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

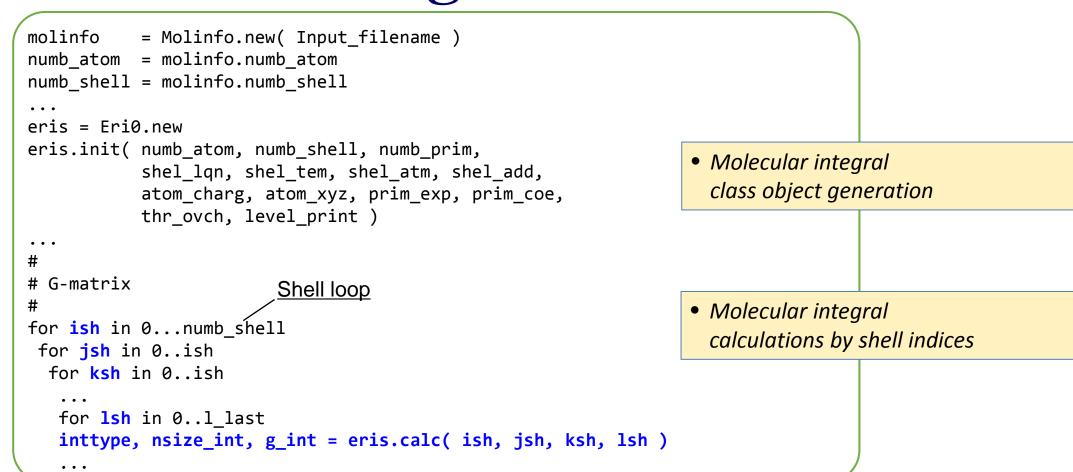
- Execution on computer center in Kyushu University
 - Parallel RHF/UHF calculation on Fujitsu CX400
 - Sequential RHF on Fujitsu FX10
- Comparisons between Ruby and C implementations⁴⁾

	Ruby	С	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 balar	nce		

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.



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
                                                                • One and two electron transformation
fmat_mo = molint.f_mo
eris_mo = molint.eris_mo
## CSF-based Energy Expression
         = Second_order_CI.new( nelec, spin, n_core, n_active, n_external )
        = First_order_CI.new ( nelec, spin, n_core, n_active, n_external
         = BrooksCases.new( drt )
                                                                • First or second order type DRT
expr1el = bc.expr_one
                                                                • Explicit one and two electron energy
expr2el = bc.expr
                                                                  expressions
## N-Electron Hamiltonian
         = H_matrix.new( molinfo, fmat_mo, eris_mo, expr1el, expr2el )
hmat.mk_hmat
                                                                • Calculation of Hamiltonian matrix
p hmat.data
                                                                  by molecular integrals and energy
## Large-Scale Sparse-Matrix Diagonalization
                                                                  expressions
         = Liu.new( hmat.data, nstate, thresh )
                                                                • Easy to access Hamiltonian matrix
ciene, civec = liu.solve
## Natural Orbital
                                                                • Large-scale sparse-matrix diagonalization
no.analysis( molinfo, n_frozen, expr1el, civec, cao, nstate )
```

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 (H₂O)₈ of DZV basis using 512 cores of E5-2680 processors.







