Instruction Manual for TAMCMC-C++
Version 1.3.1

Othman Benomar

Contents

1 Purpose of TAMCMC-C++

TAMCMC stands for Tempered Adaptive Markov Chain Monte Carlo and is a multipurpose code designed to analyse asteroseismic data. As the extension suggests, the code is written in C++, but is based on my PhD work[1], but is also described in Benomar et al. (2009). The algorithm scheme is taken from Atchadé (2006). It is designed to

- Automatically adjust the covariance matrix of the proposal law in order to optimally sample a statistical criteria with a MCMC algorithm.
- Use the local gradient to enhance the exploration properties of the algorithm. This is this 'Langevin' part of the algorithm that lead to the algorithm name: MALA (Metropolis-Hasting-Langevin Algorithm).

The current version of the code does not implement the 'Langevin' scheme, because the derivative is not known a priori when analysing seismic data. However, this might be implemented in the future.

This version of the code is designed to perform power spectrum fitting of Main-sequence solar-like stars. It comes with a suite of models that allows to perform global fits of individual pulsations. Although this was not tested thoroughly, It has a also a module to handle the fit of mode envelopes, considering a Gaussian envelope over a noise background.

2 Basic structure of the code and efficiency

The code is built so that most user do not require any coding skills for analysing basic asteroseismic data. Furthermore, if it is needed to add/modify models, this is fairly easy due to the modular structure of the code: The core computational part is independent from the model/priors/data/statistical criteria that should be used. Adding or Modifying those requires only to edit few files,

- `./tamcmc/sources/models.cpp` that contains all the models that must computed
- `./tamcmc/sources/config.cpp` that contains the list of models, priors and statistics that are accepted.
- `./tamcmc/sources/model_def.cpp` that specify which function and what arguments must be used depending on the chosen model

TAMCMC-C++ is build such that it is highly efficient in sampling the pdf of a distribution. Parallelisation of the parallel chains is possible by OpenMP. Hence, when analysing typically $\approx 10^5$ data points[2] the typical time to compute 1 Million samples is of a couple of days on standard quad core CPU.

[1] `http://www.theses.fr/2010PA112309`

[2] Typical number of the longest *Kepler* observations.

3 General setup

4 Model setup

5 External tools

References

Atchadé, Y. F.
2006. An adaptive version for the metropolis adjusted langevin algorithm with a truncated drift. *Meth. Comp. In Applied Probab.*, 8:235.

Benomar, O., T. Appourchaux, and F. Baudin
2009. The solar-like oscillations of HD 49933: a Bayesian approach. *Astron. Astrophys.*, 506:15–32.