Ruby Prototyping Environment for Molecular Orbital Calculation

Introduction

- Molecular orbital Calculation
  - Purely theoretical chemistry research
  - Designing functional material, drug discovery. etc.
- Current practical molecular orbital programs are quite large and complicated to understand
  - GAMESS (May.2013) > 1,300,000 steps
- Difficult to modify even fundamental data
  - Detailed documents are not often disclosed

*Development of **new prototyping environment** based on script language*

Based on Ruby

- *Decreasing total steps drastically*
- *Rapid prototyping*
- *Enable HPC by utilizing extended libraries*

Implementation with extended libraries

- Numerical Array class library: NArray [1)]
  - Corresponds to Numpy Python library
- MPI Module library: mpi-ruby [2)]
  - Enable to access conventional MPI library
- ACP Communication library [3)]
- Other important extended libraries
  - Molecular integrals library
  - BLAS/Lapack linear Solver

Efficiency

- Execution on computer center in Kyushu University
  - Parallel RHF/UHF calculation on Fujitsu CX400
  - Sequential RHF on Fujitsu FX10
- Comparisons between Ruby and C implementations[4)]

	Ruby	C	Ratio
Code step[4)] (lines)	409	2298	**0.18**
Execution time (sec.)	42.8	4.2	10.2
Parallelization efficiency (%)	48.4	52.5	**0.92**

512 cores, Static load balance

Ruby Language

- Dynamic type system
- Automatic memory management
- Rich standard libraries as Array, String, etc.
- Object oriented system
- Meta object protocol
- Interactive execution
- Easily extended with native libraries written by C/C++, Fortran
  - MPI/OpenMP hybrid parallel programming
  - Programming for accelerators

Molecular Integral Calc.

```
molinfo    = Molinfo.new( Input_filename )
numb_atom  = molinfo.numb_atom
numb_shell = molinfo.numb_shell
...
eris = Eri0.new
eris.init( numb_atom, numb_shell, numb_prim,
           shel_lqn, shel_tem, shel_atm, shel_add,
           atom_charg, atom_xyz, prim_exp, prim_coe,
           thr_ovch, level_print )
...
#
# G-matrix                          Shell loop
#
for ish in 0...numb_shell
 for jsh in 0..ish
  for ksh in 0..ish
   ...
   for lsh in 0..l_last
   inttype, nsize_int, g_int = eris.calc( ish, jsh, ksh, lsh )
   ...
```

- *Molecular integral class object generation*
- *Molecular integral calculations by shell indices*

Configuration Interaction Calc.

```
## Integral Transformation
fmat1dim, eris1dim, cao1dim = two_one_dimCopy( nbf, norb, fmat, eris, cao )
molint   = Molint.new( nbf, norb, fmat, eris1dim, cao1dim )
molint.trans
fmat_mo  = molint.f_mo
eris_mo  = molint.eris_mo

## CSF-based Energy Expression
drt      = Second_order_CI.new( nelec, spin, n_core, n_active, n_external )
#drt     = First_order_CI.new ( nelec, spin, n_core, n_active, n_external )
bc       = BrooksCases.new( drt )
expr1el  = bc.expr_one
expr2el  = bc.expr

## N-Electron Hamiltonian
hmat     = H_matrix.new( molinfo, fmat_mo, eris_mo, expr1el, expr2el )
hmat.mk_hmat
p hmat.data

## Large-Scale Sparse-Matrix Diagonalization
liu      = Liu.new( hmat.data, nstate, thresh )
ciene, civec = liu.solve

## Natural Orbital
no       = NO_Analysis.new
no.analysis( molinfo, n_frozen, expr1el, civec, cao, nstate )
```

- *One and two electron transformation*
- *First or second order type DRT*
- *Explicit one and two electron energy expressions*
- *Calculation of Hamiltonian matrix by molecular integrals and energy expressions*
- *Easy to access Hamiltonian matrix*
- *Large-scale sparse-matrix diagonalization*

Future Plan

- Open in this year
- Tutorials
- Implementation of MO based applications
  - FMO and Elongation methods

1) “Numerical Ruby NArray,” [On line]. Available: http://narray.rubyforge.org/index.html.ja.
2) “Seiya/ruby-mpi,” [On line]. Available: https://github.com/seiya/ruby-mpi.
3) “ACE Project,” [On line]. Avaliable: http://ace-project.kyushu-u.ac.jp/index.html.
4) RHF calculation for $(H_2O)_8$ of DZV basis using 512 cores of E5-2680 processors.

OpenFMO project
Leader: Yuichi INADOMI at Kyushu University
http://www.openfmo.org/
Contact: Hiroaki HONDA at Kyushu University
http://yebisu.cc.kyushu-u.ac.jp/~dahon/index_e.html