Numerical ODE group/population integrators in Torsten

Yi Zhang

August 7, 2019

ODE group integrators

Single ODE system	ODE group
pmx_integrate_ode_rk45	pmx_integrate_ode_group_rk45
pmx_integrate_ode_bdf	pmx_integrate_ode_group_bdf
pmx_integrate_ode_adams	pmx_integrate_ode_group_adams

Single ODE system

```
real[,]
pmx_integrate_ode_xxx(
      f,
      real[] y0, real t0,
      real[] ts,
      real[] theta,
      real[] x_r, int[] x_i,
      ...);
```

ODE group

```
matrix
pmx_integrate_ode_group_xxx(
     f,
     real[ , ] y0, real t0,
     int[] len, real[] ts,
     real[ , ] theta,
     real[ , ] x_r, int[ , ] x_i,
     ...);
```

ODE group integrators

Single ODE system

```
real[ , ]
pmx_integrate_ode_xxx(
      f,
      real[] y0, real t0,
      real[] ts,
      real[] theta,
      real[] x_r, int[] x_i,
      ...);
```

ODE group

```
matrix
pmx_integrate_ode_group_xxx(
     f,
     real[ , ] y0, real t0,
     int[] len, real[] ts,
     real[ , ] theta,
     real[ , ] x_r, int[ , ] x_i,
     ...);
```

- `len` specifies the length of data for each subject within the above ragged arrays, and the size of `len` is the size of the population.
- The group integrators return a single matrix ragged column-wise. The number of rows equals to the size of ODE system.

Exercise

autocatalytic reaction model: ODE group version

- Change the loop with the numerical integrator to use group integrator.
- Edit/Add `cmdstan/make/local`

```
TORSTEN_MPI = 1  # flag on torsten's MPI solvers
CXXFLAGS += -isystem /usr/local/include    # path to MPI
↪   library's headers
```

- Build in `cmdstan`

```
make ../example-models/chemical_reactions/chem_group
```

- Run

```
mpiexec -n 2 -l ./chem_group sample adapt delta=0.95 random
↪   seed=1104508041 data file=chem.data.R init=chem.init.R
```

Exercise

- What does output say?
- How many cores can you use until performance saturates? Why?
- Can you do it using Stan's `map_rect`? Is there a difference in style, output, and performance?