December 5, 2014
1:50

Contents

1 genint **1**

1 genint

Routine to evaluate S, T,V i.e. H and a two-electron integral file

"genints.f" 1 ≡
@m *ARB* 1
@m *YES* 0
@m *NO* 1
@m ERR −1

@m *BYTES_PER_INTEGER* 4
@m *LEAST_BYTE* 1

@m *END_OF_FILE* −1

@m *NO_OF_TYPES* 20
@m *INT_BLOCK_SIZE* 20

@m *LAST_BLOCK* 1
@m *NOT_LAST_BLOCK* 0

@m *ERROR_OUTPUT_UNIT* 6

@m *MAX_BASIS_FUNCTIONS* 255
@m *MAX_PRIMITIVES* 1000
@m *MAX_CENTRES* 50

@m *MAX_ITERATIONS* 60

@m *UHF_CALCULATION* 10
@m *CLOSED_SHELL_CALCULATION* 20

subroutine *genint*(*ngmx*, *nbfns*, *eta*, *ntype*, *ncntr*, *nfirst*, *nlast*, *vlist*, *ncmx*, *noc*, *S*, *H*, *nfile*, *ifPSE*);
double precision *S*(∗), *H*(∗);
integer *pointer*, *last*, *ifPSE*;
double precision *eta*(*ngmx*, 5), *vlist*(*ncmx*, 4);
integer *ntype*(∗), *nfirst*(∗), *nlast*(∗), *ncntr*(∗), *nfile*;

{
double precision *val*, *crit*, *ovltot*, *kintot*;
double precision *generi*, *genoei*;
integer *i*, *j*, *k*, *l*, *ltop*, *ij*, *ji*, *mu*;
integer *nr*(*NO_OF_TYPES*, 3);
data *nr*/0, 1, 0, 0, 2, 0, 0, 1, 1, 0, 3, 0, 0, 2, 2, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 2, 0, 1, 0, 1, 0, 3, 0, 1, 0, 2, 2, 0, 1, 1, 0, 0, 0, 1, 0, 0, 2, 0, 1, 1, 0, 0, 3, 0, 1, 0, 1, 2, 2, 1/;
data *crit*/$1.0 \cdot 10^{-08}$D/;

mu = 0; /∗ tag for use later perhaps, when integrals are marked for special purposes ∗/

/∗ One-electron integrals ∗/

do *i* = 1, *nbfns*;
{
do *j* = 1, *i*;
{
ij = (*j* − 1) ∗ *nbfns* + *i*;
ji = (*i* − 1) ∗ *nbfns* + *j*;

```
      if (ifPSE ≡ NO)
        H(ij) = genoei(i, j, eta, ngmx, nfirst, nlast, ntype, nr, NO_OF_TYPES,
            vlist, noc, ncmx, ovltot, kintot);    /* else H(ij) = genpse (i,j,eta,ngmx,nfirst,nlast,ntype,
            nr,NO_OF_TYPES, vlist, noc, ncmx, ovltot, kintot);H(ji)=H(ij);S(ij)=ovltot;S(ji)=ovltot;}}
    /* H now contains T + V */
  rewind nfile;
  pointer = 0;
  last = NO;    /* initialisation for putint */
  do i = 1, nbfns;
    {
    do j = 1, i;
      {
      do k = 1, i;
        {
        ltop = k;
        if (i ≡ k)
          ltop = j;    /* get upper limit for l right! */
        do l = 1, ltop;
          {
          if (l ≡ nbfns)
            last = YES;    /* last integral */
          val = generi(i, j, k, l, 0, eta, ngmx, nfirst, nlast, ntype, nr, NO_OF_TYPES);
          if (dabs(val) < crit)
            next;    /* this assumes that the last integral is never zero */
          call putint(nfile, i, j, k, l, mu, val, pointer, last);
          }
        }
      }
    }
  return;
  }
```

ARB: 1.

BYTES_PER_INTEGER: 1.

CLOSED_SHELL_CALCULATION: 1.
crit: 1.

dabs: 1.

END_OF_FILE: 1.
ERR: 1.
ERROR_OUTPUT_UNIT: 1.
eta: 1.

generi: 1.

genint: 1.
genoei: 1.

H: 1.

i: 1.
ifPSE: 1.
ij: 1.
INT_BLOCK_SIZE: 1.

j: 1.
ji: 1.

k: 1.
kintot: 1.

l: 1.
last: 1.
LAST_BLOCK: 1.
LEAST_BYTE: 1.
ltop: 1.

MAX_BASIS_FUNCTIONS: 1.
MAX_CENTRES: 1.
MAX_ITERATIONS: 1.
MAX_PRIMITIVES: 1.
mu: 1.

nbfns: 1.
ncmx: 1.
ncntr: 1.
nfile: 1.
nfirst: 1.
ngmx: 1.
nlast: 1.
NO: 1.
NO_OF_TYPES: 1.
noc: 1.
NOT_LAST_BLOCK: 1.
nr: 1.
ntype: 1.

ovltot: 1.

pointer: 1.
putint: 1.

S: 1.

UHF_CALCULATION: 1.

val: 1.
vlist: 1.

YES: 1.

COMMAND LINE: `"fweave genints.web"`.
WEB FILE: `"genints.web"`.
CHANGE FILE: `(none)`.
GLOBAL LANGUAGE: RATFOR.