How to build & run Torsten's MPI jobs

Yi Zhang

February 22, 2019

1 Build

Torsten's `pop_pk_generalOdeModel_bdf`, `pop_pk_generalOdeModel_adams`, `pop_pk_generalOdeModel_rk45` functions support MPI runs. To build a Stan model that utilizes these functions in `cmdstan`, add the following to `cmdstan/make/local`

```
TORSTEN_MPI = 1
-include $(MATH)make/setup_torsten.mk
CXXFLAGS += $(CXXFLAGS_MPI) -isystem /usr/local/mpich3/include
LDFLAGS += $(LDFLAGS_MPI)
CC=mpicxx
CXX=mpicxx
```

and make the model file from `cmdstan` folder.
This feature is currently not available to `R` interface.

2 Run

To run the current model `pop_pk_twocpt`, in the model folder, do

```
mpiexec -n 2 ./pop_pk_twocpt sample num_samples=250 num_warmup=250 data
 ↪  file=pop_pk_twocpt.data.R init=pop_pk_twocpt.init.R
```

Here we are running the job using 2 processes. Since the population size is 2, adding more processes will not benefit the solution.

3 Load balancing

Torsten's MPI solvers use static balancing to distribute the load. For a model with population size 4, if we solve it using 2 processes by issuing `mpiexec -n 2`, the load will be distributed as

process	individual
1	1, 2
2	3, 4

If we solve it using 3 processes by issuing `mpiexec -n 3`, the load will be distributed as

process	individual
1	1, 2
2	3
3	4

If we solve it using 5 processes by issuing `mpiexec -n 5`, the load will be distributed as

process	individual
1	1
2	2
3	3
4	4
5	idle