A Torsten case study: Evaluating the benefits of cross-chain communication during the warmup phase of Hamiltonian Monte Carlo (HMC) simulation

Yi Zhang and William R. Gillespie Metrum Research Group, Tariffville, CT USA

METRUM RESEARCH GROUP

Objectives

Stan is a probabilistic programming language and Bayesian inference engine (HMC simulation) [1]. Torsten is a collection of Stan functions and features to support pharmacometric modeling [2]. It includes an experimental cross-chain warmup scheme that performs dynamic warmup adaptation by chain communication and aggregation. We extend previous work on multilevel parallelization [3] by evaluating the efficiency and sampling quality of fittings using a large number parallel chains.

Methods

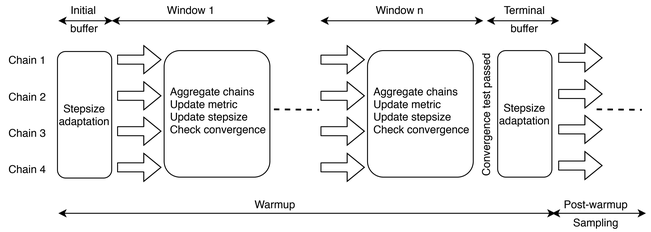

Figure 1: Proposed *cross-chain* warmup algorithm

The following is the proposed algorithm (see [3] for more details):

1. With a fixed window size w, initiate warmup with stepsize adaptation on n_{chain} chains.
2. At the end of a window, aggregate joint posterior probability from all the chains and calculate the corresponding potential scale reduction coefficients ($\hat{R}$) and effective sample sizes (ESS) [4]. For example, with default window size $w = 100$, when warmup reaches iteration 200, calculate the potential scale reduction coefficients $\hat{R}^i$ and ESS^i for $i = 1, 2$, so that $\hat{R}^1$ and ESS^1 are based on warmup iteration 1 to 200, and $\hat{R}^2$ and ESS^2 are based on warmup iteration 101 to 200.
3. At the end of window n with predefined target value $\hat{R}^0$ and ESS^0, from $1, \ldots, n$, select j with maximum ESS^j and calculate a new metric using samples from corresponding windows. Determine *convergence* by checking if $\hat{R}^j < \hat{R}^0$ and $\text{ESS}^j > \text{ESS}^0$. If converges, move to post-warmup sampling, otherwise repeat step 2.
4. After convergence the parallel chains that participate warmup begin post–warmup sampling. There is no cross–chain communication during sampling.

To evaluate its performance, we apply the algorithm to a hierarchical PKPD model. With a population of 16, the model uses the one–compartment model to describe each subject's PK and an effective compartment for PD, so that the clearance Cl_j absorption rate ka_j, and volumne of distribution V_j characterizes the PK, and ke_j the effective compartment, for subject j. These parameters formulate a linear ordinary different equation (ODE)

$$\frac{dy}{dt} = \begin{bmatrix} -ka_j & 0 & 0 \\ ka_j & -Cl_j/V_j & 0 \\ 0 & ke_j & -ke_j \end{bmatrix} y$$

and is solved by Torsten's `pmx_solve_linode` function. The drug effect is then linked to the effective compartment through a sigmoid E_{max} model, characterized by baseline E_0 and efficacy E_{max}. The population level parameters $\widehat{Cl}$, $\widehat{V}$, $\widehat{ka}$, $\widehat{ke}$, $\widehat{E_{\text{max}}}$, and $\widehat{E_0}$ are all assigned with lognormal priors.

The model is fitted using both standard Stan practice ("standard run") as well as the proposed parallel algorithm ("parallel run"). A standard run consists of 4 chains with 1000 warmup iterations and 1000 sampling iterations in each chain. A parallel run uses a fixed target $\hat{R}^0 = 1.05$ and one of the three target $\text{ESS}^0 = 200, 400, 600$. With each target ESS, we run the model with 4, 8, 16, and 32 parallel chains, so that each chain is processed by an individual parallel process, therefore $n_{\text{chain}} = n_{\text{proc}}$, where n_{proc} indicates the number of parallel processes.

From each run, we collect time spent on warmup $\text{Time}_{\text{warmup}}$, on sampling $\text{Time}_{\text{sampling}}$, and $\text{Time}_{\text{total}}$=$\text{Time}_{\text{warmup}}$+$\text{Time}_{\text{sampling}}$. We expect cross–chain warmup would reduce $\text{Time}_{\text{warmup}}$, and an increased n_{chain} would increase total ESS (aggregated from all the chains) in a fixed $\text{Time}_{\text{sampling}}$.

Results

The quality of warmup can be examined through ESS of the joint log–posterior density `lp_` per time (seconds) in each chain. As shwon in Fig. 2, in general the sampling efficiency in parallel runs becomes less efficient. On the other hand, sampling with a large (>4) n_{chain} produces higher total ESS in total (Fig. 3), thus is a strategy one can explore when the goal is to obtain a given number of effective samples. Note that in the figures we differentiate ESS_{bulk} and ESS_{tail}, as suggested in [4].

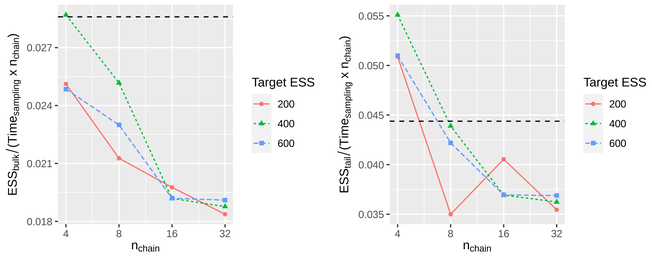

Figure 2: ESS_{bulk} (left) and ESS_{tail} (right) of `lp_` per post–warmup sampling time (second) in each chain. The horizontal dashed line indicates the standard run baseline.

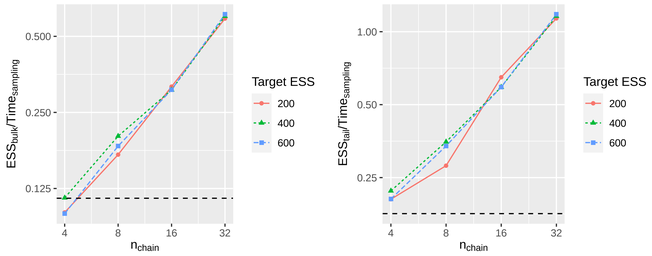

Figure 3: ESS_{bulk} (left) and ESS_{tail} (right) of `lp_` per post–warmup sampling time (second) from all the chains. The horizontal dashed line indicates the standard run baseline.

The benefit of cross–chain warmup is apparent when we replace $\text{Time}_{\text{sampling}}$ with $\text{Time}_{\text{total}}$ in the evaluation, since $\text{Time}_{\text{warmup}}$, a significant proportion of the total time cost, is reduced by the new warmup algorithm. Shown in Fig. 4, now the ESS per time in each chain becomes either close to standard run (ESS_{bulk}) or significantly improved (ESS_{tail}).

To examine the parallel performance we define

$$\text{Speedup} = \left(\frac{\text{ESS}}{\text{Time}_{\text{total}}}\bigg|_{\text{parallel run}}\right) \Big/ \left(\frac{\text{ESS}}{\text{Time}_{\text{total}}}\bigg|_{\text{standard run}}\right), \qquad \text{parallel efficiency} = 4\frac{\text{Speedup}}{n_{\text{chain}}}.$$

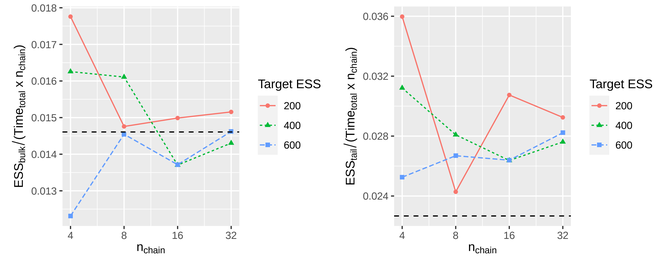

Figure 4: ESS_{bulk} (left) and ESS_{tail} (right) of `lp_` per time (second) from all the chains. The horizontal dashed line indicates the standard run baseline.

Table 1 and Fig. 5 show that the parallel setup is very performant. Compared to the standard run, the parallel runs benefit from both reduced warmup time during dynamic warmup and the increased total ESS, and they are able to achieve super–linear efficiency (parallel efficiency > 1).

Table 1: Parallel efficiency based on ESS_{bulk} (left) and ESS_{tail} (right) of `lp_`

target ESS	$n_{\text{chain}} = 4$	$n_{\text{chain}} = 8$	$n_{\text{chain}} = 16$	$n_{\text{chain}} = 32$
200	1.22	1.01	1.03	1.04
400	1.11	1.10	0.938	0.979
600	0.843	0.996	0.939	1.00

target ESS	$n_{\text{chain}} = 4$	$n_{\text{chain}} = 8$	$n_{\text{chain}} = 16$	$n_{\text{chain}} = 32$
200	1.59	1.07	1.36	1.29
400	1.38	1.24	1.16	1.22
600	1.11	1.18	1.16	1.25

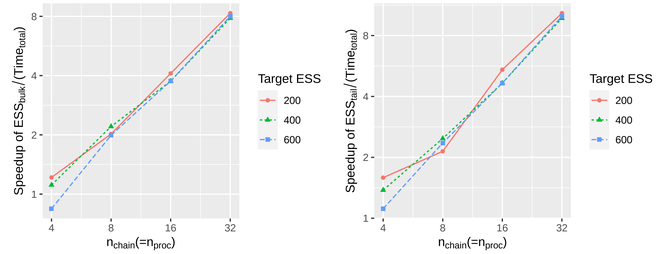

Figure 5: ESS_{bulk} (left) and ESS_{tail} (right) of `lp_` speedup.

Conclusions and future work

Using the Cross–chain warmup and an increased number of chains proves to be an efficient strategy to reduce HMC running cost and improve parallel efficiency. From the benchmark we notice

1. Setting target ESS to 200 is sufficient to achieve comparable sampling performance. In general, an enlarged target ESS provides limited benefit.
2. The warmup quality for hierarchical PKPD model can suffer from cross–chain warmup when using a large number of chains, but the performance in terms of total ESS makes up for this loss.
3. Using a large number of parallel chains is an attractive option to achieve a given ESS, as the parallel runs become highly efficient in general.

References

[1] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 76(1):1–32, January 2017.

[2] Torsten: library of C++ functions that support applications of Stan in Pharmacometrics. `https://github.com/metrumresearchgroup/Torsten`.

[3] Yi Zhang and William R. Gillespie. Speed up population bayesian inference by combining cross-chain warmup and within-chain parallelization. In *the 11th American Conference on Pharmacometrics*, November 2020.

[4] Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Burkner. Rank-Normalization, Folding, and Localization: An Improved $\widehat{R}$ for Assessing Convergence of MCMC. *Bayesian Analysis*, pages 1 – 38, 2021.

yiz@metrumrg.com | billg@metrumrg.com | metrumrg.com/all-publications