Atomic Structure Calculations using GRASP2K

Jörgen Ekman, Per Jönsson
Group for Materials Science and Applied Mathematics
Malmö University, Sweden

13 mars 2015

Grasp2K manual and references

- The CompAS group
  A practical guide to Grasp2K, 2015
- C. Froese-Fischer, T. Brage, P. Jönsson
  Computational Atomic Structure: An MCHF Approach
  IoP 1997
- J. Bieron
  Relativistic Quantum Theory of Atoms and Molecules: Theory and Computation
  Springer 2007

What is needed

- PC running under Linux
- Grasp2K code, available as compressed tar file
- gfortran compiler
- MPI libraries (if parallel processing)

Basics: orbital

Orbital

\[ \psi(nlsjm; r) = \frac{1}{r} \left( P(nlj; r) \Omega_{ljm}(\theta, \phi) \right) \]

- One-electron function
- Radial functions \( P(r) \), \( Q(r) \) numerically represented on a radial grid

Basics: configuration

Number of orbitals with occupation numbers, e.g.

\( 1s^2 2s^2 (2p^2)^2, \quad 1s^2 2s^2 (2p^2)^2 2p^3, \quad 1s^2 2s^2 2p^5, \)

Notation \( 1s, 2s, 2p \), \( 1s, 2s, 2p \) for \( 1s_{1/2}, 2s_{1/2}, 2p_{1/2}, 2p_{3/2} \), \( 1s_{3/2}, 2s_{1/2}, 2p_{1/2}, 2p_{3/2} \), \( 1s_{3/2}, 2s_{1/2}, 2p_{3/2} \).

In non-relativistic theory

\( 1s^2 2s^2 2p^2 \).

Basics: CSF

Configuration state function (CSF)

\[ \Phi(\gamma JM) \]

- Many-electron function
- Formed by angular couplings of the orbitals in a configuration
- Well defined parity and angular momentum
- Can be seen as a basis function
- \( \gamma \) denotes configuration and coupling

Relativistic multiconfiguration methods

General Relativistic Atomic Structure Package (Grasp2K).

P. Jönsson, G. Gaigalas, J. Bieron, C. Froese Fischer, I.P. Grant
Computer Physics Communications 184, 2197 (2013).

★ Conceptually easy
★ Can be generally applied
★ Can be used to generate massive data sets
★ Allows for systematic calculations giving uncertainty estimates
★ Extensive manual available
Basics: ASF

Atomic state function (ASF)

\[ \Psi(\Gamma_{JM}) = \sum_{\alpha=1}^{\text{NCSF}} c_{\alpha} \Phi(\gamma_{\alpha JM}) \]

- Many-electron function
- Approximate wave function of a state
- \( c_{\alpha} \) expansion (or mixing) coefficients

SD substitutions

SD substitutions from \( 1s^22s^22p^2 \) describing valence correlation
Reference configuration + rule + orbital set
\( 1s(2,1)2s(2,\ast)2p(2,\ast) \)
3s, 3p, 3d
- occupation (in red)
- \( i \) = inactive, no substitutions
- \( \ast \) = no restrictions on occupation after substitutions
- substitutions to \{1s, 2s, 3s, 2p, 3p, 3d\}

Tasks related to atomic structure calculations

Identify the states you want to target (NIST, Chianti, other work)
- Define nuclear properties (charge, distribution, spin, etc)
- Generate CSF expansion
- Perform angular integration and set up energy expression (sum angular coefficient times radial integrals)
- Initial estimates of radial orbitals
  \( P(a, r), Q(a, r), P(b, r), Q(b, r) \) etc on a grid
- Determine radial orbitals and expansion coefficients self-consistently (MCDHF)
- Add transverse Breit and QED and determine only expansion coefficients (RCI)
- Compute additional properties (transition rates etc)
- Generate LaTeX tables, plots etc

Important files in Grasp2K

- isodata: nuclear data
- rcsf.inp: list of CSFs arranged according to symmetry
- rwfn.inp: initial estimates of the radial functions
- rwfn.out: radial functions after a successful run
- rmix.out: expansion (mixing) coefficients for CSFs after a successful run

After successful MCDHF run above files saved as
- \textit{name.c}: list of CSFs
- \textit{name.w}: radial wave functions
- \textit{name.m}: expansion (mixing) coefficients
  where the user determines the name.
These files define the wave functions and are used as input to other programs.

Program groups in Grasp2K

Generate list of CSFs (rcsf....)
- rcsfexcitation: define rules for SD substitutions, wrapper program
- rcsfgenerate: generates the lists of CSFs based on rules from wrapper program
- rcsfinteract: removes unimportant CSFs based on analysis of couplings

Estimates of radial orbitals (rwfn....)
- rwfneestimate: estimates radial orbitals, several sources can be used
- rwfmschfmcdf: convert non-relativistic radial orbitals to relativistic ones
Program groups in Grasp2K

Program groups in Grasp2K

Compute and display wave functions and energies

- **rangular**: angular integration, defines energy expressions for rmcdhf
- **rmcdhf**: determines both radial orbitals and expansion coefficients
- **rci**: assumes radial orbitals available, determine expansion coefficients
- **jj2lsj**: transforms to LSJ-coupling
- **rlevels**: displays energies and labels

Application programs, computes transition rates, Lande factors etc from CSF file, radial wave function file, mixing coefficient file

- **rbio**: transforms radial orbitals to a biortonormal basis
- **rtransition**: computes transition parameters
- **rhfs**: computes hfs and Lande factors

Jörgen Ekman, Per Jönsson Group for Materials Science and Applied Mathematics Malmö University, Sweden

Computed by {text}

Atomic Structure Calculations using GRASP2K

Jörgen Ekman, Per Jönsson Group for Materials Science and Applied Mathematics Malmö University, Sweden

Computation steps even states

- **rci**: transverse, QED, mass, \( J = 0, 1, 2 \), \( J = 0, 1, 2 \)
- **jj2lsj**: transforms to LSJ coupling
- **more evenMR.lsj.lbl**: display LSJ-composition
- **rlevels**: display energies and labels, indata from evenMR.cm
- **rhfs**: Lande factors, indata just the name evenMR
- **more evenMR.ch**: display Lande factors

Jörgen Ekman, Per Jönsson Group for Materials Science and Applied Mathematics Malmö University, Sweden

Computation steps, odd states

- **rci**: transverse, QED, mass, \( J = 0, 1, 2 \), \( J = 0, 1, 2 \)
- **jj2lsj**: transform to LSJ coupling
- **more oddMR.lsj.lbl**: display LSJ-composition
- **rlevels**: display energies and labels, indata from evenMR.cm
- **rhfs**: Lande factors, indata just the name oddMR
- **more oddMR.ch**: display Lande factors

Jörgen Ekman, Per Jönsson Group for Materials Science and Applied Mathematics Malmö University, Sweden

Computation steps, transition parameters

- **rbio**: transform to biortonormal radial orbitals, indata just the names evenMR, oddMR
- **rtransition**: indata just the names evenMR, oddMR, multipoles E1, M2
- **more evenMR,oddMR.ct.lsj**
Improve wave functions, even states

- `rcfexcitation`, `rcfgenerate`: SD-substitutions describing valence and core-valence from \(1s^22s^22p^2\) to \(3s, 3p, 3d\)
- `rxfnestimate`: \(1s, 2s, 2p\) from old \(3s, 3p, 3d\) hydrogenic
- `rangular`: angular integration
- `rmcdhf`: optimize on \(J = 0.1 - 4, J = 1.1 - 2, J = 2.1 - 4\) only new orbitals optimized, not spectroscopic
- `rsave`: save and give name even3

Improve wave functions, odd states

- `rcfexcitation`, `rcfgenerate`: SD-substitutions describing valence and core-valence from \(1s^22s^22p^2\) to \(3s, 3p, 3d\)
- `rxfnestimate`: \(1s, 2s, 2p\) from old \(3s, 3p, 3d\) hydrogenic
- `rangular`: angular integration
- `rmcdhf`: optimize on \(J = 0.1 - 4, J = 1.1 - 2, J = 2.1 - 4\) only new orbitals optimized, not spectroscopic
- `rsave`: save and give name odd3

Computational steps, transition parameters

- `rbiotransform`: transform to biorthonormal radial orbitals, indata just the names even3, odd3
- `rtransition`: indata just the names even3, odd3, multipoles E1, M2
- `more even3.odd3.ct.1s`
Thank you for your attention.