HF Documentation

Roeland Neugarten

March 2024

1 Modules

- molecular_structure
  Contains everything pertaining to the internal structure of the considered molecule(s).
  - type molecular_structure_t
  - subroutine add_atoms_to_molecule()
- compute_integrals
  Contains the subroutines to compute the one-electron and two-electron integrals necessary for the Hartree-Fock calculation.
  - subroutine compute_1e_integrals()
  - subroutine generate_2int()
  - subroutine compute_2e_integrals()
- diagonalization
  Contains the subroutines to solve the general eigenvalue problem necessary to find the coefficients matrix.
  - subroutine solve_genev()
  - subroutine diagonalize()
- ao_basis
  Contains the datatypes and subroutines pertaining to the basis set used to describe the electrons.
  - type basis_func_info_t
  - type basis_set_info_t
  - subroutine clear_gto()
  - subroutine clear_basis()
  - subroutine add_shell_to_basis()
  - function n_ang()

- define_system
  Contains subroutines that read user input and process it into the basis set and molecular structure necessary for the Hartree-Fock calculation.
  - define_basis()
  - read_xyz()

2 Datatypes

- basis_func_info_t
  Provides the description for a singular basis function.
  - integer orb_momentum: Orbital momentum (0,1,2,3,...)
  - integer atom_number: The index of the atom on which the basis function is centered.
  - integer atom_element: The atomic number of the atom on which the basis function is centered.
  - integer n_primitives: For contracted basis sets: The number of primitives.
  - integer n_contracted: For contracted basis sets: The number of contractions made.
  - real coord(1:3): Coordinates of the center of the basis function.
  - real exponent(:): Exponents
  - real coeff(:): Coefficients of the contractions.
- basis_set_info_t
  Provides information for the basis set: The set of basis functions.
  - integer nshells: The number of shells contained in the basis set.
  - integer nao: The number of basis functions.
  - integer basis_angular: Determines coordinate system. 1=cartesian, 2=spherical.
  - basis_func_info_t pointer gtos(:)
- molecular_structure_t
  Contains the information of the molecule(s) that are considered in the calculation. Used for the potential energy calculations.
  - integer num_atoms: The number of atoms that make up the molecule.
  - real charge(:): The charge of each atom nucleus.
  - real coord(:,:): The coordinates of each atom.

3 Base Subroutines

The following subroutines are provided by the skeleton code and have not been modified. Subroutines called by subroutines are not considered.

- add_atoms_to_molecule(molecule, add_charge, add_coord)
  Allows for the addition of extra atoms to molecule.
  - molecular_structure_t molecule: The molecule to which the new atoms are added.
  - real add_charge(:): The charges/atom numbers of the added atoms.
  - real add_coord(:,:) The coordinates of the added atoms.
- add_shell_to_basis(ao_basis, angular, coord, alpha, exponents, coefficients)
  Adds the GTO basis functions corresponding to the given shell to the specified basis set. Uses the function n_ang() using the specified angular momentum to generate the correct number of magnetic angular momenta $m_l = -l, -l+1, ...l-1, l+2$.
  - basis_set_info_t ao_basis: The basis set to which the shells must be added.
  - integer angular: The azimuthal quantum number dictating the type of orbitals (s, p, d)
  - real coord(3): The coordinates around which the shell is centered (usually the nucleus of an atom)
  - OPTIONAL real alpha: Constant specifying the shape of uncontracted basis functions.
  - OPTIONAL real exponents(:): Constant specifying the shape of contracted basis functions.
  - OPTIONAL real coefficients(:): Contraction coefficients of contracted basis functions.
- compute_1e_integrals(property, ao_basis_bra, ao_basis_ket, ao_integrals, molecule)
  Computes the integrals corresponding to either the overlap matrix, the kinetic energy or the potential energy.
  - char property: Dictates which integral is evaluated. Must be either "OVL" for overlap, "KIN" for kinetic, or "POT" for potential.
  - basis_set_info_t ao_basis_bra: The basis set(s) corresponding to the "bra" of the integral.
  - basis_set_info_t ao_basis_ket: The basis set(s) corresponding to the "ket" of the integral.
  - real ao_integrals: Output containing the evaluated integral(s).
  - OPTIONAL molecular_structure_t molecule: The molecule to which the integral belongs. Necessary only for the potential energy.
- generate_2int(ao_basis, ao_integrals)
  Generates and computes the two-electron integrals for the basis set. Calls the subroutine compute_2e_integrals, where the actual computation is performed.
  - basis_set_info_t ao_basis: The basis set containing the functions to be integrated.
  - real ao_integrals(:,:,:,:): Output, providing the solutions to all integrals.
- solve_genev(matrix, metric, eigenvectors, eigenvalues)
  Solves generalized eigenvalue problem using Lowdins transformation to orthonormal basis. Calls the subroutine diagonalize.

- real matrix(:,:): Matrix operator.
- real metric(:,:): That to which is matrix is applied to yield the eigenvectors.
- eigenvectors(:,:): Output. The eigenvector solutions of the problem.
- eigenvalues(:): Output. The eigenvalues corresponding to the eigenvector solutions of the problem.

4 Modified Subroutines

The following subroutines were modified heavily or written from scratch when compared to the skeletal code. Not that the subroutine define_molecule() has been in its entierity replaced by read_xyz(), which also immediately calls define_basis().

- define_basis(ao_basis, charge, coords)
  Takes the arrays containing information about the atom and generates shells to be added to the basis set. Calls subroutine add_shell_to_basis(). Distinguishes between hydrogen atoms (charge=1) and other atoms, which are given, respectively, 3 s-orbitals or 5 s-, 3 p-, and 1 d-orbitals. All orbitals are Gaussian-Type.
  - basis_set_info_t ao_basis: The basis set to which the atoms are to be added.
  - real charge(:): 1D array containing the charges of the atoms to be added to the basis set.
  - real coords(3,:): 2D array containing the (x,y,z) coordinates of each atom.
- read_xyz(molecule, ao_basis, n_occ, conv)
  Processes the contents of "mol.xyz" into the necessary data types. Contains an array "table" containing the elements of the periodic table, which yields the charges of the elements by using FORTRAN's findloc function. The index of an element is necessarily its atomic number, thus its charge. Supports elements up to the second period. Calls subroutines define_basis() and add_atom_to_molecule() after processing the xyz file.
  - molecular_structure_t molecule molecule: Output. The structure of the considered molecule.
  - basis_set_info_t ao_basis: Output. The basis set of the considered system.
  - integer n_occ: Output. The number of occupied orbitals, found by summing the charges of all atoms and dividing by two. Must be an even number to meet the requirements of restricted HF.
  - real conv: The convergence requirement, dictated by the comment line in the xyz file.

5 User Input/Output

5.1 User Input

User input is provided through the file "mol.xyz" provided in the main directory. It is a standard xyz file. The first line contains the total number of atoms, though this is not used in the present

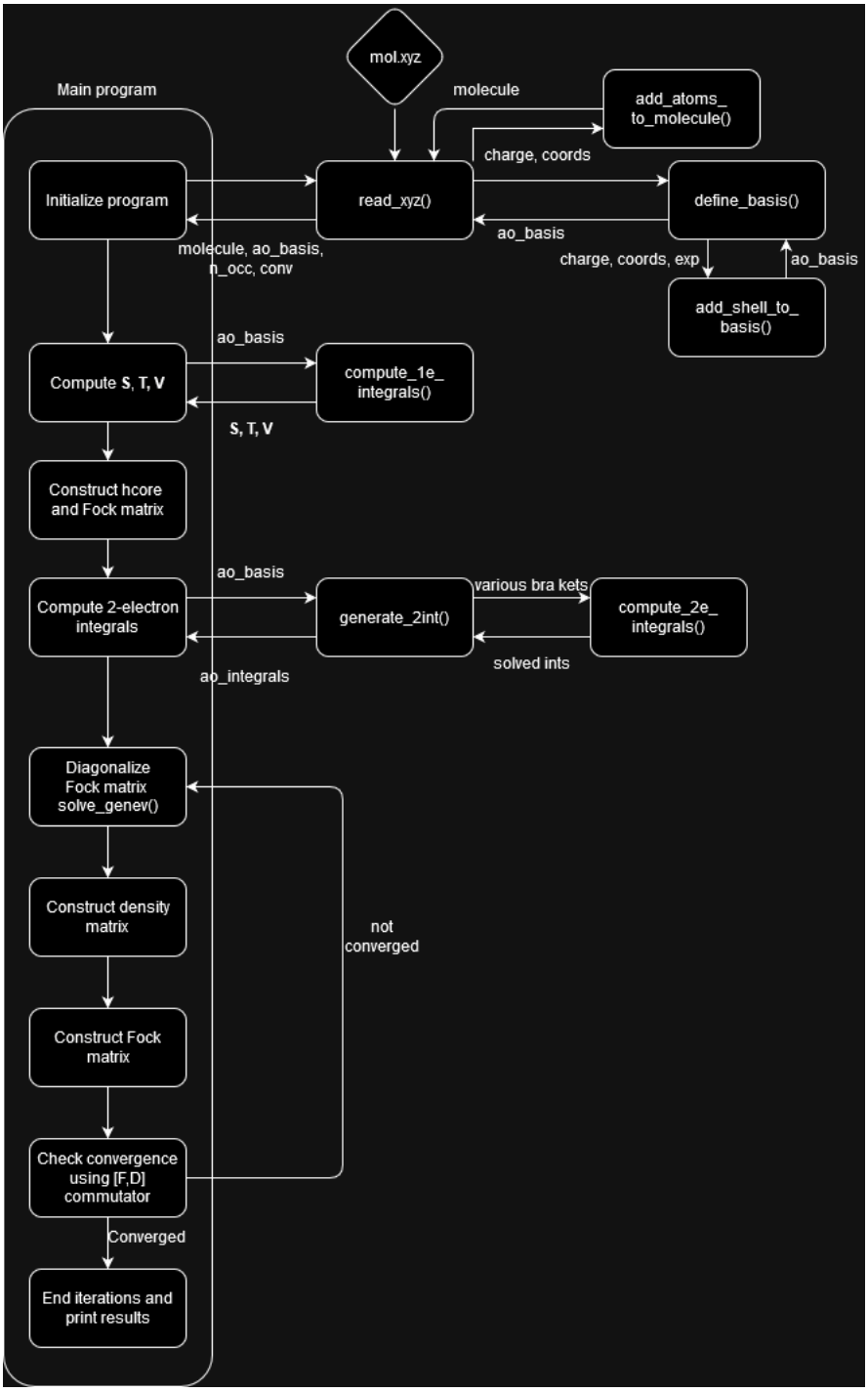
program and therefore need not necessarily be correct. The second line, which is usually a comment, here provides the level of convergence desired by the user. After that, there is a variable number of lines containing first the element of that atom, followed by its (x,y,z) coordinates. The program processes this file into the basis set necessary for the calculation, and an instance of the molecular_structure_t datatype, which provides the description for the potential energy. Note that the program is exclusively a restricted Hartree-Fock program, and thus accepts only systems with an even number of electrons. The program is terminated if this requirement is not met. Furthermore, the program supports elements up to neon.

5.2 User Output

The user is provided with the following information:

- The system to be considered, with the element types and (x,y,z) coordinates.
- The number of occupied and total orbitals.
- The convergence requirement, submitted by the user.
- The energy, $\Delta D^{(n)}$ and CPU time per iteration.
- The final $\Delta D^{(n)}$, final E_{HF} and total calculation time.

6 Flowchart


mol.xyz
Main program
molecule
add_atoms_
to_molecule()
charge, coords
Initialize program
read_xyz()
define_basis()
ao_basis
molecule, ao_basis,
n_occ, conv
charge, coords, exp
ao_basis
add_shell_to_
basis()
ao_basis
Compute S, T, V
compute_1e_
integrals()
S, T, V
Construct hcore
and Fock matrix
ao_basis
Compute 2-electron
integrals
generate_2int()
various bra kets
compute_2e_
integrals()
solved ints
ao_integrals
Diagonalize
Fock matrix
solve_genev()
Construct density
matrix
not
converged
Construct Fock
matrix
Check convergence
using [F,D]
commutator
Converged
End iterations and
print results