of parameters vectors, each with an associated fitness value, into a new population of offspring objects using the Darwinian principle of natural selection and using operations that are patterned after naturally occurring genetic operations, such as crossover (recombination) and mutation. The advantages of the algorithm are

- investigates the landscape of optimization function
- converges quickly
- lends itself to parallelization.

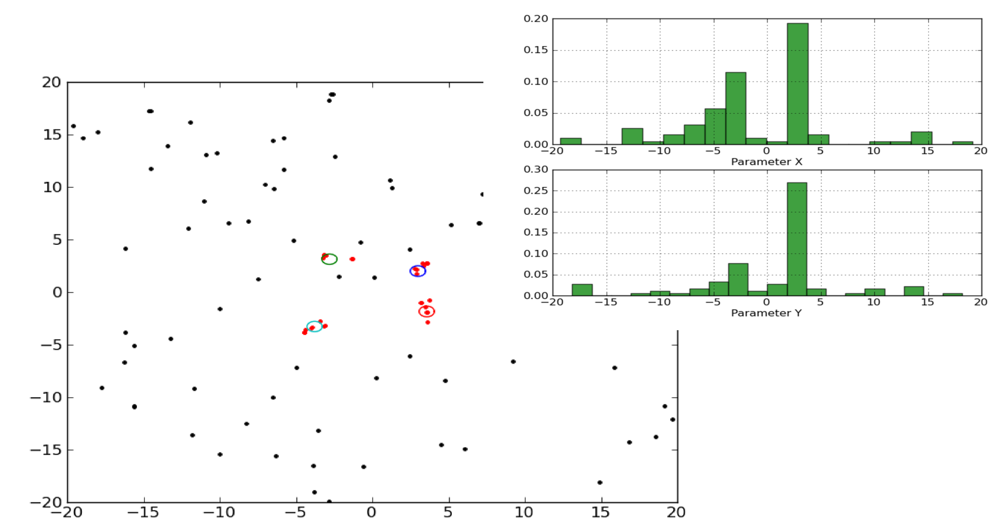

Figure 4: Genetic algorithm results on the Himmelblau function after 10 iterations (about 300 function calls). Red points - the final population and black points - initial population. The histograms show the final distributions of parameters values (the probabilities obtained by Nested Sampling algorithm).

Nested Sampling

The Nested Sampling algorithm (http://www.inference.phy.cam.ac.uk/bayesys) is a variant of the Monte Carlo technique based on a Bayesian approach, which reduces the dimensionality of the space through integration, allowing not only to find multiple solutions, but also to estimate the confidence intervals of parameters values and to evaluate Bayesian evidence. In parameter estimation, the evidence factor is usually ignored, since it is independent of the parameters, but the evidence automatically implements Occam's razor: a simpler theory with compact parameter space will have a larger evidence then more complicated one, unless the latter is significantly better at explaining the data. Thus, the evidence can be used for definition of a number of free parameters in the model. Posterior inferences can be generated using Bayes theorem from the nested sampling process. Then they can be used to calculate inferences of posterior parameters values such as means, standard deviations, covariances and etc., or to construct marginalized posterior distributions. The advantages of the algorithm are

- finds multiple minima
- gives posterior inferences
- typically requires many times fewer samples than standard MCMC methods.

The disadvantage is that the algorithm is not easy to parallelize.

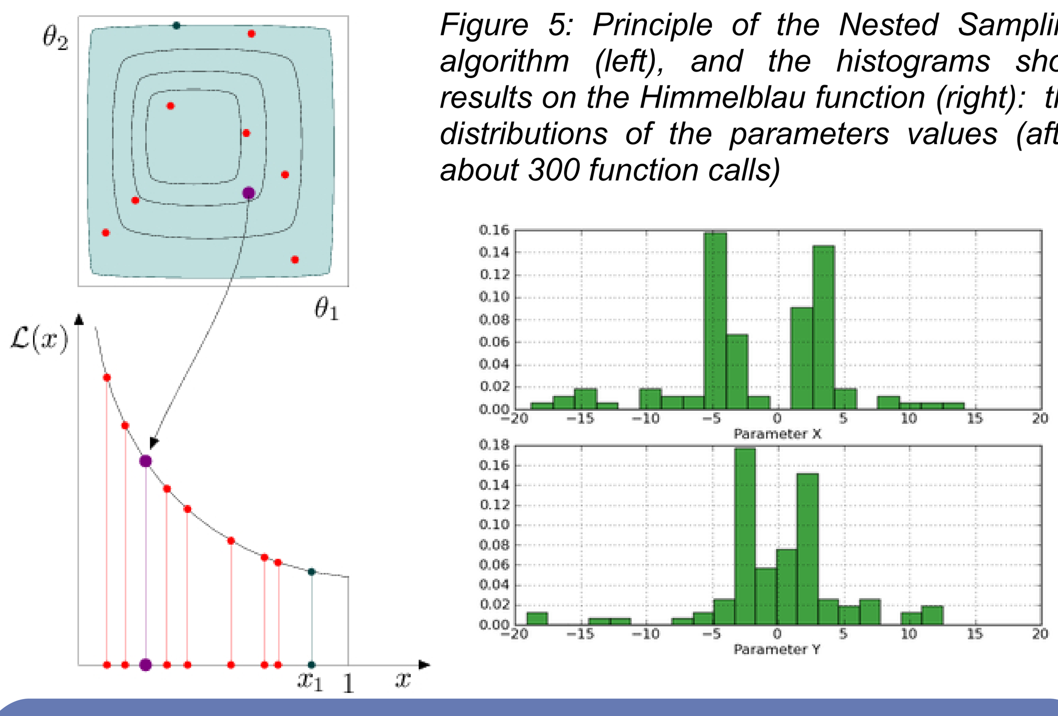

Figure 5: Principle of the Nested Sampling algorithm (left), and the histograms show results on the Himmelblau function (right): the distributions of the parameters values (after about 300 function calls)

Outlook

MAGIX complies with the data structures and reduction tools of ALMA (Atacama Large Millimeter Array), and aims to be a tool accompanying observations assembled with the ALMA interferometer, but can be used on any, even non-astronomical, data sets. Beta testers are welcome (contact Prof. Peter Schilke schilke@ph1.uni-koeln.de), and stay tuned for updates!