December 6, 2014
18:00

Contents

1 SCF

This is Version 1 of the Hartree-Fock theory implemented for closed shells (RHF) and open shells (UHF-DODS) calculations.

NAME SCF

Perform LCAO-MO-SCF calculation on a molecule.

SYNOPSIS

```
double precision function scf(H, C, nbasis, nelec, nfile,
irite, damp, interp, E, HF, V, R, Rold, Ubar, eps)
integer nbasis, nelec, nfile, irite
double precision damp, E
double precision H(ARB), C(ARB), HF(ARB), V(ARB), R(ARB)
double precision Rold(ARB), Ubar(ARB), eps(ARB)
```

DESCRIPTION

Perform LCAO-MO calculation of either closed-shell RHF type or more general open-shell (real) UHF-DODS type. The method is traditional Roothan repeated diagonalizations of Hartree-Fock matrix until self-consistency is reached:

$$\mathbf{F} \cdot \mathbf{C} = \mathbf{S} \cdot \mathbf{C} \cdot \epsilon$$

ARGUMENTS

H Input: One-electron Hamiltonian of size (`nbasis x nbasis`), i.e., matrix elements of one-electron operator

C Input/Output: An initial MO matrix - it must at least orthigonalize the basis. Normally, it is simply the orthogonalization matrix $\mathbf{S}^{-\frac{1}{2}}$. On output the SCF **C** matrix is placed here.

nbasis Input: the number of *spatial* orbitals in the basis (i.e., half of the number of the spin-basis set functions if `nelec` ≥ 0)

nelec Input: The number of electrons in the system.

nfile The electron-repulsion file unit.

itite Channel number for convergence information or zero if this information is not necessary.

damp Hartree-Fock damping parameter.

interp Interpolation parameter. If 0 no interpolation will be undertaken.

HF Output: for use as the Fock matrix

V Workspace:

R Output: Density matrix

Rold Workspace:

Ubar Workspace:

eps Output: orbital energies (first `nelec` are the occupied orbitals)

E Output: Total HF electronic energy

RETURNS

`YES` if the calculation is converged in `MAX_SCF_ITERATIONS`
`NO` if no convergence is met. Typical usage: `if ( SCF(.....) .EQ. YES ) then`
`output succesful calculation`

SEE ALSO

```
  /* STRUCTURES */
"scf.f" 1 ≡
  @m YES 0
  @m NO 100
  @m ERR −10
  @m OK 10
  @m END_OF_FILE −1
  @m NOT_END_OF_FILE 55
  @m LAST_BLOCK 12
  @m NOT_LAST_BLOCK −12
        /* OPERATIONAL CONSTANTS */
  @m ARB 1
  @m BYTES_PER_INTEGER 4
  @m LEAST_BYTE 1
  @m NO_OF_TYPES 20
  @m INT_BLOCK_SIZE 20
  @m MAX_BASIS_FUNCTIONS 255
  @m MAX_PRIMITIVES 1000
  @m MAX_CENTRES 50
  @m MAX_ITERATIONS 60
  @m UHF_CALCULATION 11
  @m CLOSED_SHELL_CALCULATION 21
        /* OUTPUT STREAM UNITS */
  @m ERROR_OUTPUT_UNIT 6
  @m ERI_UNIT 17
  @m MAX_ITERATIONS 50

    integer function SCF(H, C, nbasis, nelec, nfile, irite, damp, interp, E, HF, V, R, Rold, Cbar,
          epsilon, crit)
      ⟨ Global SCF Declarations #2 ⟩
      ⟨ Internal SCF Declarations #3 ⟩
      ⟨ Select SCF Type #4 ⟩
      ⟨ Set initial matrices and counters #5 ⟩
      do while((icon ≠ 0) ∧ (kount < MAX_ITERATIONS))
        ⟨ Sigle SCF iteration #6 ⟩
      end do
      ⟨ Write the output result #7 ⟩
      ⟨ Formats #8 ⟩
      return
    end
```

⟨ Global SCF Declarations 1.1 ⟩ ≡
```
implicit double precision (a − h, o − z)
integer nbasis, nelec, nfile, irite
integer interp
double precision H(ARB), C(ARB), HF(ARB), V(ARB), R(ARB)
double precision Rold(ARB), Cbar(ARB)
double precision epsilon(ARB)
double precision E, damp, crit
```
This code is used in section 1.

⟨ Internal SCF Declarations 1.2 ⟩ ≡
```
integer scftype, kount, maxit, nocc, m, mm, i
double precision term, turm, Rsum
double precision zero, half
data zero, half /0.0 · 10^+00D, 0.5 · 10^+00D/
```
This code is used in section 1.

⟨ Select SCF Type 1.3 ⟩ ≡
```
if (nelec > zero) then    /* closed shell case */
  scftype = CLOSED_SHELL_CALCULATION
  nocc = abs(nelec / 2)
  m = nbasis
else    /* open shell case */
  scftype = UHF_CALCULATION
  nocc = nelec
  m = nbasis * 2
  call spinor(H, nbasis)
  call spinor(C, nbasis)
end if
```
This code is used in section 1.

⟨ Set initial matrices and counters 1.4 ⟩ ≡
```
    /* basis set size */
m = m * m
do i = 1, mm
  R(i) = zero;
  Rold(i) = zero
end do
SCF = YES
kount = 0
icon = 100
```
This code is used in section 1.

⟨ Sigle SCF iteration 1.5 ⟩ ≡

```
kount = kount + 1
E = zero;
icon = 0
do i = 1, mm
  HF(i) = H(i)
  E = E + R(i) * HF(i)
enddo
call scfGR(R, HF, m, nfile)
do i = 1, mm
  E = E + R(i) * HF(i)
enddo

if (scftype ≡ UHF_CALCULATION)
  E = half * E

write (ERROR_OUTPUT_UNIT, 200) E

call gtprd(C, HF, R, m, m, m)
call gmprd(R, C, HF, m, m, m)
call eigen(HF, Cbar, m)
do i = 1, m
  epsilon(i) = HF(m * (i − 1) + i)
enddo
call gmprd(C, Cbar, V, m, m, m)
call scfR(V, R, m, nocc)
Rsum = zero
do i = 1, mm
  turm = R(i) − Rold(i)
  term = dabs(turm)
  Rold(i) = R(i)
  C(i) = V(i)
  if (term > crit)
    icon = icon + 1
  Rsum = Rsum + term
  if (kount < interp)
    R(i) = R(i) − damp * turm
enddo
```

This code is used in section 1.

⟨ Write the output result 1.6 ⟩ ≡

```
write (ERROR_OUTPUT_UNIT, 201) Rsum, icon
if ((kount ≡ MAX_ITERATIONS) ∧ (icon ≠ 0)) then
  write (ERROR_OUTPUT_UNIT, 204)
else
  write (ERROR_OUTPUT_UNIT, 202) kount
  write (ERROR_OUTPUT_UNIT, 203) (epsilon(i), i = 1, nocc)
endif
```

This code is used in section 1.

⟨ Formats 1.7 ⟩ ≡

```
200: format (" Current Electronic Energy = ", f12.6)
201: format (" Convergence in R = ", f12.5, i6, "  Changing")
202: format (" SCF converged in", i4, " iterations")
203: format (" Orbital Energies ", (7f10.5))
204: format (" SCF did not converged... quitting")
```

This code is used in section 1.

2 UTILITIES

spinor

"scf.f" 2.2 ≡

```
subroutine spinor(H, m)
  double precision H(*)
  integer m

  double precision zero
  integer i, j, ij, ji, ip, jp, ijp, ijd, nl, n
  data zero/0.0 · 10^+00D/

  n = 2 * m;
  nl = m + 1

  do i = 1, m
    do j = 1, m
      ij = m * (j − 1) + i;
      ip = i + m;
      jp = j + m
      ijp = n * (jp − 1) + ip;
      H(ijp) = H(ij)
    end do
  end do

  do i = 1, m
    do j = 1, m
      ip = i + m;
      jp = j + m;
      ijp = n * (jp − 1) + ip
      ijd = n * (j − 1) + i;
      H(ijd) = H(ijp)
    end do
  end do

  do i = 1, m
    do j = nl, n
      ij = n * (j − 1) + i;
      ji = n * (i − 1) + j
      H(ij) = zero
      H(ji) = zero
    end do
  end do

  return
end
```

⟨ Formats 1.7 ⟩ Used in section 1.
⟨ Global SCF Declarations 1.1 ⟩ Used in section 1.
⟨ Internal SCF Declarations 1.2 ⟩ Used in section 1.
⟨ Select SCF Type 1.3 ⟩ Used in section 1.
⟨ Set initial matrices and counters 1.4 ⟩ Used in section 1.
⟨ Sigle SCF iteration 1.5 ⟩ Used in section 1.
⟨ Write the output result 1.6 ⟩ Used in section 1.

COMMAND LINE: `"fweave -C3 scf.web"`.
WEB FILE: `"scf.web"`.
CHANGE FILE: `(none)`.
GLOBAL LANGUAGE: FORTRAN.