

Working Title: Improving Sampling Efficiency on Convex Polytopes

Forward simulations

Forward simulations in ^{13}C -MFA are maps from the flux space $V \subset \mathbb{R}^m$ to the isotope labelling space $U \subset \mathbb{R}^m$

$$f: V \rightarrow U$$

For steady-state scenarios, i.e. scenarios where we assume the organism to reside in a metabolic steady state, a forward simulation can be computed by solving the algebraic system

$$du/dt = 0$$

where du/dt are the time derivatives of the isotope labellings, also called the mass balance equations.

For instationary scenarios the, the isotope labelling data is complemented with timestamps of the measurement. Hence the time derivatives have to be integrated up the desired time t_0 to obtain $u(t_0)$, the labelling data predicted at time t_0 .

In general, solving the system of ordinary differential equations takes longer time than solving the algebraic system $du/dt = 0$, making instationary scenarios computationally more challenging. In fact, the time needed to perform a single forward simulation can be two orders of magnitude higher in the instationary case over the stationary one [1].

It is therefore of interest to minimize the numbers of evaluations of the instationary forward simulation. One possible way to avoid its evaluation is to approximate its value e.g. by using regression models. The natural question, which arises is on what data to fit the regression model. Creating samples is not only costly, but the samples need to represent the true underlying function and hence would have to be taken from a convergent Markov chain. This again contradicts our goal to run exactly such a Markov chain using the regression model.

On the other hand, computing models online, i.e. while sampling, would violate the fundamental Markov property destroying the theoretical basis for the convergence of our chain. The field of adaptive Markov chains deals with this issue, where the proposal distribution is adapted online from the chains history, but in a fashion s.t. convergence is still guaranteed.

Delayed-Acceptance & Multi-Stage MCMC

Multi-Stage (or apparently also sometimes called Delayed-Acceptance/Rejection) algorithms use differently costly forward simulations cascaded after each other [4,5]. The idea is to quickly compute a coarse value as an approximate to the finer ones and only consider promising looking areas. A first proposal is accepted or not using the energy function of the coarse simulation result and only if the move gets accepted by the first stage the next finer simulation is evaluated. This saves costly computations of exact solutions for proposal steps which might have been rejected anyway.

The natural questions which arises here is what kind of coarse and fine versions of the model functions to apply. In our case, where a forward simulation consist of solving a system of nonlinear ordinary differential equations using numerical integration a natural choice would be the integration step size or - when using adaptive schemes - the error tolerance. In particular, one could arrange a sequence of different error tolerances where each proposal stage solves the system again using a lower tolerance. According to [3], a decrease of accuracy from $1e-3$ to $1e-9$ might lead to an increase in computational time taken in an order of magnitude.

Further choices, which are more closely related to our domain, would be stationary and instationary measurements, which would form coarse and fine model respectively. However often this data is not available in mixed form, meaning that a isotope labelling experiment only either contains measures from the stationary or instationary phase of the experiment.

Also, MS measurements are easier to simulate than MS-MS measurements, hence one could also apply a two-stage proposal scheme here.

Another, again more general approach, is the application of linear or higher order models to approximate the function at least locally. The question arises what data to use to setup such models. Evaluation of the gradient locally is expensive in general, since it needs exact evaluations of the forward simulation. Such evaluations are obviously available from the previous samples. However, using this data to obtain a model of the forward simulation, we would violate the Markov property.

[2] proposed a method where gradient evaluations are taken from a neural network trained on a pre-run of the chain. The approximate evaluations are used within some trust-criteria, meaning that as soon as one starts to distrust the approximation (e.g. because to many rejections happened), the exact evaluation is used again and as soon as the approximation has matched the exact evaluation well enough, one switches back to using the approximation. The overall concept can most probably be used with any approximate model of the gradient, not necessarily a neural network, but for instance regression models applied to the prerun data.

An issue one might encounter using this approach, is that a model fitted on a prerun of a chain, which did not run until convergence, may fail to approximate the target function well in some non-covered areas. Although precautions are taken above, by not solely relying on the model, but only within trust criteria, this may obstruct convergence to the correct target function.

Adaptive Markov Chains

The previous point about using online models fitted on the chains data leads to the broad field of Adaptive MCMC methods, which enable usage of the chains history to tune the proposal distribution s.t. less proposals get rejected, but sill the whole chain remains convergent.

...

Active Subspaces

Informally speaking, active subspace methods try to identify the subspaces of the parameter space which explain most of the variation of some target function $f: U \rightarrow \mathbb{R}$, with $U \subset \mathbb{R}^n$. I presume, that they can also be considered as performing a principal component analysis on the samples generated by a chain and hence identifying the dimensions which explain most of the variance in the data. The value in this information lies in the fact, that a direction, along which the target function does not (or only minimally) change might not be worth exploring at all, since no information can be gained. Parameters, for which its partial derivative is zero over the whole domain, are called non-identifiable. In the case, where the active subspace may not be parallel to any parameter axes, a basis transformation towards an orthogonal basis (like the eigenbasis) might be considered, yielding linear combinations of non-identifiable parameters.

The information on the active subspaces could either be collected and used in an offline or online fashion, again meaning that one either tries to gain this information a priori using e.g. preruns of the chain or other techniques, or on the other hand by computing it while running the Markov chain and integrate the gained knowledge in an adaptive way, s.t. convergence may still be guaranteed.

Action Plan

As a starter, an easy and straight-forward strategy could be to consider, implement and test the multi-stage MCMC method using different error thresholds for the adaptive ODE solver as a first proof of concept. Careful software design could enable us to replace the coarse/fine models with arbitrary ones, allowing us to test more combinations from what was mentioned previously.

Preconditioning of MCMC methods using a priori computed data like active subspaces and approximate models combined with the trust criterion approach used in [1] also seem like a powerful, but rather simply implemented approach. Using approximate models of the function to cheaply evaluate its gradient could also be used to improve computational efficiency of gradient-based approaches like the CSmMALA proposal move. However, this most likely seems to remain compatible with the multi-stage approach, possibly allowing again for arbitrary combinations of techniques.

Generally, I consider the techniques above very powerful, since they work on a meta plane above the considerations of efficient proposal distributions on convex polytopes. This allows to consider the techniques more as a framework, where arbitrary proposal moves, which suffice our needs, may be used.

The true excitement however obviously lies in the field of the adaptive methods, which however might leave the scope of a Master Thesis, considering the broad possibilities already outlined before.

References

- [1] Muller et al.: A neural network assisted Metropolis adjusted Langevin algorithm. <https://doi.org/10.1515/mcma-2020-2060>
- [2] Fredirk
- [3] Antion
- [4] Efendiev et al.: Preconditioning MCMC simulations using coarse-scale models. <https://epubs.siam.org/doi/abs/10.1137/050628568>
- [5] Haario et al.: DRAM: Efficient Adaptive MCMC. <https://link.springer.com/article/10.1007/s11222-006-9438-0>