

(25 March 2000)

## Section 4 - Further Information

This section of the manual contains both references, and hints on how to do things. The following is a list of the topics covered:

- Computational References. 4-2
- Basis Set References, and descriptions. 4-6
- How to do RHF, ROHF, UHF, and GVB calculations. 4-12
  - General considerations 4-12
  - Direct SCF 4-13
  - SCF convergence accelerators 4-15
  - High spin open shell SCF (ROHF) 4-17
  - Other open shell SCF cases (GVB) 4-18
  - True GVB perfect pairing runs 4-20
  - The special case of TCSCF 4-21
  - A caution about symmetry 4-21
- How to do MCSCF and CI calculations. 4-22
  - MCSCF implementation 4-22
  - Orbital updates 4-23
  - CI coefficient optimization 4-24
  - CSF CI 4-26
  - determinant CI 4-26
  - starting orbitals 4-28
  - references 4-30
- Second order perturbation theory. 4-32
- Density Functional Theory 4-36
- Geometry Searches and Internal Coordinates. 4-37
  - Quasi-Newton searches 4-38
  - The nuclear hessian 4-39
  - Coordinate choices 4-40
  - The role of symmetry 4-43
  - Practical matters 4-44
  - Saddle points 4-45
  - Mode following 4-46
- Intrinsic Reaction Coordinate (IRC) methods 4-47
- Gradient Extremals 4-50
- Continuum solvation methods: SCRF, PCM and COSMO 4-53
- Effective Fragment Potential method 4-58
  - Terms in an EFP 4-58
  - Constructing an EFP 4-59
  - Current Limitations 4-59
  - Practical Hints 4-60
  - QM/MM across covalent bonds 4-60
  - References 4-62
- MOPAC calculations within GAMESS 4-64
- Molecular Properties, and conversion factors. 4-65
- Localization tips. 4-66
- Transition moments and spin-orbit coupling. 4-70

## Computational References

### GAMESS -

M.W.Schmidt, K.K.Baldrige, J.A.Boatz, S.T.Elbert, M.S.Gordon, J.J.Jensen, S.Koseki, N.Matsunaga, K.A.Nguyen, S.Su, T.L.Windus, M.Dupuis, J.A.Montgomery  
J.Comput.Chem. 14, 1347-1363 (1993 )

### HONDO -

These papers describes many of the algorithms in detail, and much of these applies also to GAMESS:

"The General Atomic and Molecular Electronic Structure System: HONDO 7.0" M.Dupuis, J.D.Watts, H.O.Villar, G.J.B.Hurst Comput.Phys.Comm. 52, 415-425(1989 )

"HONDO: A General Atomic and Molecular Electronic Structure System" M.Dupuis, P.Mougenot, J.D.Watts, G.J.B.Hurst, H.O.Villar in "MOTTECC: Modern Techniques in Computational Chemistry" E.Clementi, Ed. ESCOM, Leiden, the Netherlands, 1989 , pp 307-361.

"HONDO: A General Atomic and Molecular Electronic Structure System" M.Dupuis, A.Farazdel, S.P.Karna, S.A.Maluendes in "MOTTECC: Modern Techniques in Computational Chemistry" E.Clementi, Ed. ESCOM, Leiden, the Netherlands, 1990 , pp 277-342.

M.Dupuis, S.Chin, A.Marquez in Relativistic and Electron Correlation Effects in Molecules, G.Malli, Ed. Plenum Press, NY 1994.

### sp integrals and gradient integrals -

J.A.Pople, W.J.Hehre J.Comput.Phys. 27, 161-168(1978) H.B.Schlegel, J.Chem.Phys. 90, 5630-5634(1989 )

### spdfg integrals -

"Numerical Integration Using Rys Polynomials" H.F.King and M.Dupuis  
J.Comput.Phys. 21,144(1976 )

"Evaluation of Molecular Integrals over Gaussian Basis Functions"  
M.Dupuis,J.Rys,H.F.King J.Chem.Phys. 65,111-116(1976 )

"Molecular Symmetry and Closed Shell HF Calculations" M.Dupuis and H.F.King  
Int.J.Quantum Chem. 11,613(1977 )

"Computation of Electron Repulsion Integrals using the Rys Quadrature Method"  
J.Rys,M.Dupuis,H.F.King J.Comput.Chem. 4,154-157(1983 )

### spdfg gradient integrals -

"Molecular Symmetry. II. Gradient of Electronic Energy with respect to Nuclear Coordinates" M.Dupuis and H.F.King J.Chem.Phys. 68,3998(1978 )  
although the implementation is much newer than this paper.

### spd hessian integrals -

"Molecular Symmetry. III. Second derivatives of Electronic Energy with respect to Nuclear Coordinates" T.Takada, M.Dupuis, H.F.King J.Chem.Phys. 75, 332-336 (1981 )

### Effective core potentials (ECP) -

C.F.Melius, W.A.Goddard Phys.Rev.A, 10,1528-1540(1974)

L.R.Kahn, P.Baybutt, D.G.Truhlar J.Chem.Phys. 65, 3826-3853 (1976 )

M.Krauss, W.J.Stevens Ann.Rev.Phys.Chem. 35, 357-385(1985 )

J.Breidung, W.Thiel, A.Komornicki Chem.Phys.Lett. 153, 76-81(1988 )

See also the papers listed for SBKJC and HW basis sets.

- RHF -  
C.C.J. Roothaan Rev.Mod.Phys. 23, 69(1951 )
- UHF -  
J.A. Pople, R.K. Nesbet J.Chem.Phys 22, 571 (1954 )
- GVB and OCBSE-ROHF -  
F.W.Bobrowicz and W.A.Goddard, in Modern Theoretical Chemistry, Vol 3,  
H.F.Schaefer III, Ed., Chapter 4.
- ROHF -  
R.McWeeny, G.Diercksen J.Chem.Phys. 49,4852-4856(1968 )  
M.F.Guest, V.R.Saunders, Mol.Phys. 28, 819-828(1974 )  
J.S.Binkley, J.A.Pople, P.A.Dobosh Mol.Phys. 28, 1423-1429 (1974 )  
E.R.Davidson Chem.Phys.Lett. 21,565(1973 )  
K.Faegri, R.Manne Mol.Phys. 31,1037-1049(1976 )  
H.Hsu, E.R.Davidson, and R.M.Pitzer J.Chem.Phys. 65, 609(1976 )
- closed shell 2nd order Moller-Plesset and MP2 gradient -  
J.A.Pople, J.S.Binkley, R.Seeger Int. J. Quantum Chem. S10, 1-19(1976 )  
M.J.Frisch, M.Head-Gordon, J.A.Pople, Chem.Phys.Lett. 166, 275-280(1990 )  
the implementation is M.Dupuis, S.Chin, A.Marquez above
- open shell MP2, so called ZAPT method -  
T.J.Lee, D.Jayatilaka Chem.Phys.Lett. 201, 1-10(1993)  
T.J.Lee, A.P.Rendell, K.G.Dyall, D.Jayatilaka  
J.Chem.Phys. 100, 7400-7409(1994)
- open shell MP2, so called RMP method -  
P.J.Knowles, J.S.Andrews, R.D.Amos, N.C.Handy, J.A.Pople  
Chem.Phys.Lett. 186, 130-136 (1991)  
W.J.Lauderdale, J.F.Stanton, J.Gauss, J.D.Watts,  
R.J.Bartlett Chem.Phys.Lett. 187, 21-28(1991)
- Multiconfigurational quasidegenerate perturbation theory -  
H.Nakano, J.Chem.Phys. 99, 7983-7992(1993 )
- GUGA CI -  
B.Brooks and H.F.Schaefer J.Chem. Phys. 70,5092(1979 )  
B.Brooks, W.Laidig, P.Saxe, N.Handy, and H.F.Schaefer, Physica Scripta  
21,312(1980 ).
- Direct SCF -  
J.Almlöf, K.Faegri, K.Korsell J.Comput.Chem. 3, 385-399 (1982 )  
M.Haser, R.Ahrlrichs J.Comput.Chem. 10, 104-111 (1989 )
- DIIS (Direct Inversion in the Iterative Subspace) -  
P.Pulay J.Comput.Chem. 3, 556-560(1982 )
- SOSCF -  
T.H.Fischer, J.Almlöf, J.Phys.Chem. 96 ,9768-74(1992 )  
G. Chaban, M. W. Schmidt, M. S. Gordon Theor.Chem.Acc. 97 , 88-95(1997 )

## MOPAC 6 -

J.J.P.Stewart J.Computer-Aided Molecular Design 4, 1-105 (1990 )  
References for parameters for individual atoms may be found on the printout from your runs.

## RHF/ROHF/TCSCF coupled perturbed Hartree Fock -

Single Configuration SCF Second Derivatives on a Cray H.F.King, A.Komornicki in  
Geometrical Derivatives of Energy Surfaces and Molecular Properties  
P.Jorgensen J.Simons, Ed. D.Reidel, Dordrecht, 1986, pp 207-214.  
Y.Osamura, Y.Yamaguchi, D.J.Fox, M.A.Vincent, H.F.Schaefer J.Mol.Struct. 103,  
183-186 (1983 )  
M.Duran, Y.Yamaguchi, H.F.Schaefer J.Phys.Chem. 92, 3070-3075 (1988 )  
A New Dimension to Quantum Chemistry Y.Yamaguchi, Y.Osamura, J.D.Goddard,  
H.F.Schaefer Oxford Press, NY 1994

## EVVRSP in memory diagonalization -

S.T.Elbert Theoret.Chim.Acta 71,169-186(1987 )

## Davidson eigenvector method -

E.R.Davidson J.Comput.Phys. 17, 87(1975 ) and "Matrix Eigenvector Methods"  
p. 95 in Methods in Computational Molecular Physics ed. by G.H.F.Diercksen  
and S.Wilson

## Energy localization-

C.Edmiston, K.Ruedenberg Rev.Mod.Phys. 35, 457-465(1963 ).  
R.C.Raffenetti, K.Ruedenberg, C.L.Janssen, H.F.Schaefer, Theoret.Chim.Acta 86,  
149-165(1993)

## Boys localization-

S.F.Boys, Quantum Science of Atoms, Molecules, and Solids P.O.Lowdin, Ed,  
Academic Press, NY, 1966, pp 253-262.

Population localization - J.Pipek, P.Z.Mezey J.Chem.Phys. 90, 4916(1989 ).

Mulliken Population Analysis - R.S.Mulliken J.Chem.Phys. 23, 1833-1840, 1841-  
1846, 2338-2342, 2343-2346

## Modified Virtual Orbitals (MVOs) -

C.W.Bauschlicher, Jr. J.Chem.Phys. 72,880-885(1980 )

## Bond orders and valences -

M.Giambagi, M.Giambagi, D.R.Grepel, C.D.Heymann J.Chim.Phys. 72, 15-  
22(1975 )  
I.Mayer, Chem.Phys.Lett., 97,270-274(1983 ), 117,396(1985 ).  
M.S.Giambiagi, M.Giambagi, F.E.Jorge Z.Naturforsch. 39a, 1259-73(1984 )  
I.Mayer, Theoret.Chim.Acta, 67,315-322(1985 ).  
I.Mayer, Int.J.Quantum Chem., 29,73-84(1986 ).  
I.Mayer, Int.J.Quantum Chem., 29,477-483(1986 ).

The same formula (apart from a factor of two) may also be seen in equation  
31 of the second of these papers (the bond order formula in the 1st of these is not  
the same formula):

T.Okada, T.Fueno Bull. Chem. Soc. Japan 48, 2025-2032(1975 )

T.Okada, T.Fueno Bull. Chem. Soc. Japan 49, 1524-1530(1976 )

## Geometry optimization and saddle point location -

J.Baker J.Comput.Chem. 7, 385-395(1986 ).

T.Helgaker Chem.Phys.Lett. 182, 503-510(1991 ).

P.Culot, G.Dive, V.H.Nguyen, J.M.Ghuysen Theoret.Chim.Acta 82, 189-205(1992 ).

## Vibrational Analysis in Cartesian coordinates -

W.D.Gwinn J.Chem.Phys. 55,477-481(1971 )

## Normal coordinate decomposition analysis -

J.A.Boatz and M.S.Gordon, J.Phys.Chem. 93, 1819-1826(1989 ).

## Raman intensity -

A.Komornicki, J.W.McIver J.Chem.Phys. 70, 2014-2016(1979)

G.B.Bacskay, S.Saebo, P.R.Taylor Chem.Phys. 90, 215-224(1984)

## parallelization in GAMESS -

for SCF, the main GAMESS paper quoted above.

T.L.Windus, M.W.Schmidt, M.S.Gordon, Chem.Phys.Lett., 216 , 375-379(1993 )

T.L.Windus, M.W.Schmidt, M.S.Gordon, Theoret.Chim.Acta 89 , 77-88 (1994 )

T.L.Windus, M.W.Schmidt, M.S.Gordon, in Parallel Computing in Computational Chemistry, ACS Symposium Series 592, Ed. by T.G.Mattson, ACS Washington, 1995 , pp 16-28.

K.K.Baldrige, M.S.Gordon, J.H.Jensen, N.Matsunaga, M.W.Schmidt, T.L.Windus, J.A.Boatz, T.R.Cundari ibid, pp 29-46.

G.D.Fletcher, M.W.Schmidt, M.S.Gordon, Adv.Chem.Phys. 110, 267-294 (1999)

G.D.Fletcher, M.W.Schmidt, B.M.Bode, M.S.Gordon, Comput.Phys.Commun. In press.

## MacMolPlt -

B.M.Bode, M.S.Gordon J.Mol.Graphics Mod. 16, 133-138(1998)

## RESC -

T.Nakajima, K.Hirao Chem.Phys.Lett. 302, 383-391(1999)

## NESC -

K.G.Dyall, J.Chem.Phys. 100, 2118-2127(1994)

K.G.Dyall J.Chem.Phys. 106, 9618-9626(1997)

K.G.Dyall J.Chem.Phys. 109, 4201-4208(1998)

K.G.Dyall, T.Enevoldsen J.Chem.Phys. 111, 10000-10007(1999)

## Basis Set References

An excellent review of the relationship between the atomic basis used, and the accuracy with which various molecular properties will be computed is:

E.R.Davidson, D.Feller Chem.Rev. 86, 681-696(1986 ).

STO-NG	H-Ne	Ref. 1 and 2
	Na-Ar,	Ref. 2 and 3 **
	K,Ca,Ga-Kr	Ref. 4
	Rb,Sr,In-Xe	Ref. 5
	Sc-Zn,Y-Cd	Ref. 6

- 1) W.J.Hehre, R.F.Stewart, J.A.Pople J.Chem.Phys. 51, 2657-2664(1969 ).
- 2) W.J.Hehre, R.Ditchfield, R.F.Stewart, J.A.Pople J.Chem.Phys. 52, 2769-2773(1970 ).
- 3) M.S.Gordon, M.D.Bjorke, F.J.Marsh, M.S.Korth J.Am.Chem.Soc. 100, 2670-2678(1978 ).
- \*\* the valence scale factors for Na-Cl are taken from this paper, rather than the "official" Pople values in Ref. 2.
- 4) W.J.Pietro, B.A.Levi, W.J.Hehre, R.F.Stewart, Inorg.Chem. 19, 2225-2229(1980 ).
- 5) W.J.Pietro, E.S.Blurock, R.F.Hout,Jr., W.J.Hehre, D.J. DeFrees, R.F.Stewart Inorg.Chem. 20, 3650-3654(1980 ).
- 6) W.J.Pietro, W.J.Hehre J.Comput.Chem. 4, 241-251(1983 ).

MINI/MIDI H-Xe Ref. 9

- 9) Gaussian Basis Sets for Molecular Calculations S.Huzinaga, J.Andzelm, M.Klobukowski, E.Radzio-Andzelm, Y.Sakai, H.Tatewaki Elsevier, Amsterdam, 1984.

The MINI bases are three gaussian expansions of each atomic orbital. The exponents and contraction coefficients are optimized for each element, and s and p exponents are not constrained to be equal. As a result these bases give much lower energies than does STO-3G. The valence MINI orbitals of main group elements are scaled by factors optimized by John Deisz at North Dakota State University. Transition metal MINI bases are not scaled. The MIDI bases are derived from the MINI sets by floating the outermost primitive in each valence orbitals, and renormalizing the remaining 2 gaussians. MIDI bases are not scaled by GAMESS. The transition metal bases are taken from the lowest SCF terms in the s<sup>1</sup>, d<sup>n</sup> configurations.

3-21G	H-Ne	Ref. 10	(also 6-21G)
	Na-Ar	Ref. 11	(also 6-21G)
	K,Ca,Ga-Kr,Rb,Sr,In-Xe	Ref. 12	
	Sc-Zn	Ref. 13	
	Y-Cd	Ref. 14	

- 10) J.S.Binkley, J.A.Pople, W.J.Hehre J.Am.Chem.Soc. 102, 939-947(1980 ).  
 11) M.S.Gordon, J.S.Binkley, J.A.Pople, W.J.Pietro, W.J.Hehre J.Am.Chem.Soc. 104, 2797-2803(1982 ).  
 12) K.D.Dobbs, W.J.Hehre J.Comput.Chem. 7,359-378(1986 )  
 13) K.D.Dobbs, W.J.Hehre J.Comput.Chem. 8,861-879(1987 )  
 14) K.D.Dobbs, W.J.Hehre J.Comput.Chem. 8,880-893(1987 )

N-31G	references for	4-31G	5-31G	6-31G
	H	15	15	15
	He	23	23	23
	Li	19,24		19
	Be	20,24		20
	B	17		19
	C-F	15	16	16
	Ne	23		23
	Na-Ga			22
	Si			21 **
	P-Cl	18		22
	Ar			22
	K-Zn			25

- 15) R.Ditchfield, W.J.Hehre, J.A.Pople J.Chem.Phys. 54, 724-728(1971 ).  
 16) W.J.Hehre, R.Ditchfield, J.A.Pople J.Chem.Phys. 56, 2257-2261(1972 ).  
 17) W.J.Hehre, J.A.Pople J.Chem.Phys. 56, 4233-4234(1972 ).  
 18) W.J.Hehre, W.A.Lathan J.Chem.Phys. 56,5255-5257(1972 ).  
 19) J.D.Dill, J.A.Pople J.Chem.Phys. 62, 2921-2923(1975 ).  
 20) J.S.Binkley, J.A.Pople J.Chem.Phys. 66, 879-880(1977 ).  
 21) M.S.Gordon Chem.Phys.Lett. 76, 163-168(1980 )  
 \*\* Note that the built in 6-31G basis for Si is not that given by Pople in reference 22. The Gordon basis gives a better wavefunction, for a ROHF calculation in full atomic (Kh) symmetry,
- |  |        |             |          |
|--|--------|-------------|----------|
|  | 6-31G  | Energy      | virial   |
|  | Gordon | -288.828573 | 1.999978 |
|  | Pople  | -288.828405 | 2.000280 |
- See the input examples for how to run in Kh.
- 22) M.M.Francl, W.J.Pietro, W.J.Hehre, J.S.Binkley, M.S.Gordon, D.J.DeFrees, J.A.Pople J.Chem.Phys. 77, 3654-3665(1982 ).  
 23) Unpublished, copied out of GAUSSIAN82.  
 24) For Li and Be, 4-31G is actually a 5-21G expansion.  
 25) V.A.Rassolov, J.A.Pople, M.A.Ratner, T.L.Windus J.Chem.Phys. 109, 1223-1229(1998)

#### Extended basis sets

##### 6-311G:

- 28) R.Krishnan, J.S.Binkley, R.Seeger, J.A.Pople J.Chem.Phys. 72, 650-

654(1980 ).

valence double zeta "DZV" sets:

"DH" basis - DZV for H, Li-Ne, Al-Ar

- 30) T.H.Dunning, Jr., P.J.Hay Chapter 1 in Methods of Electronic Structure Theory, H.F.Shaefer III, Ed. Plenum Press, N.Y. 1977, pp 1-27.  
 Note that GAMESS uses inner/outer scale factors of 1.2 and 1.15 for DH's hydrogen (since at least 1983). To get Thom's usual basis, scaled 1.2 throughout:  
 HYDROGEN 1.0 x, y, z  
 DH 0 1.2 1.2

"BC" basis - DZV for Ga-Kr

- 31) J.-P.Blaudeau, M.P.McGrath, L.A.Curtiss, L.Radom, J.Chem.Phys. 107, 5016-5021(1997) "BC" basis - DZV for Ga-Kr  
 32) R.C.Binning, Jr., L.A.Curtiss J.Comput.Chem. 11, 1206-1216(1990 )

valence triple zeta "TZV" sets:

TZV for H, Li-Ne

- 40) T.H. Dunning, J.Chem.Phys. 55 (1971 ) 716-723.

TZV for Na-Ar - also known as the "MC" basis

- 41) A.D.McLean, G.S.Chandler J.Chem.Phys. 72,5639-5648(1980 ).

TZV for K,Ca

- 42) A.J.H. Wachters, J.Chem.Phys. 52 (1970 ) 1033-1036.  
 (see Table VI, Contraction 3).

TZV for Sc-Zn (taken from HONDO 7)

This is Wachters' (14s9p5d) basis (ref 42) contracted to (10s8p3d) with the following modifications

1. the most diffuse s removed;
  2. additional s spanning 3s-4s region;
  3. two additional p functions to describe the 4p;
  4. (6d) contracted to (411) from ref 43, except for Zn where Wachter's (5d)/[41] and Hay's diffuse d are used.
- 43) A.K. Rappe, T.A. Smedley, and W.A. Goddard III, J.Phys.Chem. 85 (1981 ) 2607-2611

Valence only basis sets (for use with corresponding ECPs)

SBKJC -31G splits, bigger for trans. metals (available Li-Rn)

- 50) W.J.Stevens, H.Basch, M.Krauss J.Chem.Phys. 81, 6026-6033 (1984 )  
 51) W.J.Stevens, H.Basch, M.Krauss, P.Jasien Can.J.Chem, 70, 612-630 (1992 )  
 52) T.R.Cundari, W.J.Stevens J.Chem.Phys. 98, 5555-5565(1993 )

HW -21 splits (sp exponents not shared)

transition metals (not built in at present, although they will work if you type them in).

- 53) P.J.Hay, W.R.Wadt J.Chem.Phys. 82, 270-283 (1985 )  
 main group (available Na-Xe)  
 54) W.R.Wadt, P.J.Hay J.Chem.Phys. 82, 284-298 (1985 )  
 see also  
 55) P.J.Hay, W.R.Wadt J.Chem.Phys. 82, 299-310 (1985 )

Polarization exponents

STO-NG\*



60) J.B.Collins, P. von R. Schleyer, J.S.Binkley, J.A.Pople *J.Chem.Phys.* 64, 5142-5151(1976 ).

3-21G\*. See also reference 12.

61) W.J.Pietro, M.M.Francl, W.J.Hehre, D.J.DeFrees, J.A. Pople, J.S.Binkley *J.Am.Chem.Soc.* 104,5039-5048(1982 )

6-31G\* and 6-31G\*\*. See also reference 22 above.

62) P.C.Hariharan, J.A.Pople *Theoret.Chim.Acta* 28, 213-222(1973 )

multiple polarization, and f functions

63) M.J.Frisch, J.A.Pople, J.S.Binkley *J.Chem.Phys.* 80, 3265-3269 (1984 )

STO-NG\* means d orbitals are used on third row atoms only. The original paper (ref 60) suggested  $z=0.09$  for Na and Mg, and  $z=0.39$  for Al-Cl. At NDSU we prefer to use the same exponents as in 3-21G\* and 6-31G\*, so we know we're looking at changes in the sp basis, not the d exponent.

3-21G\* means d orbitals on main group elements in the third and higher periods. Not defined for the transition metals, where there are p's already in the basis. Except for alkalis and alkali earths, the 4th and 5th row zetas are from Huzinaga, et al (ref 9). The exponents are normally the same as for 6-31G\*.

6-31G\* means d orbitals on second and third row atoms. We use Mark Gordon's  $z=0.395$  for silicon, as well as his fully optimized sp basis (ref 21). This is often written 6-31G(d) today. For the first row transition metals, the \* means an f function is added.

6-31G\*\* means the same as 6-31G\*, except that p functions are added on hydrogens. This is often written 6-31G(d,p) today.

6-311G\*\* means p orbitals on H, and d orbitals elsewhere. The exponents were derived from correlated atomic states, and so are considerably tighter than the polarizing functions used in 6-31G\*\*, etc. This is often written 6-311G(d,p) today.

The definitions for 6-31G\* for C-F are disturbing in that they treat these atoms the same. Dunning and Hay (ref 30) have recommended a better set of exponents for second row atoms and a slightly different value for H.

2p, 3p, 2d, 3p polarization sets are usually thought of as arising from applying splitting factors to the 1p and 1d values. For example, SPLIT2=2.0, 0.5 means to double and halve the single value. The default values for SPLIT2 and SPLIT3 are taken from reference 63, and were derived with correlation in mind. The SPLIT2 values often produce a higher (!) HF energy than the singly polarized run, because the exponents are split too widely. SPLIT2=0.4,1.4 will always lower the SCF energy (the values are the unpublished personal preference of MWS), and for SPLIT3 we might suggest 3.0,1.0,1/3.

With all this as background, we are ready to present the table of polarization exponents built into GAMESS.

Built in polarization exponents, chosen by POLAR= in the \$BASIS group. The values are for d functions unless otherwise indicated.

Please note that the names associated with each column are only generally descriptive. For example, the column marked "Pople" contains a value for Si with which John Pople would not agree, and the Ga-Kr values in this column are actually from the Huzinaga "green book". The exponents for K-Kr under "Dunning" are from Curtiss, et al., not Thom Dunning. And so on.

	POPLE	POPN311	DUNNING	HUZINAGA	HONDO7
H	1.1(p)	0.75(p)	1.0(p)	1.0(p)	1.0(p)
He	1.1(p)	0.75(p)	1.0(p)	1.0(p)	1.0(p)
Li	0.2	0.200		0.076(p)	
Be	0.4	0.255		0.164(p)	0.32
B	0.6	0.401	0.70	0.388	0.50
C	0.8	0.626	0.75	0.600	0.72
N	0.8	0.913	0.80	0.864	0.98
O	0.8	1.292	0.85	1.154	1.28
F	0.8	1.750	0.90	1.496	1.62
Ne	0.8	2.304	1.00	1.888	2.00
Na	0.175			0.061(p)	0.157
Mg	0.175			0.101(p)	0.234
Al	0.325			0.198	0.311
Si	0.395			0.262	0.388
P	0.55			0.340	0.465
S	0.65			0.421	0.542
Cl	0.75			0.514	0.619
Ar	0.85			0.617	0.696
K	0.2			0.260	0.039(p)
Ca	0.2			0.229	0.059(p)
Sc-Zn	0.8(f)	N/A	N/A	N/A	N/A
Ga	0.207		0.141		
Ge	0.246		0.202		
As	0.293		0.273		
Se	0.338		0.315		
Br	0.389		0.338		
Kr	0.443		0.318		
Rb	0.11			0.034(p)	
Sr	0.11			0.048(p)	

A blank means the value equals the "Pople" column.

Common d polarization for all sets ("green book"):

In	Sn	Sb	Te	I	Xe
0.160	0.183	0.211	0.237	0.266	0.297
Tl	Pb	Bi	Po	At	Rn
0.146	0.164	0.185	0.204	0.225	0.247

f polarization functions, from reference 63:

Li	Be	B	C	N	O	F	Ne
0.15	0.26	0.50	0.80	1.00	1.40	1.85	2.50
Na	Mg	Al	Si	P	S	Cl	Ar
0.15	0.20	0.25	0.32	0.45	0.55	0.70	--

## Anion diffuse functions

3-21+G, 3-21++G, etc.

- 70) T.Clark, J.Chandrasekhar, G.W.Spitznagel, P. von R. Schleyer J.Comput.Chem. 4, 294-301(1983 )  
 71) G.W.Spitznagel, Diplomarbeit, Erlangen, 1982 .

Anions usually require diffuse basis functions to properly represent their spatial diffuseness. The use of diffuse sp shells on atoms in the second and third rows is denoted by a + sign, also adding diffuse s functions on hydrogen is symbolized by ++. These designations can be applied to any of the Pople bases, e.g. 3-21+G, 3-21+G\*, 6-31++G\*\*. The following exponents are for L shells, except for H. For H-F, they are taken from ref 70. For Na-Cl, they are taken directly from reference 71. These values may be found in footnote 13 of reference 63. For Ga-Br, In-I, and Tl-At these were optimized for the atomic ground state anion, using ROHF with a flexible ECP basis set, by Ted Packwood at NDSU.

H						
0.0360						
Li	Be	B	C	N	O	F
0.0074	0.0207	0.0315	0.0438	0.0639	0.0845	0.1076
Na	Mg	Al	Si	P	S	Cl
0.0076	0.0146	0.0318	0.0331	0.0348	0.0405	0.0483
		Ga	Ge	As	Se	Br
		0.0205	0.0222	0.0287	0.0318	0.0376
		In	Sn	Sb	Te	I
		0.0223	0.0231	0.0259	0.0306	0.0368
		Tl	Pb	Bi	Po	At
		0.0170	0.0171	0.0215	0.0230	0.0294

Additional information about diffuse functions and also Rydberg type exponents can be found in reference 30.

The following atomic energies are from UHF calculations (RHF on 1-S states), with p orbitals not symmetry equivalenced, and using the default molecular scale factors. They should be useful in picking a basis of the desired energy accuracy, and estimating the correct molecular total energies.

Atom state	STO-2G	STO-3G	3-21G	6-31G
H 2-S	-.454397	-.466582	-.496199	-.498233
He 1-S	-2.702157	-2.807784	-2.835680	-2.855160
Li 2-S	-7.070809	-7.315526	-7.381513	-7.431236
Be 1-S	-13.890237	-14.351880	-14.486820	-14.566764
B 2-P	-23.395284	-24.148989	-24.389762	-24.519492
C 3-P	-36.060274	-37.198393	-37.481070	-37.677837
N 4-S	-53.093007	-53.719010	-54.105390	-54.385008
O 3-P	-71.572305	-73.804150	-74.393657	-74.780310
F 2-P	-95.015084	-97.986505	-98.845009	-99.360860
Ne 1-S	-122.360485	-126.132546	-127.803825	-128.473877
Na 2-S	-155.170019	-159.797148	-160.854065	-161.841425
Mg 1-S	-191.507082	-197.185978	-198.468103	-199.595219
Al 2-P	-233.199965	-239.026471	-240.551046	-241.854186
Si 3-P	-277.506857	-285.563052	-287.344431	-288.828598
P 4-S	-327.564244	-336.944863	-339.000079	-340.689008

S	3-P	-382.375012	-393.178951	-395.551336	-397.471414
Cl	2-P	-442.206260	-454.546015	-457.276552	-459.442939
Ar	1-S	-507.249273	-521.222881	-524.342962	-526.772151

Atom state	DH	6-311G	MC	SCF limit *
H 2-S	-.498189	-.499810		-0.5
He 1-S		-2.859895		-2.861680
Li 2-S	-7.431736	-7.432026		-7.432727
Be 1-S	-14.570907	-14.571874		-14.573023
B 2-P	-24.526601	-24.527020		-24.529061
C 3-P	-37.685571	-37.686024		-37.688619
N 4-S	-54.397260	-54.397980		-54.400935
O 3-P	-74.802707	-74.802496		-74.809400
F 2-P	-99.395013	-99.394158		-99.409353
Ne 1-S	-128.522354	-128.522553		-128.547104
Na 2-S			-161.845587	-161.858917
Mg 1-S			-199.606558	-199.614636
Al 2-P	-241.855079		-241.870014	-241.876699
Si 3-P	-288.829617		-288.847782	-288.854380
P 4-S	-340.689043		-340.711346	-340.718798
S 3-P	-397.468667		-397.498023	-397.504910
Cl 2-P	-459.435938		-459.473412	-459.482088
Ar 1-S			-526.806626	-526.817528

\* M.W.Schmidt and K.Ruedenberg, J.Chem.Phys. 71, 3951-3962(1979). These are ROHF energies in Kh symmetry.

## How to do RHF, ROHF, UHF, and GVB calculations

### \* \* \* General considerations \* \* \*

These four SCF wavefunctions are all based on Fock operator techniques, even though some GVB runs use more than one determinant. Thus all of these have an intrinsic  $N^4$  time dependence, because they are all driven by integrals in the AO basis. This similarity makes it convenient to discuss them all together. In this section we will use the term HF to refer generically to any of these four wavefunctions, including the multi-determinate GVB-PP functions. \$SCF is the main input group for all these HF wavefunctions.

As will be discussed below, in GAMESS the term ROHF refers to high spin open shell SCF only, but other open shell coupling cases are possible using the GVB code.

Analytic gradients are implemented for every possible HF type calculation possible in GAMESS, and therefore numerical Hessians are available for each.

Analytic Hessian calculation is implemented for RHF, ROHF, and any GVB case with NPAIR=0 or NPAIR=1. Analytic Hessians are more accurate, and much more quickly computed than numerical Hessians, but require additional disk storage to perform an integral transformation, and also more physical memory.

The second order Moller-Plesset energy correction (MP2) is implemented for RHF, UHF, ROHF, and MCSCF wave functions. Analytic gradients may be obtained for MP2 with a RHF reference function. MP2 properties are formed only by RHF gradient runs, or if you select MP2ROP in the \$MP2 group. All other cases give properties for the SCF function.

Direct SCF is implemented for every possible HF type calculation. The direct SCF method may not be used with DEM convergence. Direct SCF may be used during energy, gradient, numerical or analytic hessian, CI or MP2 energy correction, or localized orbitals computations.

\* \* \* direct SCF \* \* \*

Normally, HF calculations proceed by evaluating a large number of two electron repulsion integrals, and storing these on a disk. This integral file is read back in during each HF iteration to form the appropriate Fock operators. In a direct HF, the integrals are not stored on disk, but are instead reevaluated during each HF iteration. Since the direct approach *\*always\** requires more CPU time, the default for DIRSCF in \$SCF is .FALSE.

Even though direct SCF is slower, there are at least three reasons why you may want to consider using it. The first is that it may not be possible to store all of the integrals on the disk drives attached to your computer. Secondly, the index label packing scheme used by GAMESS restricts the basis set size to no more than 361 if the integrals are stored on disk, whereas for direct HF you can (in principle) use up to 2047 basis functions. Finally, what you are really interested in is reducing the wall clock time to obtain your answer, not the CPU time. Workstations have modest hardware (and sometimes software) I/O capabilities. Other environments such as an IBM mainframe shared by many users may also have very poor CPU/wall clock performance for I/O bound jobs such as conventional HF.

You can estimate the disk storage requirements for conventional HF using a P or PK file by the following formulae:

$$\begin{aligned} \text{nint} &= 1/\sigma * 1/8 * N^4 \\ \text{Mbytes} &= \text{nint} * x / 1024^2 \end{aligned}$$

Here N is the total number of basis functions in your run, which you can learn from an EXETYP=CHECK run. The 1/8 accounts for permutational symmetry within the integrals. Sigma accounts for the point group symmetry, and is difficult to estimate accurately. Sigma cannot be smaller than 1, in no symmetry (C1) calculations. For benzene, sigma would be almost six, since you generate 6 C's and 6 H's by entering only 1 of each in \$DATA. For water sigma is not much larger than one, since most of the basis set is on the unique oxygen, and the C<sub>2v</sub> symmetry applies only to the H atoms. The factor x is 12 bytes per integral for RHF, and 20 bytes per integral for ROHF, UHF, and GVB. Finally, since integrals very close to zero need not be stored on disk, the actual power dependence is not as bad as N<sup>4</sup>, and in fact in the limit of very large molecules can be as low as N<sup>2</sup>. Thus plugging in sigma=1 should give you an upper bound to the actual disk space needed. If the estimate exceeds your available disk storage, your only recourse is direct HF.

What are the economics of direct HF? Naively, if we assume the run takes 10 iterations to converge, we must spend 10 times more CPU time doing the integrals on each iteration. However, we do not have to waste any CPU time reading blocks of integrals from disk, or in unpacking their indices. We also do not have to waste any wall clock time waiting for a relatively slow mechanical device such as a disk to give us our data.

There are some less obvious savings too, as first noted by Almlöf. First, since the density matrix is known while we are computing integrals, we can use the Schwarz inequality to avoid doing some of the integrals. In a conventional SCF this inequality is used to avoid doing small integrals. In a direct SCF it can be used to avoid doing integrals whose contribution to the Fock matrix is small (density times integral=small). Secondly, we can form the Fock matrix by calculating only its change since the previous iteration. The contributions to the change in the

Fock matrix are equal to the change in the density times the integrals. Since the change in the density goes to zero as the run converges, we can use the Schwarz screening to avoid more and more integrals as the calculation progresses. The input option FDIFF in \$SCF selects formation of the Fock operator by computing only its change from iteration to iteration. The FDIFF option is not implemented for GVB since there are too many density matrices from the previous iteration to store, but is the default for direct RHF, ROHF, and UHF.

So, in our hypothetical 10 iteration case, we do not spend as much as 10 times more time in integral evaluation. Additionally, the run as a whole will not slow down by whatever factor the integral time is increased. A direct run spends no additional time summing integrals into the Fock operators, and no additional time in the Fock diagonalizations. So, generally speaking, a RHF run with 10-15 iterations will slow down by a factor of 2-4 times when run in direct mode. The energy gradient time is unchanged by direct HF, and this is a large time compared to HF energy, so geometry optimizations will be slowed down even less. This is really the converse of Amdahl's law: if you slow down only one portion of a program by a large amount, the entire program slows down by a much smaller factor.

To make this concrete, here are some times for GAMESS for a job which is a RHF energy for a SbC4O2NH4. These timings were obtained an extremely long time ago, on a DECstation 3100 under Ultrix 3.1, which was running only these tests, so that the wall clock times are meaningful. This system is typical of Unix workstations in that it uses SCSI disks, and the operating system is not terribly good at disk I/O. By default GAMESS stores the integrals on disk in the form of a P supermatrix, because this will save time later in the SCF cycles. By choosing NOPK=1 in \$INTGRL, an ordinary integral file can be used, which typically contains many fewer integrals, but takes more CPU time in the SCF. Because the DECstation is not terribly good at I/O, the wall clock time for the ordinary integral file is actually less than when the supermatrix is used, even though the CPU time is longer. The run takes 13 iterations to converge, the times are in seconds.

	P supermatrix	ordinary file
# nonzero integrals	8,244,129	6,125,653
# blocks skipped	55,841	55,841
CPU time (ints)	709	636
CPU time (SCF)	1289	1472
CPU time (job total)	2123	2233
wall time (job total)	3468	3200

When the same calculation is run in direct mode (integrals are processed like an ordinary integral disk file when running direct),

	FDIFF=.TRUE.	FDIFF=.FALSE.
iteration 1:		
# nonzero integrals	6,117,416	6,117,416
# blocks skipped	60,208	60,208
iteration 13:		
# nonzero integrals	3,709,733	6,122,912
# blocks skipped	105,278	59,415
CPU time (job total)	6719	7851
wall time (job total)	6764	7886

Note that elimination of the disk I/O dramatically increases the CPU/wall efficiency. Here's the bottom line on direct HF:

best direct CPU / best disk CPU = 6719/2123 = 3.2  
 best direct wall/ best disk wall= 6764/3200 = 2.1

Direct SCF is slower than conventional SCF, but not outrageously so! From the data in the tables, we can see that the best direct method spends about  $6719-1472 = 5247$  seconds doing integrals. This is an increase of about  $5247/636 = 8.2$  in the time spent doing integrals, in a run which does 13 iterations (13 times evaluating integrals). 8.2 is less than 13 because the run avoids all CPU charges related to I/O, and makes efficient use of the Schwarz inequality to avoid doing many of the integrals in its final iterations.

### \* \* \* SCF convergence accelerators \* \* \*

Generally speaking, the simpler the HF function, the better its convergence. In our experience, the majority of RHF, ROHF, and UHF runs will converge readily from GUESS=HUCKEL. GVB runs typically require GUESS=MOREAD, although the Huckel guess usually works for NPAIR=0. RHF convergence is the best, closely followed by ROHF. In the current implementation in GAMESS, ROHF is always better convergent than the closely related unrestricted high spin UHF. For example, the radical cation of diphosphine converges in 12 iterations for ROHF and 15 iterations for UHF, starting from the neutral's closed shell orbitals. GVB calculations require much more care, and cases with NPAIR greater than one are particularly difficult.

Unfortunately, not all HF runs converge readily. The best way to improve your convergence is to provide better starting orbitals! In many cases, this means to MOREAD orbitals from some simpler HF case. For example, if you want to do a doublet ROHF, and the HUCKEL guess does not seem to converge, do this: Do an RHF on the +1 cation. RHF is typically more stable than ROHF, UHF, or GVB, and cations are usually readily convergent. Then MOREAD the cation's orbitals into the neutral calculation which you wanted to do at first.

GUESS=HUCKEL does not always guess the correct electronic configuration. It may be useful to use PRTMO in \$GUESS during a CHECK run to examine the starting orbitals, and then reorder them with NORDER if that seems appropriate.

Of course, by default GAMESS uses the convergence procedures which are usually most effective. Still, there are cases which are difficult, so the \$SCF group permits you to select several alternative methods for improving convergence. Briefly, these are

- EXTRAP. This extrapolates the three previous Fock matrices, in an attempt to jump ahead a bit faster. This is the most powerful of the old-fashioned accelerators, and normally should be used at the beginning of any SCF run. When an extrapolation occurs, the counter at the left of the SCF printout is set to zero.
- DAMP. This damps the oscillations between several successive Fock matrices. It may help when the energy is seen to oscillate wildly. Thinking about which orbitals should be occupied initially may be an even better way to avoid oscillatory behavior.
- SHIFT. This shifts the diagonal elements of the virtual part of the Fock matrix up, in an attempt to uncouple the unoccupied orbitals from the occupied ones. At convergence, this has no effect on the orbitals, just their orbital energies, but will produce different (and hopefully better) orbitals during the iterations.
- RSTRCT. This limits mixing of the occupied orbitals with the empty ones, especially the flipping of the HOMO and LUMO to produce undesired electronic configurations or

states. This should be used with caution, as it makes it very easy to converge on incorrect electronic configurations, especially if DIIS is also used. If you use this, be sure to check your final orbital energies to see if they are sensible. A lower energy for an unoccupied orbital than for one of the occupied ones is a sure sign of problems.

- DIIS. Direct Inversion in the Iterative Subspace is a modern method, due to Pulay, using stored error and Fock matrices from a large number of previous iterations to interpolate an improved Fock matrix. This method was developed to improve the convergence at the final stages of the SCF process, but turns out to be quite powerful at forcing convergence in the initial stages of SCF as well. By giving ETHRSH as 10.0 in \$SCF, you can practically guarantee that DIIS will be in effect from the first iteration. The default is set up to do a few iterations with conventional methods (extrapolation) before engaging DIIS. This is because DIIS can sometimes converge to solutions of the SCF equations that do not have the lowest possible energy. For example, the 3-A-2 small angle state of SiLi<sub>2</sub> (see M.S.Gordon and M.W.Schmidt, Chem.Phys.Lett., 132, 294-8(1986 )) will readily converge with DIIS to a solution with a reasonable S<sup>2</sup>, and an energy about 25 milliHartree above the correct answer. A SURE SIGN OF TROUBLE WITH DIIS IS WHEN THE ENERGY RISES TO ITS FINAL VALUE. However, if you obtain orbitals at one point on a PES without DIIS, the subsequent use of DIIS with MOREAD will probably not introduce any problems. Because DIIS is quite powerful, EXTRAP, DAMP, and SHIFT are all turned off once DIIS begins to work. DEM and RSTRCT will still be in use, however.
- SOSCF. Approximate second-order (quasi-Newton) SCF orbital optimization. SOSCF will converge about as well as DIIS at the initial geometry, and slightly better at subsequent geometries. There's a bit less work solving the SCF equations, too. The method kicks in after the orbital gradient falls below SOGTOL. Some systems, particularly transition metals with ECP basis sets, may have Huckel orbitals for which the gradient is much larger than SOGTOL. In this case it is probably better to use DIIS instead, with a large ETHRSH, rather than increasing SOGTOL, since you may well be outside the quadratic convergence region. SOSCF does not exhibit true second order convergence since it uses an approximation to the inverse hessian. SOSCF will work for MOPAC runs, but is slower in this case. SOSCF will work for UHF, but the convergence is slower than DIIS. SOSCF will work for non-Abelian ROHF cases, but may encounter problems if the open shell is degenerate.
- DEM. Direct energy minimization should be your last recourse. It explores the "line" between the current orbitals and those generated by a conventional change in the orbitals, looking for the minimum energy on that line. DEM should always lower the energy on every iteration, but is very time consuming, since each of the points considered on the line search requires evaluation of a Fock operator. DEM will be skipped once the density change falls below DEMCUT, as the other methods should then be able to affect final convergence. While DEM is working, RSTRCT is held to be true, regardless of the input choice for RSTRCT. Because of this, it behooves you to be sure that the initial guess is occupying the desired orbitals. DEM is available only for RHF. The implementation in GAMESS resembles that of R.Seeger and J.A.Pople, J.Chem.Phys. 65, 265-271(1976 ). Simultaneous use of DEM and DIIS resembles the ADEM-DIOS method of H.Sellers, Chem.Phys.Lett. 180, 461-465(1991 ). DEM does not work with direct SCF.



## \* \* \* High spin open shell SCF (ROHF) \* \* \*

Open shell SCF calculations are performed in GAMESS by both the ROHF code and the GVB code. Note that when the GVB code is executed with no pairs, the run is NOT a true GVB run, and should be referred to in publications and discussion as a ROHF calculation.

The ROHF module in GAMESS can handle any number of open shell electrons, provided these have a high spin coupling. Some commonly occurring cases are:

one open shell, doublet:

```
$CONTRL SCFTYP=ROHF MULT=2 $END
```

two open shells, triplet:

```
$CONTRL SCFTYP=ROHF MULT=3 $END
```

m open shells, high spin:

```
$CONTRL SCFTYP=ROHF MULT=m+1 $END
```

John Montgomery (then at United Technologies) is responsible for the current ROHF implementation in GAMESS. The following discussion is due to him:

The Fock matrix in the MO basis has the form

	closed	open	virtual
closed	F2	Fb	(Fa+Fb)/2
open	Fb	F1	Fa
virtual	(Fa+Fb)/2	Fa	F0

where Fa and Fb are the usual alpha and beta Fock matrices any UHF program produces. The Fock operators for the doubly, singly, and zero occupied blocks can be written as

$$F2 = Acc*Fa + Bcc*Fb$$

$$F1 = Aoo*Fa + Boo*Fb$$

$$F0 = Avv*Fa + Bvv*Fb$$

Some choices found in the literature for these canonicalization coefficients are

	Acc	Bcc	Aoo	Boo	Avv	Bvv
Guest and Saunders	1/2	1/2	1/2	1/2	1/2	1/2
Roothaan single matrix	-1/2	3/2	1/2	1/2	3/2	-1/2
Davidson	1/2	1/2	1	0	1	0
Binkley, Pople, Dobosh	1/2	1/2	1	0	0	1
McWeeny and Diercksen	1/3	2/3	1/3	1/3	2/3	1/3
Faegri and Manne	1/2	1/2	1	0	1/2	1/2

The choice of the diagonal blocks is arbitrary, as ROHF is converged when the off diagonal blocks go to zero. The exact choice for these blocks can however have an effect on the convergence rate. This choice also affects the MO coefficients, and orbital energies, as the different choices produce different canonical orbitals within the three subspaces. All methods, however, will give identical total wavefunctions, and hence identical properties such as gradients and Hessians.

The default coupling case in GAMESS is the Roothaan single matrix set. Note that pre-1988 versions of GAMESS produced "Davidson" orbitals. If you would like to fool around with any of these other canonicalizations, the Acc, Aoo, Avv and Bcc, Boo, Bvv parameters can be input as the first three elements of ALPHA and BETA in \$SCF.

## \* \* \* Other open shell SCF cases (GVB) \* \* \*

Genuine GVB-PP runs will be discussed later in this section. First, we will consider how to do open shell SCF with the GVB part of the program.

It is possible to do other open shell cases with the GVB code, which can handle the following cases:

one open shell, doublet:

```
$CONTRL SCFTYP=GVB MULT=2 $END
$SCF    NCO=xx NSETO=1 NO(1)=1 $END
```

two open shells, triplet:

```
$CONTRL SCFTYP=GVB MULT=3 $END
$SCF    NCO=xx NSETO=2 NO(1)=1, 1 $END
```

two open shells, singlet:

```
$CONTRL SCFTYP=GVB MULT=1 $END
$SCF    NCO=xx NSETO=2 NO(1)=1, 1 $END
```

Note that the first two cases duplicate runs which the ROHF module can do better. Note that all of these cases are really ROHF, since the default for NPAIR in \$SCF is 0.

Many open shell states with degenerate open shells (for example, in diatomic molecules) can be treated as well. There is a sample of this in the 'Input Examples' section of this manual.

If you would like to do any cases other than those shown above, you must derive the coupling coefficients ALPHA and BETA, and input them with the occupancies F in the \$SCF group.

Mariusz Klobukowski of the University of Alberta has shown how to obtain coupling coefficients for the GVB open shell program for many such open shell states. These can be derived from the values in Appendix A of the book "A General SCF Theory" by Ramon Carbo and Josep M. Riera, Springer-Verlag (1978). The basic rule is

- (1)  $F(i) = 1/2 * \omega(i)$
- (2)  $ALPHA(i) = \alpha(i)$
- (3)  $BETA(i) = -\beta(i)$ ,

where  $\omega$ ,  $\alpha$ , and  $\beta$  are the names used by Ramon in his Tables.

The variable NSETO should give the number of open shells, and NO should give the degeneracy of each open shell. Thus the 5-S state of carbon would have NSETO=2, and NO(1)=1,3.

Some specific examples, for the lowest term in each of the atomic  $P^N$  configurations are

```
! p**1 2-P state
$CONTRL SCFTYP=GVB MULT=2 $END
$SCF    NCO=xx NSETO=1 NO=3 COUPLE=. TRUE.
      F(1)= 1.0 0.166666666666667
      ALPHA(1)= 2.0 0.333333333333333 0.000000000000000
      BETA(1)= -1.0 -0.166666666666667 -0.000000000000000 $END
! p**2 3-P state
$CONTRL SCFTYP=GVB MULT=3 $END
$SCF    NCO=xx NSETO=1 NO=3 COUPLE=. TRUE.
      F(1)= 1.0 0.333333333333333
      ALPHA(1)= 2.0 0.666666666666667 0.166666666666667
```

```

      BETA(1)= -1.0 -0.3333333333333333 -0.166666666666667 $SEND
!   p**3   4-S state
$CONTRL SCFTYP=ROHF MULT=4 $SEND
!   p**4   3-P state
$CONTRL SCFTYP=GVB MULT=3 $SEND
$SCF   NCO=xx NSETO=1 NO=3 COUPLE=. TRUE.
      F(1)= 1.0 0.666666666666667
      ALPHA(1)= 2.0 1.333333333333333 0.833333333333333
      BETA(1)= -1.0 -0.666666666666667 -0.500000000000000 $SEND
!   p**5   2-P state
$CONTRL SCFTYP=GVB MULT=2 $SEND
$SCF   NCO=xx NSETO=1 NO=3 COUPLE=. TRUE.
      F(1)= 1.0 0.833333333333333
      ALPHA(1)= 2.0 1.666666666666667 1.333333333333333
      BETA(1)= -1.0 -0.833333333333333 -0.666666666666667 $SEND

```

Be sure to give all the digits, as these are part of a double precision energy formula.  
Coupling constants for d<sup>N</sup> configurations are

```

!   d**1   2-D state
$CONTRL SCFTYP=GVB MULT=2 $SEND
$SCF   NCO=xx NSETO=1 NO=5 COUPLE=. TRUE. F(1)=1.0, 0.1
      ALPHA(1)= 2.0, 0.20, 0.00
      BETA(1)=-1.0, -0.10, 0.00 $SEND
!   d**2   average of 3-F and 3-P states
$CONTRL SCFTYP=GVB MULT=3 $SEND
$SCF   NCO=xx NSETO=1 NO=5 COUPLE=. TRUE. F(1)=1.0, 0.2
      ALPHA(1)= 2.0, 0.40, 0.05
      BETA(1)=-1.0, -0.20, -0.05 $SEND
!   d**3   average of 4-F and 4-P states
$CONTRL SCFTYP=GVB MULT=4 $SEND
$SCF   NCO=xx NSETO=1 NO=5 COUPLE=. TRUE. F(1)=1.0, 0.3
      ALPHA(1)= 2.0, 0.60, 0.15
      BETA(1)=-1.0, -0.30, -0.15 $SEND
!   d**4   5-D state
$CONTRL SCFTYP=GVB MULT=5 $SEND
$SCF   NCO=xx NSETO=1 NO=5 COUPLE=. TRUE. F(1)=1.0, 0.4
      ALPHA(1)= 2.0, 0.80, 0.30
      BETA(1)=-1.0, -0.40, -0.30 $SEND
!   d**5   6-S state
$CONTRL SCFTYP=ROHF MULT=6 $SEND
!   d**6   5-D state
$CONTRL SCFTYP=GVB MULT=5 $SEND
$SCF   NCO=xx NSETO=1 NO=5 COUPLE=. TRUE. F(1)=1.0, 0.6
      ALPHA(1)= 2.0, 1.20, 0.70
      BETA(1)=-1.0, -0.60, -0.50 $SEND
!   d**7   average of 4-F and 4-P states
$CONTRL SCFTYP=GVB MULT=4 $SEND
$SCF   NCO=xx NSETO=1 NO=5 COUPLE=. TRUE. F(1)=1.0, 0.7
      ALPHA(1)= 2.0, 1.40, 0.95
      BETA(1)=-1.0, -0.70, -0.55 $SEND
!   d**8   average of 3-F and 3-P states
$CONTRL SCFTYP=GVB MULT=3 $SEND
$SCF   NCO=xx NSETO=1 NO=5 COUPLE=. TRUE. F(1)=1.0, 0.8
      ALPHA(1)= 2.0, 1.60, 1.25
      beta(1)=-1.0, -0.80, -0.65 $SEND

```

```

!      d**9    2-D state
$CONTRL SCFTYP=GVB MULT=2 $SEND
$SCF     NCO=xx NSETO=1 NO=5 COUPLE=. TRUE.   F(1)=1. 0. 0. 9
          ALPHA(1)= 2. 0, 1. 80, 1. 60
          BETA(1)=- 1. 0, -0. 90, -0. 80 $SEND

```

The source for these values is R.Poirier, R.Kari, and I.G.Csizmadia's book Handbook of Gaussian Basis Sets, Elsevier, Amsterdam, 1985.

Note that GAMESS can do a proper calculation on the ground terms for the  $d^2$ ,  $d^3$ ,  $d^7$ , and  $d^8$  configurations only by means of state averaged MCSCF. For  $d^8$ , use

```

$CONTRL SCFTYP=MCSCF MULT=3 $SEND
$SDRT   GROUP=C1 FORS=. TRUE.  NMCC=xx NDOC=3 NALP=2 $SEND
$GUGDIA NSTATE=10 $SEND
$GUGDM2 WSTATE(1)=1, 1, 1, 1, 1, 1, 1, 0, 0, 0 $SEND

```

Open shell cases such as  $s^{**1}, d^{**n}$  are probably most easily tackled with the state-averaged MCSCF program.

### \* \* \* True GVB perfect pairing runs \* \* \*

True GVB runs are obtained by choosing NPAIR nonzero. If you wish to have some open shell electrons in addition to the geminal pairs, you may add the pairs to the end of any of the GVB coupling cases shown above. The GVB module assumes that you have reordered your MOs into the order: NCO double occupied orbitals, NSETO sets of open shell orbitals, and NPAIR sets of geminals (with NORDER=1 in the \$GUESS group).

Each geminal consists of two orbitals and contains two singlet coupled electrons (perfect pairing). The first MO of a geminal is probably heavily occupied (such as a bonding MO  $u$ ), and the second is probably weakly occupied (such as an antibonding, correlating orbital  $v$ ). If you have more than one pair, you must be careful that the initial MOs are ordered  $u_1, v_1, u_2, v_2, \dots$ , which is -NOT- the same order that RHF starting orbitals will be found in. Use NORDER=1 to get the correct order.

These pair wavefunctions are actually a limited form of MCSCF. GVB runs are much faster than MCSCF runs, because the natural orbital  $u,v$  form of the wavefunction permits a Fock operator based optimization. However, convergence of the GVB run is by no means assured. The same care in selecting the correlating orbitals that you would apply to an MCSCF run must also be used for GVB runs. In particular, look at the orbital expansions when choosing the starting orbitals, and check them again after the run converges.

GVB runs will be carried out entirely in orthonormal natural  $u,v$  form, with strong orthogonality enforced on the geminals. Orthogonal orbitals will pervade your thinking in both initial orbital selection, and the entire orbital optimization phase (the CICOEF values give the weights of the  $u,v$  orbitals in each geminal). However, once the calculation is converged, the program will generate and print the nonorthogonal, generalized valence bond orbitals. These GVB orbitals are an entirely equivalent way of presenting the wavefunction, but are generated only after the fact.

Convergence of true GVB runs is by no means as certain as convergence of RHF, UHF, ROHF, or GVB with NPAIR=0. You can assist convergence by doing a preliminary RHF or ROHF calculation, and use these orbitals for GUESS=MOREAD. Few, if any, GVB runs with NPAIR nonzero will converge without using GUESS=MOREAD. Generation of MVOs during the preliminary

SCF can also be advantageous. In fact, all the advice outlined for MCSCF computations below is germane, for GVB-PP is a type of MCSCF computation.

The total number of electrons in the GVB wavefunction is given by the following formula:

$$NE = 2*NCO + \sum 2*F(i)*NO(i) + 2*NPAIR$$

The charge is obtained by subtracting the total number of protons given in \$DATA. The multiplicity is implicit in the choice of alpha and beta constants. Note that ICHARG and MULT must be given correctly in \$CONTRL anyway, as the number of electrons from this formula is double checked against the ICHARG value.

\* \* \* the special case of TCSCF \* \* \*

The wavefunction with NSETO=0 and NPAIR=1 is called GVB-PP(1) by Goddard, two configuration SCF (TCSCF) by Schaefer or Davidson, and CAS-SCF with two electrons in two orbitals by others. Note that this is just semantics, as these are all identical. This is a very important type of wavefunction, as TCSCF is the minimum acceptable treatment for singlet biradicals. The TCSCF wavefunction can be obtained with SCFTYP=MCSCF, but it is usually much faster to use the Fock based SCFTYP=GVB. Because of its importance, the TCSCF function (if desired, with possible open shells) permits analytic hessian computation.

\* \* \* A caution about symmetry \* \* \*

Caution! Some exotic calculations with the GVB program do not permit the use of symmetry. The symmetry algorithm in GAMESS was "derived assuming that the electronic charge density transforms according to the completely symmetric representation of the point group", Dupuis/King, JCP, 68, 3998(1978). This may not be true for certain open shell cases, and in fact during GVB runs, it may not be true for closed shell singlet cases!

First, consider the following correct input for the singlet-delta state of NH:

```
$CONTRL SCFTYP=GVB NOSYM=1 $END
$SCF     NCO=3 NSETO=2 NO(1)=1, 1 $END
```

for the  $x^1y^1$  state, or for the  $x^2-y^2$  state,

```
$CONTRL SCFTYP=GVB NOSYM=1 $END
$SCF     NCO=3 NPAIR=1 CI COEF(1)=0.707, -0.707 $END
```

Neither gives correct results, unless you enter NOSYM=1. The electronic term symbol is degenerate, a good tip off that symmetry cannot be used. However, some degenerate states can still use symmetry, because they use coupling constants averaged over all degenerate states within a single term, as is done in EXAM15 and EXAM16. Here the "state averaged SCF" leads to a charge density which is symmetric, and these runs can exploit symmetry.

Secondly, since GVB runs exploit symmetry for each of the "shells", or type of orbitals, some calculations on totally symmetric states may not be able to use symmetry. An example is CO or N<sub>2</sub>, using a three pair GVB to treat the sigma and pi bonds. Individual configurations such as  $(\sigma)^{**2}, (\pi-x)^{**2}, (\pi-y)^{**2}$  do not have symmetric charge densities since neither the pi nor pi\* level is completely filled. Correct answers for the sigma-plus ground states result only if you input NOSYM=1.

Problems of the type mentioned should not arise if the point group is Abelian, but will be fairly common in linear molecules. Since GAMESS cannot detect that the GVB electronic state is not totally symmetric (or averaged to at least have a totally symmetric density), it is left up to you to decide when to input NOSYM=1. If you have any question about the use of symmetry, try it both ways. If you get the same energy, both ways, it remains valid to use symmetry to speed up your run.

And beware! Brain dead computations, such as RHF on singlet O<sub>2</sub>, which actually is a half filled degenerate shell, violate the symmetry assumptions, and also violate nature. Use of partially filled degenerate shells always leads to very wild oscillations in the RHF energy, which is how the program tries to tell you to think first, and compute second. Configurations such as pi\*\*2, e\*\*1, or f<sub>2u</sub>\*\*4 can be treated, but require GVB wavefunctions and F, ALPHA, BETA values from the sources mentioned.

### How to do MCSCF and CI calculations

Multi-configuration self consistent field (MCSCF) wavefunctions are the most general SCF type, offering a description of chemical processes involving the separation of electrons (bond breaking, electronically excited states, etc), which are often not well represented using the single configuration SCF methods.

MCSCF wavefunctions, as the name implies, contain more than one configuration, each of which is multiplied by a "configuration interaction (CI) coefficient", determining its weight. In addition, the orbitals which form each of the configurations are optimized, just as in a simpler SCF, to self consistency.

Typically each chemical problem requires that an MCSCF wavefunction be designed to treat it, on a case by case basis. For example, one may be interested in describing the reactivity of a particular functional group, instead of elsewhere in the molecule. This means some attention must be paid in order to obtain correct results.

Procedures for the selection of configurations (which amounts to choosing the number of active electrons and active orbitals), for the two mathematical optimizations just mentioned, ways to interpret the resulting MCSCF wavefunction, and the treatment for dynamical correlation not included in the MCSCF wavefunction are the focus of a recent review article:

"The Construction and Interpretation of MCSCF wavefunctions"

M.W.Schmidt and M.S.Gordon, *Ann.Rev.Phys.Chem.* 49,233-266(1998)

One section of this is devoted to the problem of designing the correct active space to treat your problem. Additional reading is listed at the end of this section.

The most efficient technique implemented in GAMESS for finding the dynamic correlation energy is second order perturbation theory, in the variant known as MCQDPT. MCQDPT is discussed in a different section of this chapter. The use of CI, probably in the form of second order CI, will be described below, en passant, during discussion of the input defining the configurations for MCSCF. Selection of a CI following some type of SCF (except UHF) is made with CITYP in the \$CONTRL group, and masterminded by the \$CIINP group.

#### --- MCSCF implementation ---

With the exception of the QUAD converger, the MCSCF program is of the type termed "unfolded two step" by Roos. This means the orbital and CI coefficient optimizations are separated. The latter are obtained in a conventional CI diagonalization, while the former are optimized by a separate orbital improvement step.

Each MCSCF iteration consists of the following steps:

- 1) transformation of AO integrals to the current MO basis,
- 2) generation of the Hamiltonian matrix and optimization of the CI coefficients by a Davidson diagonalization,

- 3) generation of the first and second order density matrix,
- 4) improvement of the molecular orbitals.

The CI problem in steps two and three has two options, namely a determinant or configuration state function (CSF) many electron basis set. The choice of these is determined by CISTEP in \$MCSCF. More will be said just below about the differences between determinants and CSFs. The word "configuration" is used in this section to refer to either when a generic term is needed for the many-electron basis, so please note there is a distinction between this and the very similar term CSF.

The orbital problem in step four has three options, namely FOCAS, SOSCF, and FULLNR, listed here in order of their increasing mathematical sophistication, convergence characteristics, and of course, their computer resource requirements. Again, these are chosen by keywords in the \$MCSCF group. More will be said just below about the relative merits of these three.

Finally, we mention again the QUAD converger, which works only for a CSF basis, in which the two optimization problems are treated simultaneously, for modest numbers of configurations (50-100 is probably the limit). In principle, this is the most robust method available, but in practice, it has not received very much use compared to the unfolded methods.

Depending on the converger chosen, the program will select the appropriate kind of integral transformation. There's seldom need to try to fine tune this, but note that the \$TRANS group does let you pick AO integral direct transformation with the DIRTRF flag.

On the first iteration at the first geometry, you will receive the normal amount of output from each of these stages, while each subsequent iterations will produce only a single summarizing line.

--- orbital updates ---

There are presently four orbital improvement options, namely FOCAS, SOSCF, FULLNR, and QUAD. All MCSCF orbital optimizations run in parallel. The four convergers are discussed briefly in the following paragraphs, in order of increasing robustness.

FOCAS is a first order, complete active space MCSCF optimization procedure. The FOCAS code was written by Michel Dupuis and Antonio Marquez at IBM. It is based on a novel approach due to Meier and Staemmler, using very fast but numerous microiterations to improve the convergence of what is intrinsically a first order method. Since FOCAS requires only one virtual orbital index in the integral transformation to compute the orbital gradient (aka the Lagrangian), the total MCSCF job may take less time than a second order method, even though it may require twice as many iterations to converge. The use of microiterations is crucial to FOCAS' ability to converge. It is important to take a great deal of care choosing the starting orbitals.

SOSCF is a method built upon the FOCAS code, which seeks to combine the speed of FOCAS with second order convergence properties. Thus SOSCF is an approximate Newton-Raphson, based on a diagonal guess at the orbital hessian, and in fact has much in common with the SOSCF option in \$SCF. Its time requirements per iteration are like FOCAS, with a convergence rate better than FOCAS but not as good as true second order. Storage of only the diagonal of the orbital hessian allows the SOSCF method to be used with much larger basis sets than exact second order methods. Because it usually requires the least CPU time, disk space, and memory needs, SOSCF is the default. Good convergence by the SOSCF method requires that you prepare starting orbitals carefully, and read in all MOs in \$VEC, as the provision of canonicalized

virtual orbitals increases the diagonal dominance of the orbital hessian.

FULLNR means a full Newton-Raphson orbital improvement step is taken, using the exact orbital hessian. FULLNR is a quite powerful convergence method, and normally takes the fewest iterations to converge. Computing the exact orbital hessian requires two virtual orbital indices be included in the transformation, making this step quite time consuming, and of course memory for storage of the orbital hessian must be available. Because both the transformation and orbital improvement steps of FULLNR are time consuming, FULLNR is not the default. You may want to try FULLNR when convergence is difficult, assuming you have already tried preparing good starting orbitals by the hints below.

The FULLNR MCSCF code in GAMESS is adapted from the HONDO7 program, and was written by Michel Dupuis at IBM. It uses the the augmented hessian matrix approach to solve the Newton-Raphson equations. There are two suboptions for computation of the orbital hessian. DM2 is the fastest but takes more memory than TEI.

QUAD uses a fully quadratic, or second order approach and is thus the most powerful MCSCF converger. The QUAD code is also adapted from Michel Dupuis's HONDO. QUAD runs begin with unfolded FULLNR iterations, until the orbitals approach convergence sufficiently. QUAD then begins the simultaneous optimization of CI coefficients and orbitals, and convergence should be obtained in 3-4 additional MCSCF iterations. The QUAD method requires building the full hessian, including orbital/orbital, orbital/CI, and CI/CI blocks, which is a rather big matrix. QUAD may be helpful in converging excited electronic states, but note that you may not use state averaging with QUAD. QUAD is a memory hog, and so may be used only for fairly small numbers of configurations.

The input to control the orbital update step is the \$MCSCF group, where you should pick one of the convergence procedures. Most of the input in this group is rather specialized, but note in particular MAXIT and ACURCY which control the convergence behaviour. In some circumstances the diagonalizations of the core and virtual orbitals to canonicalize these (after overall MCSCF convergence) may produce spatial orbital symmetry loss, particularly in point groups with degeneracy present, and then CANONC might be used to turn this canonicalization off.

--- CI coefficient optimization ---

Determinants or configuration state functions (CSFs) may be used to form the many electron basis set. It is necessary to explain these in a bit of detail so that you can understand the advantages of each.

A determinant is a simple object: a product of spin orbitals with a given  $S_z$  quantum number, that is, the number of alpha spins and number of beta spins are a constant. Matrix elements involving determinants are correspondingly simple, but unfortunately determinants are not necessarily eigenfunctions of the  $S^2$  operator.

To expand on this point, consider the four familiar 2e- functions which satisfy the Pauli principle. Here  $u, v$  are space orbitals, and  $a, b$  are the alpha and beta spin functions. As you know, the singlet and triplets are:

$$S1 = (uv + vu)/\sqrt{2} * (ab - ba)/\sqrt{2}$$

$$T1 = (uv) * aa$$

$$T2 = (uv - vu)/\sqrt{2} * (ab + ba)/\sqrt{2}$$

$$T3 = (uv) * bb$$

It is a simple matter to multiply out S1 and T2, and to expand the two determinants which have  $S_z=0$ ,

$$D1 = |ua vb| \quad D2 = |va ub|$$



This reveals that

$$S1 = (D1+D2)/\sqrt{2} \quad \text{or} \quad D1 = (S1 + T2)/\sqrt{2}$$

$$T2 = (D1-D2)/\sqrt{2} \quad D2 = (S1 - T2)/\sqrt{2}$$

Thus, one must take a linear combination of determinants in order to have a wavefunction with the desired total spin. There are two important points to note:

- A two by two Hamiltonian matrix over D1 and D2 has eigenfunctions with -different- spins,  $S=0$  and  $S=1$ .
- use of all determinants with  $S_z=0$  does allow for the construction of spin adapted states.  $D1+D2$ , or  $D1-D2$ , are -not- spin contaminated.

By itself, a determinant such as D1 is said to be "spin contaminated", being a fifty-fifty admixture of singlet and triplet (it is curious that calculations with just one such determinant are often called "singlet UHF"). Of course, some determinants are spin adapted all by themselves, for example the spin adapted functions T1 and T3 above are single determinants, as are the closed shells

$$S2 = (uu) * (ab - ba)/\sqrt{2}.$$

$$S3 = (vv) * (ab - ba)/\sqrt{2}.$$

It is possible to perform a triplet calculation, with no singlet states present, by choosing determinants with  $S_z=1$  such as T1, since then no state with  $S_z=0$  as is required when  $S=0$  exists in the determinant basis set. To summarize, the eigenfunctions of a Hamiltonian formed by determinants with any particular  $S_z$  will be spin states with  $S=S_z$ ,  $S=S_z+1$ ,  $S=S_z+2$ , ... but will not contain any  $S$  values smaller than  $S_z$ .

CSFs are an antisymmetrized combination of a space orbital product, and a spin adapted linear combination of simple alpha-beta products. Namely, the following CSF

$$C1 = A (uv) * (ab-ba)/\sqrt{2}$$

which has a singlet spin function is identical to S1 above if you write out what the antisymmetrizer A does, and the CSFs

$$C2 = A (uv) * aa$$

$$C3 = A (uv - vu)/\sqrt{2} * (ab + ba)/\sqrt{2}$$

$$C4 = A (uv) * bb$$

equal T1-T3. Since the three triplet CSFs have the same energy, GAMESS works with the simpler form C2. Singlet and triplet computations using CSFs are done in separate runs, because when spin-orbit coupling is not considered, the Hamiltonian is block diagonal in a CSF basis.

Technical information about the CSFs are that they use Yamanouchi-Kotani spin couplings, and matrix elements are obtained using a GUGA, or graphical unitary group approach.

The determinant implementation in GAMESS is written at Ames Laboratory, and can perform only full CI computations, meaning its primary use is for MCSCF wavefunctions of the full active space type. The CSF code is capable of more general CI computations, and so can be used for first or second order CI computations. Other comparisons between the determinant and CSF implementations, as they exist in GAMESS today, are

	determinants	CSFs
parallel execution	no	yes
direct CI	yes	no
exploits space symmetry	no	yes
state average mixed spins	yes	no
first order density	yes	yes
state averaged densities	yes	yes
can form CI Lagrangian	no	yes

In cases where there is high spatial symmetry, this and the restriction to spin-adapted functions, makes the CSF code the fastest. However, the determinant code generates matrix elements very quickly, cases with less spatial symmetry run faster with determinants, and

because it is a direct implementation, determinants do not use disk space to store Hamiltonian elements. Finally we note the initial guess of the CI eigenvector in the determinant code is much better than in the CSF code, where sometimes convergence to excited roots instead of lower ones occurs. On balance, the default CISTEP in \$MCSCF selects the ALDET, the Ames Laboratory determinant CI code.

The next two sections describe in detail the input for specification of the configurations, either determinants or CSFs.

#### --- determinant CI ---

The determinant CI code is capable only of full CI wavefunctions. As the implementation cannot exploit the spatial symmetry that might be present, the \$DET group becomes very simple. Many runs can be done by specifying only the orbital and electron counts: NCORE, NACT, and NELS. The number of electrons is  $2*NCORE+NELS$ , and will be checked against the charge implied by ICHARG. The MULT given in \$CONTRL is used to determine the desired Sz value, by extracting S from  $MULT=2S+1$ , then by default  $Sz=S$ . If you wish to include lower spin multiplicities, which will increase the CPU time of the run, but will let you know what the energies of such states are, just input a smaller value for SZ. The states whose orbitals will be MCSCF optimized will be those having the requested MULT value, unless you choose otherwise with the PURES flag.

The remaining parameters in the \$DET group give extra control over the diagonalization process. Most are not given in normal circumstances, except NSTATE, which you may need to adjust to produce enough roots of the desired MULT value. The only important keyword which has not been discussed is the WSTATE array, giving the weights for each state in forming the first and second order density matrix elements, which drive the orbital update methods. Note that analytic gradients are available only when the WSTATE array is a unit vector, corresponding to a pure state, such as  $WSTATE(1)=0,1,0$  which permits gradients of the first excited state to be computed. When used for state averaged MCSCF, WSTATE is normalized to a unit sum, thus input of  $WSTATE(1)=1,1,1$  really means a weight of 0.33333... for the each of the states being averaged.

If you are performing a CITYP=ALDET calculation, you give the input defining this full CI in a \$CIDET group, containing the same keywords just mentioned.

#### --- CSF CI ---

The GUGA-based CSF package was originally a set of different programs, so the input to control it is spread over several input groups. The CSFs are specified by a \$CIDRT group in the case of CITYP=GUGA, and by a \$DRT group for MCSCF wavefunctions. Thus it is possible to perform an MCSCF defined by a \$DRT input (or perhaps using \$DET during the MCSCF), and follow this with a second order CI defined by a \$CIDRT group, in the same run.

The remaining input groups used by the GUGA CSFs are \$CISORT, \$GUGEM, \$GUGDIA, and \$GUGDM2 for MCSCF runs, with the latter two being the most important, and in the case of CI computations, \$GUGDM and possibly \$LAGRAN groups are relevant. Perhaps the most interesting variables outside the \$DRT/\$CIDRT group are NSTATE in \$GUGDIA to include excited states in the CI computation, IROOT in \$GUGDM to select the state for properties, and WSTATE in \$GUGDM2 to control which (average) state's orbitals are optimized.

The \$DRT and \$CIDRT groups are almost the same, with the only difference being orbitals restricted to double occupancy are called MCC in \$DRT, and FZC in \$CIDET. Therefore the rest of this section refers only to "\$DRT".

The CSFs are specified by giving a reference CSF, together with a maximum degree of electron excitation from that single CSF. The MOs in the reference CSF are filled in the order MCC or FZC first, followed by DOC, AOS, BOS, ALP, VAL, and EXT (the Aufbau principle). AOS, BOS, and ALP are singly occupied MOs. ALP means a high spin alpha coupling, while AOS/BOS are an alpha/beta coupling to an open shell singlet. This requires the value NAOS=NBOS, and their MOs alternate. An example is

```
NFZC=1 NDOC=2 NAOS=2 NBOS=2 NALP=1 NVAL=3
```

which gives the reference CSF

```
FZC,DOC,DOC,AOS,BOS,AOS,BOS,ALP,VAL,VAL,VAL
```

This is a doublet state with five unpaired electrons. VAL orbitals are unoccupied only in the reference CSF, they will become occupied as the other CSFs are generated. This is done by giving an excitation level, either explicitly by the IEXCIT variable, or implicitly by the FORS, FOCl, or SOCl flags. One of these four keywords must be chosen, and during MCSCF runs, this is usually FORS.

Consider another simpler example, for an MCSCF run,

```
NMCC=3 NDOC=3 NVAL=2
```

which gives the reference CSF

```
MCC,MCC,MCC,DOC,DOC,DOC,VAL,VAL
```

having six electrons in five active orbitals. Usually, MCSCF calculations are usually of the Full Optimized Reaction Space (FORS) type. Some workers refer to FORS as CASSCF, complete active space SCF. These are the same, but the keyword is spelled FORS. In the present instance, choosing FORS=.TRUE. gives an excitation level of 4, as the 6 valence electrons have only 4 holes available for excitation. MCSCF runs typically have only a small number of VAL orbitals. It is common to summarize this example as "six electrons in five orbitals".

The next example is a first or second order multi-reference CI wavefunction, where

```
NFZC=3 NDOC=3 NVAL=2 NEXT=-1
```

leads to the reference CSF

```
FZC,FZC,FZC,DOC,DOC,DOC,VAL,VAL,EXT,EXT,...
```

FOCl or SOCl is chosen by selecting the appropriate flag, the correct excitation level is automatically generated. Note that the negative one for NEXT causes all remaining MOs to be included in the external orbital space. One way of viewing FOCl and SOCl wavefunctions is as all singles, or all singles and doubles, from the entire MCSCF wavefunction as a reference. An equivalent way of saying this is that all CSFs with N electrons (in this case N=6) distributed in the valence orbitals in all ways (that is the FORS MCSCF wavefunction) to make the base wavefunction. To this, FOCl adds all CSFs with N-1 electrons in active and 1 electron in external orbitals. SOCl adds all CSFs with N-2 electrons in active orbitals and 2 in external orbitals. SOCl is often prohibitively large, but is also a very accurate wavefunction.

Sometimes people use the CI package for ordinary single reference CI calculations, such as

```
NFZC=3 NDOC=5 NVAL=34
```

which means the reference RHF wavefunction is

```
FZC FZC FZC DOC DOC DOC VAL VAL ... VAL
```

and in this case NVAL is a large number conveying the total number of -virtual- orbitals into which electrons are excited. The excitation level would be given as IEXCIT=2, perhaps, to perform a SD-CI. All excitations smaller than the value of IEXCIT are automatically included in the CI. Note that NVAL's spelling was chosen to make the most sense for MCSCF calculations, and so it is a bit of a misnomer here.

Before going on, there is a quirk related to single reference CI that should be mentioned. Whenever the single reference contains unpaired electrons, such as

```
NFZC=3 NDOC=4 NALP=2 NVAL=33
```

some "extra" CSFs will be generated. The reference here can be abbreviated

```
2222 11 000 000 000 000 000 000 000 000 000 000
```

Supposing IEXCIT=2, the following CSF

```
2200 22 000 011 000 000 000 000 000 000 000 000
```

will be generated and used in the CI. Most people would prefer to think of this as a quadrupole excitation from the reference, but acting solely on the reasoning that no more than two electrons went into previously vacant NVAL orbitals, the GUGA CSF package decides it is a double. So, an open shell SD-CI calculation with GAMESS will not give the same result as other programs, although the result for any such calculation with these "extras" is correctly computed.

As was discussed above, the CSFs are automatically spin-symmetry adapted, with S implicit in the reference CSF. The spin quantum number you appear to be requesting in \$DRT (basically,  $S = \text{NALP}/2$ ) will be checked against the value of MULT in \$CONTRL, and the total number of electrons,  $2 * \text{NMCC}(\text{or NFZC}) + 2 * \text{NDOC} + \text{NAOS} + \text{NBOS} + \text{NALP}$  will be checked against the input given for ICHARG.

The CSF package is also able to exploit spatial symmetry, which like the spin and charge, is implicitly determined by the choice of the reference CSF. The keyword GROUP in \$DRT governs the use of spatial symmetry.

The CSF program works with Abelian point groups, which are  $D_{2h}$  and any of its subgroups. However, \$DRT allows the input of some (but not all) higher point groups. For non-Abelian groups, the program automatically assigns the orbitals to an irrep in the highest possible Abelian subgroup. For the other non-Abelian groups, you must at present select GROUP=C1. Note that when you are computing a Hessian matrix, many of the displaced geometries are asymmetric, hence you must choose C1 in \$DRT (however, be sure to use the highest symmetry possible in \$DATA!).

The symmetry of the reference CSF given in your \$DRT determines the symmetry of the CSFs which are generated. As an example, consider a molecule with Cs symmetry, and these two reference CSFs

```
...MCC...DOC DOC VAL VAL
...MCC...DOC AOS BOS VAL
```

Suppose that the 2nd and 3rd active MOs have symmetries a' and a". Both of these generate singlet wavefunctions, with 4 electrons in 4 active orbitals, but the former constructs 1-A' CSFs, while the latter generates 1-A" CSFs. However, if the 2nd and 3rd orbitals have the same symmetry type, an identical list of CSFs is generated.

In cases with high point group symmetry, it may be possible to generate correct state degeneracies only by using no symmetry (GROUP=C1) when generating CSFs. As an example, consider the 2-pi ground state of NO. If you use GROUP=C4V, which will be mapped into its highest Abelian subgroup  $C_{2v}$ , the two components of the pi state will be seen as belonging to different irreps, B1 and B2. The only way to ensure that both sets of CSFs are generated is to enforce no symmetry at all, so that CSFs for both components of the pi level are generated. This permits state averaging (WSTATE(1)=0.5,0.5) to preserve cylindrical symmetry. It is however perfectly feasible to use  $C_{4v}$  or  $D_{4h}$  symmetry in \$DRT when treating sigma states.

The use of spatial symmetry decreases the number of CSFs, and thus the size of the Hamiltonian that must be computed. In molecules with high symmetry, this may lead to faster run times with the GUGA CSF code, compared to the determinant code.

--- starting orbitals ---

The first step is to partition the orbital space into core, active, and external sets, in a manner which is sensible for your chemical problem. This is a bit of an art, and the user is referred to the references quoted at the end of this section. Having decided what MCSCF to perform, you now must consider the more pedantic problem of what orbitals to begin the MCSCF calculation with.

You should always start an MCSCF run with orbitals from some other run, by means of GUESS=MOREAD. Do not expect to be able to use HCORE or HUCKEL! Example 6 is a poor example, converging only because methylene has so much symmetry, and the basis is so small. If you are beginning your MCSCF problem, use orbitals from some appropriate converged SCF run. Thus, a realistic example of an MCSCF calculation is examples 8 and 9. Once you get an MCSCF to converge, you can and should use these MCSCF MOs (which will be Schmidt orthogonalized) at other nearby geometries.

Starting from SCF orbitals can take a little bit of care. Most of the time (but not always) the orbitals you want to correlate will be the highest occupied orbitals in the SCF. Fairly often, however, the correlating orbitals you wish to use will not be the lowest unoccupied virtuals of the SCF. You will soon become familiar with NORDER=1 in \$GUESS, as reordering is needed in 50% or more cases.

The occupied and especially the virtual canonical SCF MOs are often spread out over regions of the molecule other than "where the action is". Orbitals which remedy this can be generated by two additional options at almost no CPU cost.

One way to improve upon the SCF orbitals as starting MOs is to generate modified virtual orbitals (MVOs). MVOs are obtained by diagonalizing the Fock operator of a very positive ion, within the virtual orbital space only. As implemented in GAMESS, MVOs can be obtained at the end of any RHF, ROHF, or GVB run by setting MVOQ in \$SCF nonzero, at the cost of a single SCF cycle. Typically, we use MVOQ=+6. Generating MVOs does not change any of the occupied SCF orbitals of the original neutral, but gives more valence-like LUMOs.

Another way to improve SCF starting orbitals is by a partial localization of the occupied orbitals. Typically MCSCF active orbitals are concentrated in the part of the molecule where bonds are breaking, etc. Canonical SCF MOs are normally more spread out. By choosing LOCAL=BOYS along with SYMLOC=.TRUE. in \$LOCAL, you can get orbitals which are localized, but still retain orbital symmetry to help speed the MCSCF along. In groups with an inversion center, a SYMLOC Boys localization does not change the orbitals, but you can instead use LOCAL=POP. Localization tends to order the orbitals fairly randomly, so be prepared to reorder them appropriately.

Pasting the virtuals from a MVOQ run onto the occupied orbitals of a SYMLOC run (both can be done in the same SCF computation) gives the best possible set of starting orbitals. If you also take the time to design your active space carefully, select the appropriate starting orbitals from this combined \$VEC, and inspect your converged results, you will be able to carry out MCSCF computations correctly.

Convergence of MCSCF is by no means guaranteed. Poor convergence can invariably be traced back to either a poor initial selection of orbitals, or a poorly chosen number of active orbitals. My advice is, before you even start:

"Look at the orbitals.

Then look at the orbitals again".

Later, if you have any trouble:

"Look at the orbitals some more".

Few people are able to see the orbital shapes in the LCAO matrix in a log file, and so need a

visualization program. If you have a Macintosh, download a copy of MacMolPlt from <http://www.msg.ameslab.gov/GAMESS/GAMESS.html> for 2D or 3D plots, or use PLTORB under X-windows for 2D.

Even if you don't have any trouble, look at the orbitals to see if they converged to what you expected, and have reasonable occupation numbers. MCSCF is by no means the sort of "black box" that RHF is these days, so LOOK VERY CAREFULLY AT YOUR RESULTS.

--- miscellaneous hints ---

It is very helpful to execute a EXETYP=CHECK run before doing any MCSCF or CI run. The CHECK run will tell you the total number of configurations and the charge and multiplicity based on your \$DET, or if you are using a \$DRT, the electronic state as well. The CHECK run also lets the program feel out the memory that will be required to actually do the run. Thus the CHECK run can potentially prevent costly mistakes, or tell you when a calculation is prohibitively large.

A very common MCSCF wavefunction has 2 electrons in 2 active MOs. This is the simplest possible wavefunction describing a singlet diradical. While this function can be obtained in an MCSCF run (using NACT=2 NELS=2 or NDOC=1 NVAL=1), it can be obtained much faster by use of the GVB code, with one GVB pair. This GVB-PP(1) wavefunction is also known in the literature as two configuration SCF, or TCSCF. The two configurations of this GVB are equivalent to the three configurations used in this MCSCF, as orbital optimization in natural form (configurations 20 and O<sub>2</sub>) causes the coefficient of the 11 configuration to vanish.

If you are using many configurations (say 75,000 or more), the main bottleneck in the GAMESS CI/MCSCF code is the formation and diagonalization of the Hamiltonian, not the integral transformation and orbital improvement steps. In this case, you would be wise to switch to FULLNR, which will minimize the total number of iterations. In addition, each orbital improvement may contain some microiterations, which consists of an integral transformation over the new MOs, followed immediately by a orbital improvement, reusing the current 2nd order density matrix. This avoids the CI diagonalization step, which may be of some use in MCSCF calculations with a large number of configurations. Since the determinant CI is a direct CI, it is probably better to use it in this circumstance to avoid the very large disk file used to store the CSF Hamiltonian, and its associated I/O bottleneck.

If you choose an excitation level IEXCIT smaller than that needed to generate the FORS space, you must use the FULLNR method (FOCAS and SOSCF convergers assume complete active spaces). Be sure to set FORS=.FALSE. in \$MCSCF or else very poor convergence will result. Actually, the convergence for incomplete active spaces is likely to be awful, anyway.

--- references ---

There are several review articles about MCSCF listed below. Of these, the first two are a nice overview of the subject, the final 3 are more technical.

1. "The Construction and Interpretation of MCSCF wavefunctions"  
M.W.Schmidt and M.S.Gordon, Ann.Rev.Phys.Chem. 49,233-266 (1998 )
- 2a. "The Multiconfiguration SCF Method" B.O.Roos, in "Methods in Computational Molecular Physics", edited by G.H.F.Diercksen and S.Wilson D.Reidel Publishing, Dordrecht, Netherlands, 1983 , pp 161-187.
- 2b."The Multiconfiguration SCF Method" B.O.Roos, in "Lecture Notes in Quantum Chemistry", edited by B.O.Roos, Lecture Notes in Chemistry v58, Springer-Verlag, Berlin, 1994 , pp 177-254.

3. "Optimization and Characterization of a MCSCF State" J.Olsen, D.L.Yeager, P.Jorgensen Adv.Chem.Phys. 54, 1-176(1983).
4. "Matrix Formulated Direct MCSCF and Multiconfiguration Reference CI Methods" H.-J.Werner, Adv.Chem.Phys. 69, 1-62(1987).
5. "The MCSCF Method" R.Shepard, Adv.Chem.Phys. 69, 63-200(1987).

There is an entire section on the choice of active spaces in Reference 1. As this is a matter of great importance, here are two alternate presentations of the design of active spaces:

6. "The CASSCF Method and its Application in Electronic Structure Calculations" B.O.Roos, in "Advances in Chemical Physics", vol.69, edited by K.P.Lawley, Wiley Interscience, New York, 1987, pp 339-445.
7. "Are Atoms Intrinsic to Molecular Electronic Wavefunctions?" K.Ruedenberg, M.W.Schmidt, M.M.Gilbert, S.T.Elbert Chem.Phys. 71, 41-49, 51-64, 65-78 (1982).

Two papers germane to the FOCAS implementation are

8. "An Efficient first-order CASSCF method based on the renormalized Fock-operator technique." U.Meier, V.Staemmler Theor.Chim.Acta 76, 95-111(1989)
9. "Modern tools for including electron correlation in electronic structure studies" M.Dupuis, S.Chen, A.Marquez, in "Relativistic and Electron Correlation Effects in Molecules and Solids", edited by G.L.Malli, Plenum, NY 1994

The paper germane to the the SOSCF method is

- 10."Approximate second order method for orbital optimization of SCF and MCSCF wavefunctions" G.Chaban, M.W.Schmidt, M.S.Gordon Theor.Chem.Acc. 97: 88-95(1997)

Two papers germane to the FULLNR implementation are

- 11."General second order MCSCF theory: A Density Matrix Directed Algorithm" B.H.Lengsfeld, III, J.Chem.Phys. 73,382-390(1980).
- 12."The use of the Augmented Matrix in MCSCF Theory" D.R.Yarkony, Chem.Phys.Lett. 77,634-635(1981).

For determinants and CSFs, respectively, see

- 13."A new determinant-based full CI method" P.J.Knowles, N.C.Handy Chem.Phys.Lett. 111,315-321(1984)
- 14."The GUGA approach to the electron correlation problem" B.R.Brooks, H.F.Schaefer J.Chem.Phys. 70, 5092-5106(1979)

The final references are simply some examples of FORS MCSCF applications, the latter done with GAMESS.

- 15.D.F.Feller, M.W.Schmidt, K.Ruedenberg, J.Am.Chem.Soc. 104, 960-967(1982).
- 16.M.W.Schmidt, P.N.Truong, M.S.Gordon, J.Am.Chem.Soc. 109, 5217-5227(1987).

## Second order perturbation theory

The perturbation theory techniques available in GAMESS expand to the second order energy correction only, but permit use of a broad range of zeroth order wavefunctions. Since MP2 theory for systems well described as closed shells recovers only about 80% of the correlation energy (assuming the use of large basis sets), it is common to extend the perturbative treatment to higher order, or to use coupled cluster theory. While this is reasonable for systems well described by RHF or UHF with small spin contamination, this is probably a poor approach when the system is not well described at zeroth order by these wavefunctions.

The input for second order perturbation calculations based on SCFTYP=RHF, UHF, or ROHF is found in \$MP2, while for SCFTYP=MCSCF, see \$MCQDPT.

### --- RHF and UHF MP2

These methods are well defined, due to the uniqueness of the Fock matrix definitions. These methods are also well understood, and there is little need to say more.

One point which may not be commonly appreciated is that the density matrix for the first order wavefunction for the RHF case, which is generated during gradient runs or if properties are requested in the \$MP2 group, is of the type known as "response density", which differs from the more usual "expectation value density". The eigenvalues of the response density matrix (which are the occupation numbers of the MP2 natural orbitals) can therefore be greater than 2 for frozen core orbitals, or even negative values for the highest 'virtual' orbitals. The sum is of course exactly the total number of electrons. We have seen values outside the range 0-2 in several cases where the single configuration RHF wavefunction was not an appropriate description of the system, and thus these occupancies may serve as a guide to the wisdom of using a RHF reference.

RHF energy corrections are the only method currently programmed for parallel computation. The analytic energy gradient is available only for RHF references, and this does permit frozen cores.

### --- high spin ROHF MP2

There are a number of open shell perturbation theories described in the literature. It is important to note that these methods give different results for the second order energy correction, reflecting ambiguities in the selection of the zeroth order Hamiltonian and in defining the ROHF Fock matrices. Two of these are available in GAMESS.

One theory is known as RMP, which it should be pointed out, is entirely equivalent to the ROHF-MBPT2 method. The theory is as UHF-like as possible, and can be chosen in GAMESS by selection of OSPT=RMP in \$MP2. The second order energy is defined by

1. P.J.Knowles, J.S.Andrews, R.D.Amos, N.C.Handy, J.A.Pople *Chem.Phys.Lett.* 186, 130-136(1991)
2. W.J.Lauderdale, J.F.Stanton, J.Gauss, J.D.Watts, R.J.Bartlett *Chem.Phys.Lett.* 187, 21-28(1991).

The submission dates are in inverse order of publication dates, and -both- papers should be cited when using this method. Here we will refer to the method as RMP in keeping with much of the literature. The RMP method diagonalizes the alpha and beta Fock matrices separately, so that their occupied-occupied and virtual-virtual blocks are canonicalized. This generates two distinct orbital sets, whose double excitation contributions are processed by the usual UHF MP2 program, but an additional energy term from single excitations is required.



RMP's use of different orbitals for different spins adds to the CPU time required for integral transformations, of course. RMP is invariant under all of the orbital transformations for which the ROHF itself is invariant. Unlike UMP2, the second order RMP energy does not suffer from spin contamination, since the reference ROHF wavefunction has no spin contamination. The RMP wavefunction, however, is spin contaminated at 1st and higher order, and therefore the 3rd and higher order RMP energies are spin contaminated. Other workers have extended the RMP theory to gradients and Hessians at second order, and to fourth order in the energy,

3. W.J.Lauderdale, J.F.Stanton, J.Gauss, J.D.Watts, R.J.Bartlett *J.Chem.Phys.* 97, 6606-6620(1992)
4. J.Gauss, J.F.Stanton, R.J.Bartlett, *J.Chem.Phys.* 97, 7825-7828(1992)
5. D.J.Tozer, J.S.Andrews, R.D.Amos, N.C.Handy *Chem.Phys.Lett.* 199, 229-236(1992)
6. D.J.Tozer, N.C.Handy, R.D.Amos, J.A.Pople, R.H.Nobes, Y.Xie, H.F.Schaefer *Mol.Phys.* 79, 777-793(1993)

We deliberately omit references to the RMP precursor to the RMP formalism.

The ZAPT formalism is also implemented in GAMESS, as OSPT=ZAPT in \$MP2. Because this theory is not spin-contaminated at any order, and has only a single set of orbitals in the MO transformation, it is the default. References for ZAPT (Z-averaged perturbation theory) are

7. T.J.Lee, D.Jayatilaka *Chem.Phys.Lett.* 201, 1-10(1993)
8. T.J.Lee, A.P.Rendell, K.G.Dyall, D.Jayatilaka *J.Chem.Phys.* 100, 7400-7409(1994)

The formulae for the seven terms in the energy are most clearly summarized in the paper

9. I.M.B.Nielsen, E.T.Seidl, *J.Comput.Chem.* 16, 1301-1313(1995)

We would like to thank Tim Lee for very gracious assistance in the implementation of ZAPT.

There are a number of other open shell theories, with names such as HC, OPT1, OPT2, and IOPT. The literature for these is

- 10.I.Hubac, P.Carsky *Phys.Rev.A* 22, 2392-2399(1980)
- 11.C.Murray, E.R.Davidson, *Chem.Phys.Lett.* 187,451-454(1991)
- 12.C.Murray, E.R.Davidson, *Int.J.Quantum Chem.* 43, 755-768(1992)
- 13.P.M.Kozlowski, E.R.Davidson, *Chem.Phys.Lett.* 226, 440-446(1994)
- 14.C.W.Murray, N.C.Handy, *J.Chem.Phys.* 97, 6509-6516(1992)
- 15.T.D.Crawford, H.F.Schaefer, T.J.Lee, *J.Chem.Phys.* 105, 1060-1069(1996)

The latter two of these give comparisons of the various high spin methods, and the numerical results in ref. 15 are the basis for the conventional wisdom that restricted open shell theory is better convergent with order of the perturbation level than unrestricted theory. Paper 8 has some numerical comparisons of spin-restricted theories as well. We are aware of one paper on low-spin coupled open shell SCF perturbation theory

- 16.J.S.Andrews, C.W.Murray, N.C.Handy, *Chem.Phys.Lett.* 201, 458-464(1993) but this is not implemented in GAMESS. See the MCQDPT code for cases such as this.

--- GVB based MP2

This is not implemented in GAMESS. Note that the MCSCF MP2 program discussed below should be able to develop the perturbation correction for open shell singlets, by using a \$DRT input such as

```
NMCC=N/2-1 NDOC=0 NAOS=1 NBOS=1 NVAL=0
```

which generates a single CSF if the two open shells have different symmetry, or for a one pair GVB function

```
NMCC=N/2-1 NDOC=1 NVAL=1
```

which generates a 3 CSF function entirely equivalent to the two configuration TCSCF, a.k.a GVB-PP(1). For the record, we note that if we attempt a triplet state with the MCSCF program,

```
NMCC=N/2-1 NDOC=0 NALP=2 NVAL=0
```

we get a result equivalent to the OPT1 open shell method described above, not the RMP result. It is possible to generate the orbitals with a simpler SCF computation than the MCSCF \$DRT examples just given, and read them into the MCSCF based MP2 program described below, by INORB=1.

--- MCSCF based MP2

Just as for the open shell case, there are several ways to define a multireference perturbation theory. The most noteworthy are the CASPT2 method of Roos' group, the MRMP2 method of Hirao, the MROPTn methods of Davidson, and the MCQDPT2 method of Nakano. Although the results of each method are different, energy differences should be rather similar. In particular, the MCQDPT2 method implemented in GAMESS gives results for the singlet-triplet splitting of methylene in close agreement to CASPT2, MRMP2(Fav), and MROPT1, and differs by 2 Kcal/mole from MRMP2(Fhs), and the MROPT2 to MROPT4 methods.

The MCQDPT method implemented in GAMESS is a multistate perturbation theory. If applied to 1 state, it is similar to the MRMP model of Hirao. When applied to more than one state, it is of the philosophy "perturb first, diagonalize second". This means that perturbations are made to both the diagonal and offdiagonal elements of an effective Hamiltonian, whose dimension equals the number of states being treated. The perturbed Hamiltonian is diagonalized to give the corrected state energies. Diagonalization after inclusion of the offdiagonal perturbation ensures that avoided crossings of states of the same symmetry are treated correctly. Such an avoided crossing is found in the LiF molecule, as shown in the first of the two papers on the MCQDPT method:

H.Nakano, J.Chem.Phys. 99, 7983-7992(1993 )

H.Nakano, Chem.Phys.Lett. 207, 372-378(1993 )

The closely related "diagonalize, then perturb" MRMP model is discussed by

K.Hirao, Chem.Phys.Lett. 190, 374-380(1992 )

K.Hirao, Chem.Phys.Lett. 196, 397-403(1992 )

K.Hirao, Int.J.Quant.Chem. S26, 517-526(1992 )

K.Hirao, Chem.Phys.Lett. 201, 59-66(1993 )

Computation of reference weights and energy contributions is illustrated by

H.Nakano, K.Nakayama, K.Hirao, M.Dupuis J.Chem.Phys. 106, 4912-4917(1997)

T.Hashimoto, H.Nakano, K.Hirao J.Mol.Struct.(THEOCHEM) 451, 25-33(1998)

Single state MCQDPT computations are very similar to MRMP computations. A beginning set of references to the other multireference methods used includes:

P.M.Kozlowski, E.R.Davidson J.Chem.Phys. 100, 3672-3682(1994 )

K.G.Dyall J.Chem.Phys. 102, 4909-4918(1995 )

B.O.Roos, K.Andersson, M.K.Fulscher, P.-A.Malmqvist, L.Serrano-Andres, K.Pierloot, M.Merchan Adv.Chem.Phys. 93, 219-331(1996 )

The MCQDPT code was written by Haruyuki Nakano, and was interfaced to GAMESS by him in the summer of 1996. After a few months experience, we can say that this code seems to run in memory, disk, and CPU time comparable to the MCSCF computation itself. It has been used on a 230 AO problem, for example. Efficiency is improved if you can add extra physical memory to reduce the number of file reads.

We close the discussion with an input example which illustrates RMP and MCQDPT computations on the ground state of NH2 radical:

```
! 2nd order perturbation test on NH2, following
! T. J. Lee, A. P. Rendell, K. G. Dyall, D. Jayatilaka
! J. Chem. Phys. 100, 7400-7409(1994), Table III.
! State is 2-B-1, 69 AOs, 49 CSFs.
```

```

!
! For 1 CSF reference,
!   E(ROHF) = -55.5836109825
!   E(RMP)  = -55.7772299929 (lit. RMP = -75.777230)
!   E(MCQDPT) = -55.7830423024 (lit. OPT1= -75.783044)
! [E(MCQDPT) = -55.7830437413 at the lit's OPT1 geometry]
!
! For 49 CSF reference,
!   E(MCSCF) = -55.6323324949
!   E(MCQDPT) = -55.7857458575
!
! $contrl scftyp=mcscf mplevl=2 runtyp=energy mult=2 $end
! $system timlim=60 memory=1000000 $end
! $guess guess=moread norb=69 $end
! $mcscf fullnr=.true. $end
!
! Next two lines carry out a MCQDPT computation, after
! carrying out a full valence MCSCF orbital optimization.
!
! $drtd group=c2v fors=.t. nmcc=1 ndoc=3 nalp=1 nval=2 $end
! $mcqdpt inorb=0 mult=2 nmofzc=1 nmodoc=0 nmoact=6
!       istsym=3 nstate=1 $end
!
! Next two lines carry out a single reference computation,
! using converged ROHF orbitals from the $VEC.
!
! --- $drtd group=c2v fors=.t. nmcc=4 ndoc=0 nalp=1 nval=0 $end
! --- $mcqdpt inorb=1 nmofzc=1 nmodoc=3 nmoact=1
! ---       istsym=3 nstate=1 $end
!
! $data
! NH2...2-B-1...TZ2Pf basis, RMP geom. used by Lee, et al.
! Cnv 2
!
Nitrogen 7.0
S 6
  1 13520.0 0.000760
  2 1999.0 0.006076
  3 440.0 0.032847
  4 120.9 0.132396
  5 38.47 0.393261
  6 13.46 0.546339
S 2
  1 13.46 0.252036
  2 4.993 0.779385
S 1 ; 1 1.569 1.0
S 1 ; 1 0.5800 1.0
S 1 ; 1 0.1923 1.0
P 3
  1 35.91 0.040319
  2 8.480 0.243602
  3 2.706 0.805968
P 1 ; 1 0.9921 1.0
P 1 ; 1 0.3727 1.0
P 1 ; 1 0.1346 1.0
D 1 ; 1 1.654 1.0
D 1 ; 1 0.469 1.0
F 1 ; 1 1.093 1.0
!
Hydrogen 1.0 0.0 0.7993787 0.6359684
S 3 ! note that this is unscaled
  1 33.64 0.025374
  2 5.058 0.189684
  3 1.147 0.852933

```

```

S 1 ; 1 0.3211 1.0
S 1 ; 1 0.1013 1.0
P 1 ; 1 1.407 1.0
P 1 ; 1 0.388 1.0
D 1 ; 1 1.057 1.0

```

```

Send
E(ROHF) = -55.5836109825, E(NUC) = 7.5835449477, 9 ITERs
$VEC
 1 1 5.58322965E-01...
... omitted...
69 14 2.48059888E-02...
$SEND

```

## Density Functional Theory

The present code is a research tool into the use of the resolution of the identity to simplify evaluation of integral over functionals, rather than quadrature grids. This trades the use of finite grids and their associated errors for the use of a finite basis set used to expand the identity, with an associated truncation error. The present RI auxiliary basis sets obtain energies close to the correct total ones, but their exponents have not been optimized. Further work will be needed to obtain auxiliary basis sets that hopefully might be both more accurate and also smaller (and therefore faster). The present choice of auxiliary basis sets was obtained by tests on small 2nd row molecules like NH<sub>3</sub> and N<sub>2</sub>, and hence the built in bases for the 3rd row are not as well developed. Auxiliary bases for the remaining elements do not exist at the present time.

The total energy presently varies by about 5D-8 for molecules computed on different machines. This is believed to be due to some redundancies in the auxiliary RI basis sets. For example, at a C1 AM1 geometry for ethanol, the Becke/6-31G(d) energy is

```

-154.0845919891 Linux
-154.0845920132 IBM RS/6000
-154.0845919997 Sun

```

which at 3d-8 is two digits more than the energy change for other methods in GAMESS when the computer type is changed. To illustrate the error from the RI basis for a case with three second row atoms, the result from MOLPRO for ethanol using grids is -154.105052.

The present gradient code contains some errors, and consequently no one should trust these results. As an example, a Becke/6-31G(d) geometry optimization of water started from the EXAM08 geometry behaves as:

```

FINAL E= -76.0439853638, RMS GRADIENT = .0200293
FINAL E= -76.0413274662, RMS GRADIENT = .0231574
FINAL E= -76.0455283912, RMS GRADIENT = .0045887
FINAL E= -76.0457360477, RMS GRADIENT = .0009356
FINAL E= -76.0457239113, RMS GRADIENT = .0001222
FINAL E= -76.0457216186, RMS GRADIENT = .0000577
FINAL E= -76.0457202264, RMS GRADIENT = .0000018
FINAL E= -76.0457202253, RMS GRADIENT = .0000001

```

The MOLPRO result at the initial geometry is -76.047441. Examination shows that the point on the PES where the gradient is zero is not where the energy is lowest, in fact the 4th geometry is the lowest encountered.

The behaviour for Becke/6-31G(d) ethanol is as follows:

```

FINAL E= -154.0845920132, RMS GRADIENT = .0135540

```

```

FINAL E= -154.0933138447, RMS GRADIENT = .0052778
FINAL E= -154.0885472996, RMS GRADIENT = .0009306
FINAL E= -154.0886268185, RMS GRADIENT = .0002043
FINAL E= -154.0886352947, RMS GRADIENT = .0000795
FINAL E= -154.0885599794, RMS GRADIENT = .0000342
FINAL E= -154.0885514829, RMS GRADIENT = .0000679
FINAL E= -154.0884955093, RMS GRADIENT = .0000205
FINAL E= -154.0886438244, RMS GRADIENT = .0000330
FINAL E= -154.0886596883, RMS GRADIENT = .0000325
FINAL E= -154.0886094081, RMS GRADIENT = .0000120
FINAL E= -154.0886054003, RMS GRADIENT = .0000109
FINAL E= -154.0885939751, RMS GRADIENT = .0000152
FINAL E= -154.0886711482, RMS GRADIENT = .0000439
FINAL E= -154.0886972557, RMS GRADIENT = .0000230

```

with similar fluctuations through a total of 50 steps without locating a zero gradient. Note that the second energy above is substantially below all later points, so geometry optimizations with the grid-free DFT gradient code are at this time unsatisfactory.

The reason for including the grid-free code is to facilitate future developments in auxiliary basis sets, and hopefully to locate the problem in the gradient code. Perhaps not very many people will want to use this as a code for production computations, in its present form.

Some reading on the grid-free approach to density functional theory is:

Y.C.Zheng, J.Almlof Chem.Phys.Lett. 214, 397-401(1996)

Y.C.Zheng, J.Almlof J.Mol.Struct.(Theochem) 288, 277(1996)

K.Glaesemann, M.S.Gordon J.Chem.Phys. 108, 9959-9969(1998)

K.Glaesemann, M.S.Gordon J.Chem.Phys. 110, 6580-6582(1999)

K.Glaesemann, M.S.Gordon J.Chem.Phys. in press

## Geometry Searches and Internal Coordinates

Stationary points are places on the potential energy surface with a zero gradient vector (first derivative of the energy with respect to nuclear coordinates). These include minima (whether relative or global), better known to chemists as reactants, products, and intermediates; as well as transition states (also known as saddle points).

The two types of stationary points have a precise mathematical definition, depending on the curvature of the potential energy surface at these points. If all of the eigenvalues of the hessian matrix (second derivative of the energy with respect to nuclear coordinates) are positive, the stationary point is a minimum. If there is one, and only one, negative curvature, the stationary point is a transition state. Points with more than one negative curvature do exist, but are not important in chemistry. Because vibrational frequencies are basically the square roots of the curvatures, a minimum has all real frequencies, and a saddle point has one imaginary vibrational "frequency".

GAMESS locates minima by geometry optimization, as RUNTYP=OPTIMIZE, and transition states by saddle point searches, as RUNTYP=SADPOINT. In many ways these are similar, and in fact nearly identical FORTRAN code is used for both. The term "geometry search" is used here to describe features which are common to both procedures. The input to control both RUNTYPs is found in the \$STATPT group.

As will be noted in the symmetry section below, an OPTIMIZE run does not always find a

minimum, and a SADPOINT run may not find a transition state, even though the gradient is brought to zero. You can prove you have located a minimum or saddle point only by examining the local curvatures of the potential energy surface. This can be done by following the geometry search with a RUNTYP=HESSIAN job, which should be a matter of routine.

### \* \* \* Quasi-Newton Searches \* \* \*

Geometry searches are most effectively done by what is called a quasi-Newton-Raphson procedure. These methods assume a quadratic potential surface, and require the exact gradient vector and an approximation to the hessian. It is the approximate nature of the hessian that makes the method "quasi". The rate of convergence of the geometry search depends on how quadratic the real surface is, and the quality of the hessian. The latter is something you have control over, and is discussed in the next section.

GAMESS contains different implementations of quasi-Newton procedures for finding stationary points, namely METHOD=NR, RFO, QA, and the seldom used SCHLEGEL. They differ primarily in how the step size and direction are controlled, and how the Hessian is updated. The CONOPT method is a way of forcing a geometry away from a minimum towards a TS. It is not a quasi-Newton method, and is described at the very end of this section.

The NR method employs a straight Newton-Raphson step. There is no step size control, the algorithm will simply try to locate the nearest stationary point, which may be a minimum, a TS, or any higher order saddle point. NR is not intended for general use, but is used by GAMESS in connection with some of the other methods after they have homed in on a stationary point, and by Gradient Extremal runs where location of higher order saddle points is common. NR requires a very good estimate of the geometry in order to converge on the desired stationary point.

The RFO and QA methods are two different versions of the so-called augmented Hessian techniques. They both employ Hessian shift parameter(s) in order to control the step length and direction.

In the RFO method, the shift parameter is determined by approximating the PES with a Rational Function, instead of a second order Taylor expansion. For a RUNTYP=SADPOINT, the TS direction is treated separately, giving two shift parameters. This is known as a Partitioned Rational Function Optimization (P-RFO). The shift parameter(s) ensure that the augmented Hessian has the correct eigenvalue structure, all positive for a minimum search, and one negative eigenvalue for a TS search. The (P)-RFO step can have any length, but if it exceeds DXMAX, the step is simply scaled down.

In the QA (Quadratic Approximation) method, the shift parameter is determined by the requirement that the step size should equal DXMAX. There is only one shift parameter for both minima and TS searches. Again the augmented Hessian will have the correct structure. There is another way of describing the same algorithm, namely as a minimization on the "image" potential. The latter is known as TRIM (Trust Radius Image Minimization). The working equation is identical in these two methods.

When the RFO steplength is close to DXMAX, there is little difference between the RFO and QA methods. However, the RFO step may in some cases exceed DXMAX significantly, and a simple scaling of the step will usually not produce the best direction. The QA step is the best step on the hypersphere with radius DXMAX. For this reason QA is the default algorithm.

Near a stationary point the straight NR algorithm is the most efficient. The RFO and QA may be viewed as methods for guiding the search in the "correct" direction when starting far from

the stationary point. Once the stationary point is approached, the RFO and QA methods switch to NR, automatically, when the NR steplength drops below 0.10 or DXMAX, whichever is the smallest.

The QA method works so well that we use it exclusively, and so the SCHLEGEL method will probably be omitted from some future version of GAMESS.

You should read the papers mentioned below in order to understand how these methods are designed to work. The first 3 papers describe the RFO and TRIM/QA algorithms. A good but somewhat dated summary of various search procedures is given by Bell and Crighton, and see also the review by Schlegel. Most of the FORTRAN code for geometry searches, and some of the discussion in this section was written by Frank Jensen of Odense University, whose paper compares many of the algorithms implemented in GAMESS:

1. J.Baker J.Comput.Chem. 7, 385-395(1986 )
2. T.Helgaker Chem.Phys.Lett. 182, 305-310(1991 )
3. P.Culot, G.Dive, V.H.Nguyen, J.M.Ghuysen Theoret.Chim.Acta 82, 189-205(1992 )
4. H.B.Schlegel J.Comput.Chem. 3, 214(1982 )
5. S.Bell, J.S.Crighton J.Chem.Phys. 80, 2464-2475, (1984 ).
6. H.B.Schlegel Advances in Chemical Physics (Ab Initio Methods in Quantum Chemistry, Part I), volume 67, K.P.Lawley, Ed. Wiley, New York, 1987, pp 249-286.
7. F.Jensen J.Chem.Phys. 102, 6706-6718(1995 ).

#### \* \* \* the Hessian \* \* \*

Although quasi-Newton methods require only an approximation to the true hessian, the choice of this matrix has a great affect on convergence of the geometry search.

There is a procedure contained within GAMESS for guessing a diagonal, positive definite hessian matrix, HESS=GUESS. If you are using Cartesian coordinates, the guess hessian is 1/3 times the unit matrix. The guess is more sophisticated when internal coordinates are defined, as empirical rules will be used to estimate stretching and bending force constants. Other force constants are set to 1/4. The diagonal guess often works well for minima, but cannot possibly find transition states (because it is positive definite). Therefore, GUESS may not be selected for SADPOINT runs.

Two options for providing a more accurate hessian are HESS=READ and CALC. For the latter, the true hessian is obtained by direct calculation at the initial geometry, and then the geometry search begins, all in one run. The READ option allows you to feed in the hessian in a \$HESS group, as obtained by a RUNTYP=HESSIAN job. The second procedure is actually preferable, as you get a chance to see the frequencies. Then, if the local curvatures look good, you can commit to the geometry search. Be sure to include a \$GRAD group (if the exact gradient is available) in the HESS=READ job so that GAMESS can take its first step immediately.

Note also that you can compute the hessian at a lower basis set and/or wavefunction level, and read it into a higher level geometry search. In fact, the \$HESS group could be obtained at the semiempirical level. This trick works because the hessian is  $3N \times 3N$  for  $N$  atoms, no matter what atomic basis is used. The gradient from the lower level is of course worthless, as the geometry search must work with the exact gradient of the wavefunction and basis set in current use. Discard the \$GRAD group from the lower level calculation!

You often get what you pay for. HESS=GUESS is free, but may lead to significantly more steps in the geometry search. The other two options are more expensive at the beginning, but may pay back by rapid convergence to the stationary point.

The hessian update frequently improves the hessian for a few steps (especially for HESS=GUESS), but then breaks down. The symptoms are a nice lowering of the energy or the RMS gradient for maybe 10 steps, followed by crazy steps. You can help by putting the best coordinates into \$DATA, and resubmitting, to make a fresh determination of the hessian.

The default hessian update for OPTIMIZE runs is BFGS, which is likely to remain positive definite. The POWELL update is the default for SADPOINT runs, since the hessian can develop a negative curvature as the search progresses. The POWELL update is also used by the METHOD=NR and CONOPT since the Hessian may have any number of negative eigenvalues in these cases. The MSP update is a mixture of Murtagh-Sargent and Powell, suggested by Josep Bofill, (J.Comput.Chem., 15, 1-11, 1994 ). It sometimes works slightly better than Powell, so you may want to try it.

### \* \* \* Coordinate Choices \* \* \*

Optimization in cartesian coordinates has a reputation of converging slowly. This is largely due to the fact that translations and rotations are usually left in the problem. Numerical problems caused by the small eigenvalues associated with these degrees of freedom are the source of this poor convergence. The methods in GAMESS project the hessian matrix to eliminate these degrees of freedom, which should not cause a problem. Nonetheless, Cartesian coordinates are in general the most slowly convergent coordinate system.

The use of internal coordinates (see NZVAR in \$CONTRL as well as \$ZMAT) also eliminates the six rotational and translational degrees of freedom. Also, when internal coordinates are used, the GUESS hessian is able to use empirical information about bond stretches and bends. On the other hand, there are many possible choices for the internal coordinates, and some of these may lead to much poorer convergence of the geometry search than others. Particularly poorly chosen coordinates may not even correspond to a quadratic surface, thereby ending all hope that a quasi-Newton method will converge.

Internal coordinates are frequently strongly coupled. Because of this, Jerry Boatz has called them "infernial coordinates"! A very common example to illustrate this might be a bond length in a ring, and the angle on the opposite side of the ring. Clearly, changing one changes the other simultaneously. A more mathematical definition of "coupled" is to say that there is a large off-diagonal element in the hessian. In this case convergence may be unsatisfactory, especially with a diagonal GUESS hessian, where a "good" set of internals is one with a diagonally dominant hessian. Of course, if you provide an accurately computed hessian, it will have large off-diagonal values where those are truly present. Even so, convergence may be poor if the coordinates are coupled through large 3rd or higher derivatives. The best coordinates are therefore those which are the most "quadratic".

One very popular set of internal coordinates is the usual "model builder" Z-matrix input, where for N atoms, one uses N-1 bond lengths, N-2 bond angles, and N-3 bond torsions. The popularity of this choice is based on its ease of use in specifying the initial molecular geometry. Typically, however, it is the worst possible choice of internal coordinates, and in the case of rings, is not even as good as Cartesian coordinates.

However, GAMESS does not require this particular mix of the common types. GAMESS' only requirement is that you use a total of 3N-6 coordinates, chosen from these 3 basic types, or several more exotic possibilities. (Of course, we mean 3N-5 throughout for linear molecules). These additional types of internal coordinates include linear bends for 3 collinear atoms, out of plane bends, and so on. There is no reason at all why you should place yourself in a



straightjacket of N-1 bonds, N-2 angles, and N-3 torsions. If the molecule has symmetry, be sure to use internals which are symmetrically related.

For example, the most effective choice of coordinates for the atoms in a four membered ring is to define all four sides, any one of the internal angles, and a dihedral defining the ring pucker. For a six membered ring, the best coordinates seem to be 6 sides, 3 angles, and 3 torsions. The angles should be every other internal angle, so that the molecule can "breathe" freely. The torsions should be arranged so that the central bond of each is placed on alternating bonds of the ring, as if they were pi bonds in Kekule benzene. For a five membered ring, we suggest all 5 sides, 2 internal angles, again alternating every other one, and 2 dihedrals to fill in. The internal angles of necessity skip two atoms where the ring closes. Larger rings should generalize on the idea of using all sides but only alternating angles. If there are fused rings, start with angles on the fused bond, and alternate angles as you go around from this position.

Rings and more especially fused rings can be tricky. For these systems, especially, we suggest the Cadillac of internal coordinates, the "natural internal coordinates" of Peter Pulay. For a description of these, see

P.Pulay, G.Fogarosi, F.Pang, J.E.Boggs, J.Am.Chem.Soc. 101, 2550-2560 (1979 ).  
G.Fogarasi, X.Zhou, P.W.Taylor, P.Pulay J.Am.Chem.Soc. 114, 8191-8201 (1992 ).

These are linear combinations of local coordinates, except in the case of rings. The examples given in these two papers are very thorough.

An illustration of these types of coordinates is given in the example job EXAM25.INP, distributed with GAMESS. This is a nonsense molecule, designed to show many kinds of functional groups. It is defined using standard bond distances with a classical Z-matrix input, and the angles in the ring are adjusted so that the starting value of the unclosed OO bond is also a standard value.

Using Cartesian coordinates is easiest, but takes a very large number of steps to converge. This however, is better than using the classical Z-matrix internals given in \$DATA, which is accomplished by setting NZVAR to the correct  $3N-6$  value. The geometry search changes the OO bond length to a very short value on the 1st step, and the SCF fails to converge. (Note that if you have used dummy atoms in the \$DATA input, you cannot simply enter NZVAR to optimize in internal coordinates, instead you must give a \$ZMAT which involves only real atoms).

The third choice of internal coordinates is the best set which can be made from the simple coordinates. It follows the advice given above for five membered rings, and because it includes the OO bond, has no trouble with crashing this bond. It takes 20 steps to converge, so the trouble of generating this \$ZMAT is certainly worth it compared to the use of Cartesians.

Natural internal coordinates are defined in the final group of input. The coordinates are set up first for the ring, including two linear combinations of all angles and all torsions within the ring. After this the methyl is hooked to the ring as if it were a NH group, using the usual terminal methyl hydrogen definitions. The H is hooked to this same ring carbon as if it were a methine. The NH and the CH2 within the ring follow Pulay's rules exactly. The amount of input is much greater than a normal Z-matrix. For example, 46 internal coordinates are given, which are then placed in  $3N-6=33$  linear combinations. Note that natural internals tend to be rich in bends, and short on torsions.

The energy results for the three coordinate systems which converge are as follows:

NSERCH	Cart	good Z-mat	nat. int.
--------	------	------------	-----------

0	-48.6594935049	-48.6594935049	-48.6594935049
1	-48.6800538676	-48.6806631261	-48.6838361406
2	-48.6822702585	-48.6510215698	-48.6874045449
3	-48.6858299354	-48.6882945647	-48.6932811528
4	-48.6881499412	-48.6849667085	-48.6946836332
5	-48.6890226067	-48.6911899936	-48.6959800274
6	-48.6898261650	-48.6878047907	-48.6973821465
7	-48.6901936624	-48.6930608185	-48.6987652146
8	-48.6905304889	-48.6940607117	-48.6996366016
9	-48.6908626791	-48.6949137185	-48.7006656309
10	-48.6914279465	-48.6963767038	-48.7017273728
11	-48.6921521142	-48.6986608776	-48.7021504975
12	-48.6931136707	-48.7007305310	-48.7022405019
13	-48.6940437619	-48.7016095285	-48.7022548935
14	-48.6949546487	-48.7021531692	-48.7022569328
15	-48.6961698826	-48.7022080183	-48.7022570260
16	-48.6973813002	-48.7022454522	-48.7022570662
17	-48.6984850655	-48.7022492840	
18	-48.6991553826	-48.7022503853	
19	-48.6996239136	-48.7022507037	
20	-48.7002269303	-48.7022508393	
21	-48.7005379631		
22	-48.7008387759		
	...		
50	-48.7022519950		

from which you can see that the natural internals are actually the best set. The \$ZMAT exhibits upward burps in the energy at step 2, 4, and 6, so that for the same number of steps, these coordinates are always at a higher energy than the natural internals.

The initial hessian generated for these three columns contains 0, 33, and 46 force constants. This assists the natural internals, but is not the major reason for its superior performance. The computed hessian at the final geometry of this molecule, when transformed into the natural internal coordinates is almost diagonal. This almost complete uncoupling of coordinates is what makes the natural internals perform so well. The conclusion is of course that not all coordinate systems are equal, and natural internals are the best. As another example, we have run the ATCHCP molecule, which is a popular geometry optimization test, due to its two fused rings:

H.B.Schlegel, Int.J.Quantum Chem., Symp. 26, 253-264(1992 )  
 T.H.Fischer and J.Almlof, J.Phys.Chem. 96, 9768-9774(1992 )  
 J.Baker, J.Comput.Chem. 14, 1085-1100(1993 )

Here we have compared the same coordinate types, using a guess hessian, or a computed hessian. The latter set of runs is a test of the coordinates only, as the initial hessian information is identical. The results show clearly the superiority of the natural internals, which like the previous example, give an energy decrease on every step:

	HESS=GUESS	HESS=READ
Cartesians	65	41 steps
good Z-matrix	32	23
natural internals	24	13

A final example is phosphinoazasilatrane, with three rings fused on a common SiN bond, in which 112 steps in Cartesian space became 32 steps in natural internals. The moral is:

"A little brain time can save a lot of CPU time."

In late 1998, a new kind of internal coordinate method was included into GAMESS. This is the delocalized internal coordinate (DLC) of

J.Baker, A. Kessi, B.Delley J.Chem.Phys. 105, 192-212(1996 )

although as is the usual case, the implementation is not exactly the same. Bonds are kept as independent coordinates, while angles are placed in linear combination by the DLC process. There are some interesting options for applying constraints, and other options to assist the automatic DLC generation code by either adding or deleting coordinates. It is simple to use DLCs in their most basic form:

```
$contrl nzvar=xx $end
```

```
$zmat dlc=.true. auto=.true. $end
```

Our initial experience is that the quality of DLCs is not as good as explicitly constructed natural internals, which benefit from human chemical knowledge, but are almost always better than carefully crafted \$ZMATs using only the primitive internal coordinates (although we have seen a few exceptions). Once we have more numerical experience with the use of DLC's, we will come back and revise the above discussion of coordinate choices. In the meantime, they are quite simple to choose, so give them a go.

### \* \* \* The Role of Symmetry \* \* \*

At the end of a successful geometry search, you will have a set of coordinates where the gradient of the energy is zero. However your newly discovered stationary point is not necessarily a minimum or saddle point!

This apparent mystery is due to the fact that the gradient vector transforms under the totally symmetric representation of the molecular point group. As a direct consequence, a geometry search is point group conserving. (For a proof of these statements, see J.W.McIver and A.Komornicki, Chem.Phys.Lett., 10,303-306(1971 )). In simpler terms, the molecule will remain in whatever point group you select in \$DATA, even if the true minimum is in some lower point group. Since a geometry search only explores totally symmetric degrees of freedom, the only way to learn about the curvatures for all degrees of freedom is RUNTYP=HESSIAN.

As an example, consider disilene, the silicon analog of ethene. It is natural to assume that this molecule is planar like ethene, and an OPTIMIZE run in  $D_{2h}$  symmetry will readily locate a stationary point. However, as a calculation of the hessian will readily show, this structure is a transition state (one imaginary frequency), and the molecule is really trans-bent ( $C_{2h}$ ). A careful worker will always characterize a stationary point as either a minimum, a transition state, or some higher order stationary point (which is not of great interest!) by performing a RUNTYP=HESSIAN.

The point group conserving properties of a geometry search can be annoying, as in the preceding example, or advantageous. For example, assume you wish to locate the transition state for rotation about the double bond in ethene. A little thought will soon reveal that ethene is  $D_{2h}$ , the 90 degrees twisted structure is  $D_{2d}$ , and structures in between are  $D_2$ . Since the saddle point is actually higher symmetry than the rest of the rotational surface, you can locate it by RUNTYP=OPTIMIZE within  $D_{2d}$  symmetry. You can readily find this stationary point with the diagonal guess hessian! In fact, if you attempt to do a RUNTYP=SADPOINT within  $D_{2d}$  symmetry, there will be no totally symmetric modes with negative curvatures, and it is unlikely that the geometry search will be very well behaved.

Although a geometry search cannot lower the symmetry, the gain of symmetry is quite possible. For example, if you initiate a water molecule optimization with a trial structure which has unequal bond lengths, the geometry search will come to a structure that is indeed  $C_{2v}$  (to within OPTTOL, anyway). However, GAMESS leaves it up to you to realize that a gain of symmetry has occurred.

In general, Mother Nature usually chooses more symmetrical structures over less symmetrical structures. Therefore you are probably better served to assume the higher symmetry, perform the geometry search, and then check the stationary point's curvatures. The alternative is to start with artificially lower symmetry and see if your system regains higher symmetry. The problem with this approach is that you don't necessarily know which subgroup is appropriate, and you lose the great speedups GAMESS can obtain from proper use of symmetry. It is good to note here that "lower symmetry" does not mean simply changing the name of the point group and entering more atoms in \$DATA, instead the nuclear coordinates themselves must actually be of lower symmetry.

### \* \* \* Practical Matters \* \* \*

Geometry searches do not bring the gradient exactly to zero. Instead they stop when the largest component of the gradient is below the value of OPTTOL, which defaults to a reasonable 0.0001. Analytic Hessians usually have residual frequencies below  $10\text{ cm}^{-1}$  with this degree of optimization. The sloppiest value you probably ever want to try is 0.0005.

If a geometry search runs out of time, or exceeds NSTEP, it can be restarted. For RUNTYP=OPTIMIZE, restart with the coordinates having the lowest total energy (do a string search on "FINAL"). For RUNTYP=SADPOINT, restart with the coordinates having the smallest gradient (do a string search on "RMS", which means root mean square). These are not necessarily at the last geometry!

The "restart" should actually be a normal run, that is you should not try to use the restart options in \$CONTRL (which may not work anyway). A geometry search can be restarted by extracting the desired coordinates for \$DATA from the printout, and by extracting the corresponding \$GRAD group from the PUNCH file. If the \$GRAD group is supplied, the program is able to save the time it would ordinarily take to compute the wavefunction and gradient at the initial point, which can be a substantial savings. There is no input to trigger reading of a \$GRAD group: if found, it is read and used. Be careful that your \$GRAD group actually corresponds to the coordinates in \$DATA, as GAMESS has no check for this.

Sometimes when you are fairly close to the minimum, an OPTIMIZE run will take a first step which raises the energy, with subsequent steps improving the energy and perhaps finding the minimum. The erratic first step is caused by the GUESS hessian. It may help to limit the size of this wrong first step, by reducing its radius, DXMAX. Conversely, if you are far from the minimum, sometimes you can decrease the number of steps by increasing DXMAX.

When using internals, the program uses an iterative process to convert the internal coordinate change into Cartesian space. In some cases, a small change in the internals will produce a large change in Cartesians, and thus produce a warning message on the output. If these warnings appear only in the beginning, there is probably no problem, but if they persist you can probably devise a better set of coordinates. You may in fact have one of the two problems described in the next paragraph. In some cases (hopefully very few) the iterations to find the Cartesian displacement may not converge, producing a second kind of warning message. The fix for this may very well be a new set of internal coordinates as well, or adjustment of ITBMAT in \$STATPT.

There are two examples of poorly behaved internal coordinates which can give serious problems. The first of these is three angles around a central atom, when this atom becomes planar (sum of the angles nears 360). The other is a dihedral where three of the atoms are nearly linear, causing the dihedral to flip between 0 and 180. Avoid these two situations if you want your geometry search to be convergent.

Sometimes it is handy to constrain the geometry search by freezing one or more coordinates, via the IFREEZ array. For example, constrained optimizations may be useful while trying to determine what area of a potential energy surface contains a saddle point. If you try to freeze coordinates with an automatically generated \$ZMAT, you need to know that the order of the coordinates defined in \$DATA is

```

y
y x r1
y x r2 x a3
y x r4 x a5 x w6
y x r7 x a8 x w9

```

and so on, where y and x are whatever atoms and molecular connectivity you happen to be using.

### \* \* \* Saddle Points \* \* \*

Finding minima is relatively easy. There are large tables of bond lengths and angles, so guessing starting geometries is pretty straightforward. Very nasty cases may require computation of an exact hessian, but the location of most minima is straightforward.

In contrast, finding saddle points is a black art. The diagonal guess hessian will never work, so you must provide a computed one. The hessian should be computed at your best guess as to what the transition state (T.S.) should be. It is safer to do this in two steps as outlined above, rather than HESS=CALC. This lets you verify you have guessed a structure with one and only one negative curvature. Guessing a good trial structure is the hardest part of a RUNTYP=SADPOINT!

This point is worth iterating. Even with sophisticated step size control such as is offered by the QA/TRIM or RFO methods, it is in general very difficult to move correctly from a region with incorrect curvatures towards a saddle point. Even procedures such as CONOPT or RUNTYP=GRADEXTR will not replace your own chemical intuition about where saddle points may be located.

The RUNTYP=HESSIAN's normal coordinate analysis is rigorously valid only at stationary points on the surface. This means the frequencies from the hessian at your trial geometry are untrustworthy, in particular the six "zero" frequencies corresponding to translational and rotational (T&R) degrees of freedom will usually be 300-500 cm<sup>-1</sup>, and possibly imaginary. The Sayvetz conditions on the printout will help you distinguish the T&R "contaminants" from the real vibrational modes. If you have defined a \$ZMAT, the PURIFY option within \$STATPT will help zap out these T&R contaminants).

If the hessian at your assumed geometry does not have one and only one imaginary frequency (taking into account that the "zero" frequencies can sometimes be 300i!), then it will probably be difficult to find the saddle point. Instead you need to compute a hessian at a better guess for the initial geometry, or read about mode following below.

If you need to restart your run, do so with the coordinates which have the smallest RMS gradient. Note that the energy does not necessarily have to decrease in a SADPOINT run, in contrast to an OPTIMIZE run. It is often necessary to do several restarts, involving

recomputation of the hessian, before actually locating the saddle point.

Assuming you do find the T.S., it is always a good idea to recompute the hessian at this structure. As

described in the discussion of symmetry, only totally symmetric vibrational modes are probed in a geometry search. Thus it is fairly common to find that at your "T.S." there is a second imaginary frequency, which corresponds to a non-totally symmetric vibration. This means you haven't found the correct T.S., and are back to the drawing board. The proper procedure is to lower the point group symmetry by distorting along the symmetry breaking "extra" imaginary mode, by a reasonable amount. Don't be overly timid in the amount of distortion, or the next run will come back to the invalid structure.

The real trick here is to find a good guess for the transition structure. The closer you are, the better. It is often difficult to guess these structures. One way around this is to compute a linear least motion (LLM) path. This connects the reactant structure to the product structure by linearly varying each coordinate. If you generate about ten structures intermediate to reactants and products, and compute the energy at each point, you will in general find that the energy first goes up, and then down. The maximum energy structure is a "good" guess for the true T.S. structure. Actually, the success of this method depends on how curved the reaction path is.

A particularly good paper on the symmetry which a saddle point (and reaction path) can possess is by

P.Pechukas, J.Chem.Phys. 64, 1516-1521 (1976 )

#### \* \* \* Mode Following \* \* \*

In certain circumstances, METHOD=RFO and QA can walk from a region of all positive curvatures (i.e. near a minimum) to a transition state. The criteria for whether this will work is that the mode being followed should be only weakly coupled to other close-lying Hessian modes. Especially, the coupling to lower modes should be almost zero. In practice this means that the mode being followed should be the lowest of a given symmetry, or spatially far away from lower modes (for example, rotation of methyl groups at different ends of the molecule). It is certainly possible to follow also modes which do not obey these criteria, but the resulting walk (and possibly TS location) will be extremely sensitive to small details such as the stepsize.

This sensitivity also explain why TS searches often fail, even when starting in a region where the Hessian has the required one negative eigenvalue. If the TS mode is strongly coupled to other modes, the direction of the mode is incorrect, and the maximization of the energy along that direction is not really what you want (but what you get).

Mode following is really not a substitute for the ability to intuit regions of the PES with a single local negative curvature. When you start near a minimum, it matters a great deal which side of the minima you start from, as the direction of the search depends on the sign of the gradient. We strongly urge that you read before trying to use IFOLOW, namely the papers by Frank Jensen and Jon Baker mentioned above, and see also Figure 3 of

C.J.Tsai, K.D.Jordan, J.Phys.Chem. 97, 11227-11237 (1993 )  
which is quite illuminating on the sensitivity of mode following to the initial geometry point.

Note that GAMESS retains all degrees of freedom in its hessian, and thus there is no reason to suppose the lowest mode is totally symmetric. Remember to lower the symmetry in the input deck if you want to follow non-symmetric modes. You can get a printout of the modes in internal

coordinate space by a EXETYP=CHECK run, which will help you decide on the value of IFOLOW.

\* \* \*

CONOPT is a different sort of saddle point search procedure. Here a certain "CONstrained OPTimization" may be considered as another mode following method. The idea is to start from a minimum, and then perform a series of optimizations on hyperspheres of increasingly larger radii. The initial step is taken along one of the Hessian modes, chosen by IFOLOW, and the geometry is optimized subject to the constraint that the distance to the minimum is constant. The convergence criteria for the gradient norm perpendicular to the constraint is taken as  $10 \times \text{OPTTOL}$ , and the corresponding steplength as  $100 \times \text{OPTTOL}$ .

After such a hypersphere optimization has converged, a step is taken along the line connecting the two previous optimized points to get an estimate of the next hypersphere geometry. The stepsize is DXMAX, and the radius of hyperspheres is thus increased by an amount close (but not equal) to DXMAX. Once the pure NR step size falls below  $\text{DXMAX}/2$  or 0.10 (whichever is the largest) the algorithm switches to a straight NR iterate to (hopefully) converge on the stationary point.

The current implementation always conducts the search in cartesian coordinates, but internal coordinates may be printed by the usual specification of NZVAR and ZMAT. At present there is no restart option programmed.

CONOPT is based on the following papers, but the actual implementation is the modified equations presented in Frank Jensen's paper mentioned above.

Y. Abashkin, N. Russo, J.Chem.Phys. 100, 4477-4483(1994).

Y. Abashkin, N. Russo, M. Toscano, Int.J.Quant.Chem. 52, 695-704(1994).

There is little experience on how this method works in practice, experiment with it at your own risk!

### IRC methods

The Intrinsic Reaction Coordinate (IRC) is defined as the minimum energy path connecting the reactants to products via the transition state. In practice, the IRC is found by first locating the transition state for the reaction. The IRC is then found in halves, going forward and backwards from the saddle point, down the steepest descent path in mass weighted Cartesian coordinates. This is accomplished by numerical integration of the IRC equations, by a variety of methods to be described below.

The IRC is becoming an important part of polyatomic dynamics research, as it is hoped that only knowledge of the PES in the vicinity of the IRC is needed for prediction of reaction rates, at least at threshold energies. The IRC has a number of uses for electronic structure purposes as well. These include the proof that a certain transition structure does indeed connect a particular set of reactants and products, as the structure and imaginary frequency normal mode at the saddle point do not always unambiguously identify the reactants and products. The study of the electronic and geometric structure along the IRC is also of interest. For example, one can obtain localized orbitals along the path to determine when bonds break or form.

The accuracy to which the IRC is determined is dictated by the use one intends for it. Dynamical calculations require a very accurate determination of the path, as derivative information (second derivatives of the PES at various IRC points, and path curvature) is required later. Thus, a sophisticated integration method (such as AMPC4 or RK4), and small

step sizes (STRIDE=0.05, 0.01, or even smaller) may be needed. In addition to this, care should be taken to locate the transition state carefully (perhaps decreasing OPTTOL by a factor of 10), and in the initiation of the IRC run. The latter might require a hessian matrix obtained by double differencing, certainly the hessian should be PURIFY'd. Note also that EVIB must be chosen carefully, as described below.

On the other hand, identification of reactants and products allows for much larger step sizes, and cruder integration methods. In this type of IRC one might want to be careful in leaving the saddle point (perhaps STRIDE should be reduced to 0.10 or 0.05 for the first few steps away from the transition state), but once a few points have been taken, larger step sizes can be employed. In general, the defaults in the \$IRC group are set up for this latter, cruder quality IRC. The STRIDE value for the GS2 method can usually be safely larger than for other methods, no matter what your interest in accuracy is.

The simplest method of determining an IRC is linear gradient following, PACE=LINEAR. This method is also known as Euler's method. If you are employing PACE=LINEAR, you can select "stabilization" of the reaction path by the Ishida, Morokuma, Komornicki method. This type of corrector has no apparent mathematical basis, but works rather well since the bisector usually intersects the reaction path at right angles (for small step sizes). The ELBOW variable allows for a method intermediate to LINEAR and stabilized LINEAR, in that the stabilization will be skipped if the gradients at the original IRC point, and at the result of a linear prediction step form an angle greater than ELBOW. Set ELBOW=180 to always perform the stabilization.

A closely related method is PACE=QUAD, which fits a quadratic polynomial to the gradient at the current and immediately previous IRC point to predict the next point. This pace has the same computational requirement as LINEAR, and is slightly more accurate due to the reuse of the old gradient. However, stabilization is not possible for this pace, thus a stabilized LINEAR path is usually more accurate than QUAD.

Two rather more sophisticated methods for integrating the IRC equations are the fourth order Adams-Moulton predictor-corrector (PACE=AMPC4) and fourth order Runge-Kutta (PACE=RK4). AMPC4 takes a step towards the next IRC point (prediction), and based on the gradient found at this point (in the near vicinity of the next IRC point) obtains a modified step to the desired IRC point (correction). AMPC4 uses variable step sizes, based on the input STRIDE. RK4 takes several steps part way toward the next IRC point, and uses the gradient at these points to predict the next IRC point. RK4 is the most accurate integration method implemented in GAMESS, and is also the most time consuming.

The Gonzalez-Schlegel 2nd order method finds the next IRC point by a constrained optimization on the surface of a hypersphere, centered at 1/2 STRIDE along the gradient vector leading from the previous IRC point. By construction, the reaction path between two successive IRC points is thus a circle tangent to the two gradient vectors. The algorithm is much more robust for large steps than the other methods, so it has been chosen as the default method. Thus, the default for STRIDE is too large for the other methods. The number of energy and gradients need to find the next point varies with the difficulty of the constrained optimization, but is normally not very many points. Be sure to provide the updated hessian from the previous run when restarting PACE=GS2.

The number of wavefunction evaluations, and energy gradients needed to jump from one point on the IRC to the next point are summarized in the following table:

<u>PACE</u>	<u># energies</u>	<u># gradients</u>
LINEAR	1	1
stabilized		



LINEAR	3	2
QUAD	1	1 (+ reuse of historical gradient)
AMPC4	2	2 (see note)
RK4	4	4
GS2	2-4	2-4 (equal numbers)

Note that the AMPC4 method sometimes does more than one correction step, with each such correction adding one more energy and gradient to the calculation. You get what you pay for in IRC calculations: the more energies and gradients which are used, the more accurate the path found.

A description of these methods, as well as some others that were found to be not as good is given by Kim Baldrige and Lisa Pederson, *Pi Mu Epsilon Journal*, 9, 513-521 (1993).

\* \* \*

All methods are initiated by jumping from the saddle point, parallel to the normal mode (CMODE) which has an imaginary frequency. The jump taken is designed to lower the energy by an amount EVIB. The actual distance taken is thus a function of the imaginary frequency, as a smaller FREQ will produce a larger initial jump. You can simply provide a \$HESS group instead of CMODE and FREQ, which involves less typing. To find out the actual step taken for a given EVIB, use EXETYP=CHECK. The direction of the jump (towards reactants or products) is governed by FORWRD. Note that if you have decided to use small step sizes, you must employ a smaller EVIB to ensure a small first step. The GS2 method begins by following the normal mode by one half of STRIDE, and then performing a hypersphere minimization about that point, so EVIB is irrelevant to this PACE.

The only method which proves that a properly converged IRC has been obtained is to regenerate the IRC with a smaller step size, and check that the IRC is unchanged. Again, note that the care with which an IRC must be obtained is highly dependent on what use it is intended for.

A small program which converts the IRC results punched to file IRCDATA into a format close to that required by the POLYRATE VTST dynamics program written in Don Truhlar's group is available. For a copy of this IRCED program, contact Mike Schmidt. The POLYRATE program must be obtained from the Truhlar group.

Some key IRC references are:

- K.Ishida, K.Morokuma, A.Komornicki *J.Chem.Phys.* 66, 2153-2156 (1977)
- K.Muller *Angew.Chem., Int.Ed.Engl.* 19, 1-13 (1980)
- M.W.Schmidt, M.S.Gordon, M.Dupuis *J.Am.Chem.Soc.* 107, 2585-2589 (1985)
- B.C.Garrett, M.J.Redmon, R.Steckler, D.G.Truhlar, K.K.Baldrige, D.Bartol,  
M.W.Schmidt, M.S.Gordon *J.Phys.Chem.* 92, 1476-1488(1988)
- K.K.Baldrige, M.S.Gordon, R.Steckler, D.G.Truhlar *J.Phys.Chem.* 93, 5107-  
5119(1989)
- C.Gonzales, H.B.Schlegel  
*J.Chem.Phys.* 90, 2154-2161(1989)

The IRC discussion closes with some practical tips:

The \$IRC group has a confusing array of variables, but fortunately very little thought need be given to most of them. An IRC run is restarted by moving the coordinates of the next predicted IRC point into \$DATA, and inserting the new \$IRC group into your input file. You must select the desired value for NPOINT. Thus, only the first job which initiates the IRC requires much thought about \$IRC.

The symmetry specified in the \$DATA deck should be the symmetry of the reaction path. If a saddle point happens to have higher symmetry, use only the lower symmetry in the \$DATA deck when initiating the IRC. The reaction path will have a lower symmetry than the saddle point whenever the normal mode with imaginary frequency is not totally symmetric. Be careful that the order and orientation of the atoms corresponds to that used in the run which generated the hessian matrix.

If you wish to follow an IRC for a different isotope, use the \$MASS group. If you wish to follow the IRC in regular Cartesian coordinates, just enter unit masses for each atom. Note that CMODE and FREQ are a function of the atomic masses, so either regenerate FREQ and CMODE, or more simply, provide the correct \$HESS group.

### Gradient Extremals

This section of the manual, as well as the source code to trace gradient extremals was written by Frank Jensen of Odense University.

A Gradient Extremal (GE) curve consists of points where the gradient norm on a constant energy surface is stationary. This is equivalent to the condition that the gradient is an eigenvector of the Hessian. Such GE curves radiate along all normal modes from a stationary point, and the GE leaving along the lowest normal mode from a minimum is the gentlest ascent curve. This is not the same as the IRC curve connecting a minimum and a TS, but may in some cases be close.

GEs may be divided into three groups: those leading to dissociation, those leading to atoms colliding, and those which connect stationary points. The latter class allows a determination of many (all?) stationary points on a PES by tracing out all the GEs. Following GEs is thus a semi-systematic way of mapping out stationary points. The disadvantages are:

- i) There are many (but finitely many!) GEs for a large molecule.
- ii) Following GEs is computationally expensive.
- lii) There is no control over what type of stationary point (if any) a GE will lead to.

Normally one is only interested in minima and TSs, but many higher order saddle points will also be found. Furthermore, it appears that it is necessary to follow GEs radiating also from TSs and second (and possibly also higher) order saddle point to find all the TSs.

A rather complete map of the extremals for the H<sub>2</sub>CO potential surface is available in a paper which explains the points just raised in greater detail:

K.Bondensgaard, F.Jensen, J.Chem.Phys. 104, 8025-8031(1996 ).

An earlier paper gives some of the properties of GEs:

D.K.Hoffman, R.S.Nord, K.Ruedenberg, Theor. Chim. Acta 69, 265-279(1986 ).

There are two GE algorithms in GAMESS, one due to Sun and Ruedenberg (METHOD=SR), which has been extended to include the capability of locating bifurcation points and turning points, and another due to Jorgensen, Jensen, and Helgaker (METHOD=JJH):

J. Sun, K. Ruedenberg, J.Chem.Phys. 98, 9707-9714(1993 )

P. Jorgensen, H. J. Aa. Jensen, T. Helgaker Theor. Chim. Acta 73, 55 (1988 ).

The Sun and Ruedenberg method consist of a predictor step taken along the tangent to the GE curve, followed by one or more corrector steps to bring the geometry back to the GE. Construction of the GE tangent and the corrector step requires elements of the third derivative of the energy, which is obtained by a numerical differentiation of two Hessians. This puts some

limitations on which systems the GE algorithm can be used for. First, the numerical differentiation of the Hessian to produce third derivatives means that the Hessian should be calculated by analytical methods, thus only those types of wavefunctions where this is possible can be used. Second, each predictor/corrector step requires at least two Hessians, but often more. Maybe 20-50 such steps are necessary for tracing a GE from one stationary point to the next. A systematic study of all the GE radiating from a stationary point increases the work by a factor of  $\sim 2 \cdot (3N-6)$ . One should thus be prepared to invest at least hundreds, and more likely thousands, of Hessian calculations. In other words, small systems, small basis sets, and simple wavefunctions.

The Jorgensen, Jensen, and Helgaker method consists of taking a step in the direction of the chosen Hessian eigenvector, and then a pure NR step in the perpendicular modes. This requires (only) one Hessian calculation for each step. It is not suitable for following GEs where the GE tangent forms a large angle with the gradient, and it is incapable of locating GE bifurcations.

Although experience is limited at present, the JJH method does not appear to be suitable for following GEs in general (at least not in the current implementation). Experiment with it at your own risk!

The flow of the SR algorithm is as follows: A predictor geometry is produced, either by jumping away from a stationary point, or from a step in the tangent direction from the previous point on the GE. At the predictor geometry, we need the gradient, the Hessian, and the third derivative in the gradient direction. Depending on HSDFDB, this can be done in two ways. If .TRUE. The gradient is calculated, and two Hessians are calculated at SNUMH distance to each side in the gradient direction. The Hessian at the geometry is formed as the average of the two displaced Hessians. This corresponds to a double-sided differentiation, and is the numerical most stable method for getting the partial third derivative matrix. If HSDFDB = .FALSE., the gradient and Hessian are calculated at the current geometry, and one additional Hessian is calculated at SNUMH distance in the gradient direction. This corresponds to a single-sided differentiation. In both cases, two full Hessian calculations are necessary, but HSDFDB = .TRUE. require one additional wavefunction and gradient calculation. This is usually a fairly small price compared to two Hessians, and the numerically better double-sided differentiation has therefore been made the default.

Once the gradient, Hessian, and third derivative is available, the corrector step and the new GE tangent are constructed. If the corrector step is below a threshold, a new predictor step is taken along the tangent vector. If the corrector step is larger than the threshold, the correction step is taken, and a new micro iteration is performed. DELCOR thus determines how closely the GE will be followed, and DPRED determine how closely the GE path will be sampled.

The construction of the GE tangent and corrector step involve solution of a set of linear equations, which in matrix notation can be written as  $Ax=B$ . The A-matrix is also the second derivative of the gradient norm on the constant energy surface.

After each corrector step, various things are printed to monitor the behavior: The projection of the gradient along the Hessian eigenvalues (the gradient is parallel to an eigenvector on the GE), the projection of the GE tangent along the Hessian eigenvectors, and the overlap of the Hessian eigenvectors with the mode being followed from the previous (optimized) geometry. The sign of these overlaps are not significant, they just refer to an arbitrary phase of the Hessian eigenvectors.

After the micro iterations has converged, the Hessian eigenvector curvatures are also displayed, this is an indication of the coupling between the normal modes. The number of negative eigenvalues in the A-matrix is denoted the GE index. If it changes, one of the

eigenvalues must have passed through zero. Such points may either be GE bifurcations (where two GEs cross) or may just be "turning points", normally when the GE switches from going uphill in energy to downhill, or vice versa. The distinction is made based on the B-element corresponding to the A-matrix eigenvalue = 0. If the B-element = 0, it is a bifurcation, otherwise it is a turning point.

If the GE index changes, a linear interpolation is performed between the last two points to locate the point where the A-matrix is singular, and the corresponding B-element is determined. The linear interpolation points will in general be off the GE, and thus the evaluation of whether the B-element is 0 is not always easy. The program additionally evaluates the two limiting vectors which are solutions to the linear sets of equations, these are also used for testing whether the singular point is a bifurcation point or turning point.

Very close to a GE bifurcation, the corrector step become numerically unstable, but this is rarely a problem in practice. It is a priori expected that GE bifurcation will occur only in symmetric systems, and the crossing GE will break the symmetry. Equivalently, a crossing GE may be encountered when a symmetry element is formed, however such crossings are much harder to detect since the GE index does not change, as one of the A-matrix eigenvalues merely touches zero. The program prints a message if the absolute value of an A-matrix eigenvalue reaches a minimum near zero, as such points may indicate the passage of a bifurcation where a higher symmetry GE crosses. Run a movie of the geometries to see if a more symmetric structure is passed during the run.

An estimate of the possible crossing GE direction is made at all points where the A-matrix is singular, and two perturbed geometries in the + and - direction are written out. These may be used as predictor geometries for following a crossing GE. If the singular geometry is a turning point, the + and - geometries are just predictor geometries on the GE being followed.

In any case, a new predictor step can be taken to trace a different GE from the newly discovered singular point, using the direction determined by interpolation from the two end point tangents (the GE tangent cannot be uniquely determined at a bifurcation point). It is not possible to determine what the sign of IFOLLOW should be when starting off along a crossing GE at a bifurcation, one will have to try a step to see if it returns to the bifurcation point or not.

In order to determine whether the GE index change it is necessary to keep track of the order of the A-matrix eigenvalues. The overlap between successive eigenvectors are shown as "Alpha mode overlaps".

Things to watch out for:

- 1) The numerical differentiation to get third derivatives requires more accuracy than usual. The SCF convergence should be at least 100 times smaller than SNUMH, and preferably better. With the default SNUMH of  $10^{**}(-4)$  the SCF convergence should be at least  $10^{**}(-6)$ . Since the last few SCF cycles are inexpensive, it is a good idea to tighten the SCF convergence as much as possible, to maybe  $10^{**}(-8)$  or better. You may also want to increase the integral accuracy by reducing the cutoffs (ITOL and ICUT) and possibly also try more accurate integrals (INTTYP=HONDO). The CUTOFF in \$TRNSFM may also be reduced to produce more accurate Hessians. Don't attempt to use a value for SNUMH below  $10^{**}(-6)$ , as you simply can't get enough accuracy. Since experience is limited at present, it is recommended that some tests runs are made to learn the sensitivity of these factors for your system.
- 2) GEs can be followed in both directions, uphill or downhill. When starting from a stationary point, the direction is implicitly given as away from the stationary point. When starting

from a non-stationary point, the "+" and "-" directions (as chosen by the sign of IFOLOW) refers to the gradient direction. The "+" direction is along the gradient (energy increases) and "-" is opposite to the gradient (energy decreases).

- 3) A switch from one GE to another may be seen when two GE come close together. This is especially troublesome near bifurcation points where two GEs actually cross. In such cases a switch to a GE with -higher- symmetry may occur without any indication that this has happened, except possibly that a very large GE curvature suddenly shows up. Avoid running the calculation with less symmetry than the system actually has, as this increases the likelihood that such switches occurring. Fix: alter DPRED to avoid having the predictor step close to the crossing GE.
- 4) "Off track" error message: The Hessian eigenvector which is parallel to the gradient is not the same as the one with the largest overlap to the previous Hessian mode. This usually indicate that a GE switch has occurred (note that a switch may occur without this error message), or a wrong value for IFOLOW when starting from a non-stationary point. Fix: check IFOLOW, if it is correct then reduce DPRED, and possibly also DELCOR.
- 5) Low overlaps of A-matrix eigenvectors. Small overlaps may give wrong assignment, and wrong conclusions about GE index change. Fix: reduce DPRED.
- 6) The interpolation for locating a point where one of the A-matrix eigenvalues is zero fail to converge. Fix: reduce DPRED (and possibly also DELCOR) to get a shorter (and better) interpolation line.
- 7) The GE index changes by more than 1. A GE switch may have occurred, or more than one GE index change is located between the last and current point. Fix: reduce DPRED to sample the GE path more closely.
- 8) If SNRMAX is too large the algorithm may try to locate stationary points which are not actually on the GE being followed. Since GEs often pass quite near a stationary point, SNRMAX should only be increased above the default 0.10 after some consideration.

## Continuum solvation methods

In a very thorough 1994 review of continuum solvation models, Tomasi and Persico divide the possible approaches to the treatment of solvent effects into four categories:

- a) virial equations of state, correlation functions
- b) Monte Carlo or molecular dynamics simulations
- c) continuum treatments
- d) molecular treatments

The Effective Fragment Potential method, documented in the following section of this chapter, falls into the latter category, as each EFP solvent molecule is modeled as a distinct object. This section describes the three continuum models which are implemented in the standard version of GAMESS, and a fourth model which can be obtained and easily interfaced.

Continuum models typically form a cavity of some sort containing the solute molecule, while the solvent outside the cavity is thought of as a continuous medium and is categorized by a limited amount of physical data, such as the dielectric constant. The electric field of the charged particles comprising the solute interact with this background medium, producing a polarization in it, which in turn feeds back upon the solute's wavefunction.

\* \* \*

A simple continuum model is the Onsager cavity model, often called the Self-Consistent Reaction Field, or SCRF model. This represents the charge distribution of the solute in terms of a multipole expansion. SCRF usually uses an idealized cavity (spherical or ellipsoidal) to allow an analytic solution to the interaction energy between the solute multipole and the multipole which this induces in the continuum. This method is implemented in GAMESS in the simplest possible fashion:

- i) a spherical cavity is used
- ii) the molecular electrostatic potential of the solute is represented as a dipole only, except a monopole is also included for an ionic solute.

The input for this implementation of the Kirkwood-Onsager model is provided in \$SCRF.

Some references on the SCRF method are

1. J.G.Kirkwood J.Chem.Phys. 2, 351 (1934 )
2. L.Onsager J.Am.Chem.Soc. 58, 1486 (1936 )
3. O.Tapia, O.Goscinski Mol.Phys. 29, 1653 (1975 )
4. M.M.Karelson, A.R.Katritzky, M.C.Zerner Int.J.Quantum Chem., Symp. 20, 521-527 (1986 )
5. K.V.Mikkelsen, H.Agren, H.J.Aa.Jensen, T.Helgaker J.Chem.Phys. 89, 3086-3095 (1988 )
6. M.W.Wong, M.J.Frisch, K.B.Wiberg J.Am.Chem.Soc. 113, 4776-4782 (1991 )
7. M.Szafran, M.M.Karelson, A.R.Katritzky, J.Koput, M.C.Zerner J.Comput.Chem. 14, 371-377 (1993 )
8. M.Karelson, T.Tamm, M.C.Zerner J.Phys.Chem. 97, 11901-11907 (1993 )

The method is very sensitive to the choice of the solute RADIUS, but not very sensitive to the particular DIELEC of polar solvents. The plots in reference 7 illustrate these points very nicely. The SCRF implementation in GAMESS is Zerner's Method A, described in the same reference. The total solute energy includes the Born term, if the solute is an ion. Another limitation is that a solute's electrostatic potential is not likely to be fit well as a dipole moment only, for example see Table VI of reference 5 which illustrates the importance of higher multipoles. Finally, the restriction to a spherical cavity may not be very representative of the solute's true shape. However, in the special case of a roundish molecule, and a large dipole which is geometry sensitive, the SCRF model may include sufficient physics to be meaningful:

M.W.Schmidt, T.L.Windus, M.S.Gordon J.Am.Chem.Soc. 117, 7480-7486(1995 ).

\* \* \*

A much more sophisticated continuum method, named the Polarizable Continuum Model, is also available. The PCM method places a solute in a cavity formed by a union of spheres centered on each atom. PCM also includes a more exact treatment of the electrostatic interaction with the surrounding medium, as the electrostatic potential of the solute generates an 'apparent surface charge' on the cavity's surface. The computational procedure divides this surface into small tesserae, on which the charge (and contributions to the gradient) are evaluated. Typically the spheres defining the cavity are taken to be 1.2 times the van der Waals radii. A technical difficulty caused by the penetration of the solute charge density outside this cavity is dealt with by a renormalization. The solvent is characterized by its dielectric constant, surface tension, size, density, and so on. Procedures are provided not only for the computation of the electrostatic interaction of the solute with the apparent surface charges, but also for the cavitation energy, and dispersion and repulsion contributions to the solvation free energy.

The main input group is \$PCM, with \$PCMCAV providing auxiliary cavity information. If any of the optional energy computations are requested in \$PCM, the additional input groups

\$NEWCAV, \$DISBS, or \$DISREP may be required.

Solvation of course affects the non-linear optical properties of molecules. The PCM implementation extends RUNTYP=TDHF to include solvent effects. Both static and frequency dependent hyperpolarizabilities can be found. Besides the standard PCM electrostatic contribution, the IREP and IDP keywords can be used to determine the effects of repulsion and dispersion on the polarizabilities.

Due to its sophistication, users of the PCM model are strongly encouraged to read the primary literature:

General papers on the PCM method:

- 1) S.Miertus, E.Scrocco, J.Tomasi Chem.Phys. 55, 117-129(1981 )
- 2) J.Tomasi, M.Persico Chem.Rev. 94, 2027-2094(1994 )
- 3) R.Cammi, J.Tomasi J.Comput.Chem. 16, 1449-1458(1995 )

The GEPOL method for cavity construction:

- 4) J.L.Pascual-Ahuir, E.Silla, J.Tomasi, R.Bonaccorsi J.Comput.Chem. 8, 778-787(1987 )

Charge renormalization (see also ref. 3):

- 5) B.Mennucci, J.Tomasi J.Chem.Phys. 106, 5151-5158(1997 )

Derivatives with respect to nuclear coordinates: (energy gradient and hessian)

- 6) R.Cammi, J.Tomasi J.Chem.Phys. 100, 7495-7502(1994 )
- 7) R.Cammi, J.Tomasi J.Chem.Phys. 101, 3888-3897(1995 )
- 8) M.Cossi, B.Mennucci, R.Cammi J.Comput.Chem. 17, 57-73(1996 )

Derivatives with respect to applied electric fields:

(polarizabilities and hyperpolarizabilities)

- 9) R.Cammi, J.Tomasi Int.J.Quantum Chem. Symp. 29, 465-474(1995 )
- 10) R.Cammi, M.Cossi, J.Tomasi J.Chem.Phys. 104, 4611-4620(1996 )
- 11) R.Cammi, M.Cossi, B.Mennucci, J.Tomasi J.Chem.Phys. 105, 10556-10564(1996 )
- 12) B. Mennucci, C. Amovilli, J. Tomasi J.Chem.Phys. submitted.

Cavitation energy:

- 13) R.A.Pierotti Chem.Rev. 76, 717-726(1976 )
- 14) J.Langlet, P.Claverie, J.Caillet, A.Pullman J.Phys.Chem. 92, 1617-1631(1988 )

Dispersion and repulsion energies:

- 15) F.Floris, J.Tomasi J.Comput.Chem. 10, 616-627(1989 )
- 16) C.Amovilli, B.Mennucci J.Phys.Chem. B101, 1051-1057(1997 )

At the present time, the PCM model in GAMESS has the following limitations:

- a) SCFTYP=RHF and MCSCF, only.
- b) point group symmetry is switched off internally during PCM.
- c) The PCM model does not run in parallel.
- d) electric field integrals at normals to the surface elements are stored on disk, even during DIRSCF runs. The file size may be considerable.
- e) To minimize common block storage, the maximum number of spheres forming the cavity is 100, with an upper limit on the number of surface tesserae set to 2500. These may be raised by the 'mung' script listed in the Programming chapter.
- f) nuclear derivatives are limited to gradients, although theory for Hessians is given in

Ref. 7.

The calculation shown below illustrates the use of most PCM options. Since methane is non-polar, its internal energy change and the direct PCM electrostatic interaction is smaller than the cavitation, repulsion, and dispersion corrections. Note that the use of ICAV, IREP, and IDP are currently incompatible with gradients, so a reasonable calculation sequence might be to perform the geometry optimization with PCM electrostatics turned on, then perform an additional calculation to include the other solvent effects, adding extra functions to improve the dispersion correction.

```

! calculation of CH4 (metano) in PCM water.
! This input reproduces the data in Table 2, line 6, of
! C. Amovilli, B. Mennucci J. Phys. Chem. B101, 1051-7(1997)
!
! The gas phase FINAL energy is -40.2075980280
! The FINAL energy in PCM water= -40.2143590161
!
!
! FREE ENERGY IN SOLVENT      = -25234.89 KCAL/MOL
! INTERNAL ENERGY IN SOLVENT = -25230.64 KCAL/MOL
! DELTA INTERNAL ENERGY      =      .01 KCAL/MOL ( 0.0)
! ELECTROSTATIC INTERACTION    =     -.22 KCAL/MOL (-0.2)
! PIEROTTI CAVITATION ENERGY =      5.98 KCAL/MOL ( 6.0)
! DISPERSION FREE ENERGY      =     -6.00 KCAL/MOL (-6.0)
! REPULSION FREE ENERGY       =      1.98 KCAL/MOL ( 2.0)
! TOTAL INTERACTION            =      1.73 KCAL/MOL ( 1.8)
! TOTAL FREE ENERGY IN SOLVENT= -25228.91 KCAL/MOL
!
!
! $contrl scftyp=rhf runtyp=energy $end
! $guess guess=huckel $end
! $system memory=300000 $end
! the W1 basis input here exactly matches HONDO's DZP
! $DATA
! CH4...gas phase geometry...in PCM water
! Td
!
! Carbon      6.
!   DZV
!   D 1 ; 1 0.75 1.0
!
! Hydrogen    1.  0.6258579976  0.6258579976  0.6258579976
!   DZV 0 1.20 1.15 ! inner and outer scale factors
!   P 1 ; 1 1.00 1.0
!
! $END
! reference cited used value for H2O's solvent radius
! which differs from the built in constants.
! SPCM ICOMP=2 IREP=1 IDP=1 ICAV=1
!     SOLVNT=WATER RSOLV=1.35 $END
! SNEWCAV IPTYPE=2 ITSNUM=540 $END
! dispersion W2 basis uses exponents which are
! 1/3 of smallest exponent in W1 basis of $DATA.
! $DISBS NADD=11 NKTYP(1)=0, 1, 2, 0, 1, 0, 1, 0, 1, 0, 1
!     XYZE(1)=0.0, 0.0, 0.0, 0.0, 0.0511
!           0.0, 0.0, 0.0, 0.0, 0.0382

```



```

0. 0, 0. 0, 0. 0, 0. 25
1. 1817023, 1. 1817023, 1. 1817023, 0. 05435467
1. 1817023, 1. 1817023, 1. 1817023, 0. 33333333
-1. 1817023, 1. 1817023, -1. 1817023, 0. 05435467
-1. 1817023, 1. 1817023, -1. 1817023, 0. 33333333
1. 1817023, -1. 1817023, -1. 1817023, 0. 05435467
1. 1817023, -1. 1817023, -1. 1817023, 0. 33333333
-1. 1817023, -1. 1817023, 1. 1817023, 0. 05435467
-1. 1817023, -1. 1817023, 1. 1817023, 0. 33333333 $end

```

COSMO (conductor-like screening model) represents a different approach for carrying out polarized continuum calculations. The model was originally developed by Andreas Klamt, with extensions to ab initio computation in GAMESS by Kim Baldridge.

In the COSMO method, the surrounding medium is modeled as a conductor rather than as a dielectric in order to establish the initial boundary conditions. The assumption that the surrounding medium is well modelled as a conductor simplifies the electrostatic computations and corrections may be made a posteriori for dielectric behavior.

The current implementation of COSMO involves the computation of distributed multipoles up to hexadecapoles to represent the charge distribution of the molecule within the cavity. The multipole moments induce the formation of charges on the surface of the cavity that contains the molecule. These charges are then fed back into the SCF, and both the molecular wavefunction and the surface charges are iterated to self-consistency.

The original model of Klamt was introduced using a molecular shaped cavity which had open parts along the crevices of intersecting atomic spheres. While having considerable technical advantages, this approximation causes artifacts in the context of the more generalized theory, so the current method for cavity construction includes a closure of the cavity to eliminate crevices or pockets.

At present, the COSMO model accounts only for the electrostatic interactions between solvent and solute. Klamt has proposed a novel statistical scheme to compute the full solvation free energy for neutral solutes, which will be formulated for GAMESS by Baldridge et al.

The simplicity of the COSMO model allows computation of gradients, allowing optimization within the context of the solvent. The method is programmed for closed shell RHF energy and gradient, and the MP2 energy correction may be obtained.

Some references on the COSMO model are:

A.Klamt, G.Schuurman *J.Chem.Soc.Perkin Trans 2*, 799-805(1993)

A.Klamt *J.Phys.Chem.* 99, 2224-2235(1995)

K.Baldridge, A.Klamt *J.Chem.Phys.* 106, 6622-6633 (1997)

A final possible continuum treatment is the "solution model 5" approach. Ab initio SM5 is described in

J.Li, G.D.Hawkins, C.J.Cramer, D.G.Truhlar *Chem.Phys.Lett.*, 288, 293-298(1998)

SM5 represents the molecule's electrostatics as a set of atomic point charges. These are chosen by a procedure based on correcting Lowdin atomic charges according to a quadratic function of the computed Mayer bond orders, which is designed to better reproduce experimental dipole moments. These charges are called "charge model 2", and CM2 is described in

J.Li, T.Zhu, C.J.Cramer, D.G.Truhlar *J.Phys.Chem.A*, 102, 1820-1831(1998)

In addition to a self-consistent reaction field treatment of the CM2 electrostatics, SM5 includes

a term accounting for the following first solvation shell effects: cavity creation, dispersion, and changes in solvent structure, which are modeled by atomic surface tension parameters. It is possible to use this code simply to extract gas phase CM2 charges. The implementation is termed GAMESOL (one S), by

J.Li, G.D.Hawkins, D.A.Liotard, C.J.Cramer, D.G.Truhlar  
and is available at

<http://comp.chem.umn.edu/gamesol>

After signing a license not much more stringent than the license for GAMESS itself, you can obtain the new source code from the U. of Minnesota. The interface is not clean, as considerable code is inserted directly into RHFUHF and other GAMESS modules, so you must be very careful to obtain code that matches the dates on the top of your original GAMESS source files.

## The Effective Fragment Potential Method

The basic idea behind the effective fragment potential (EFP) method is to replace the chemically inert part of a system by EFPs, while performing a regular ab initio calculation on the chemically active part. Here "inert" means that no covalent bond breaking process occurs. This "spectator region" consists of one or more "fragments", which interact with the ab initio "active region" through non-bonded interactions, and so of course these EFP interactions affect the ab initio wavefunction. A simple example of an active region might be a solute molecule, with a surrounding spectator region of solvent molecules represented by fragments. Each discrete solvent molecule is represented by a single fragment potential, in marked contrast to continuum models for solvation.

The quantum mechanical part of the system is entered in the \$DATA group, along with an appropriate basis. The EFPs defining the fragments are input by means of a \$EFRAG group, one or more \$FRAGNAME groups describing each fragment's EFP, and a \$FRGRPL group. These groups define non-bonded interactions between the ab initio system and the fragments, and between the fragments. The former interactions enter via one-electron operators in the ab initio Hamiltonian, while the latter interactions are treated by analytic functions. The only electrons explicitly treated (e.g. with basis functions used to expand occupied orbitals) are those in the active region, so there are no new two electron terms. Thus the use of EFPs leads to significant time savings compared to full ab initio calculations on the same system.

At ISU, the EFPs are currently used to model RHF/DZP water molecules in order to study aqueous solvation effects, for example references 1,2,3. Our co-workers at NIST have also used EFPs to model parts of enzymes, see reference 4.

\*\*\* Terms in an EFP \*\*\*

The non-bonded interactions currently implemented are:

1) Coulomb interaction. The charge distribution of the fragments is represented by an arbitrary number of charges, dipoles, quadrupoles, and octupoles, which interact with the ab initio hamiltonian as well as with multipoles on other fragments. It is possible to input a screening term that accounts for the charge penetration. Typically the multipole expansion points are located on atomic nuclei and at bond midpoints.

2) Dipole polarizability. An arbitrary number of dipole polarizability tensors can be used to calculate the induced dipole on a fragment due to the electric field of the ab initio system as well as all the other fragments. These induced dipoles interact with the ab initio system as well as the other EFPs, in turn changing their electric fields. All induced dipoles are therefore iterated to self-consistency. Typically the polarizability tensors are located at

the centroid of charge of each localized orbital of a fragment.

3) Repulsive potential. Two different forms for the repulsive potentials are used: one for ab initio-EFP repulsion and one for EFP-EFP repulsion. The form of the potentials is empirical, and consists of distributed gaussian or exponential functions, respectively. The primary contribution to the repulsion is the quantum mechanical exchange repulsion, but the fitting technique used to develop this term also includes the effects of charge transfer. Typically these fitted potentials are located on atomic nuclei within the fragment.

### \*\*\* Constructing an EFP Using GAMESS \*\*\*

RUNTYPE=MOROKUMA assists in the decomposition of inter- molecular interaction energies into electrostatic, polarization, charge transfer, and exchange repulsion contributions. This is very useful in developing EFPs since potential problems can be attributed to a particular term by comparison to these energy components for a particular system.

A molecular multipole expansion can be obtained using \$ELMOM. A distributed multipole expansion can be obtained by either a Mulliken-like partitioning of the density (using \$STONE) or by using localized molecular orbitals (\$LOCAL: DIPDCM and QADDCM). The molecular dipole polarizability tensor can be obtained during a Hessian run (\$CPHF), and a distributed LMO polarizability expression is also available (\$LOCAL: POLDCM).

The repulsive potential is derived by fitting the difference between ab initio computed intermolecular interaction energies, and the form used for Coulomb and polarizability interactions. This difference is obtained at a large number of different interaction geometries, and is then fitted. Thus, the repulsive term is implicitly a function of the choices made in representing the Coulomb and polarizability terms. Note that GAMESS currently does not provide a way to obtain these repulsive potential, or the charge interpenetration screening parameters.

Since you cannot develop all terms necessary to define a new EFP's \$FRAGNAME group using GAMESS, in practice you will be limited to using the internally stored H2OEF2 potential mentioned below.

### \*\*\* Current Limitations \*\*\*

1. The energy and energy gradient are programmed, which permits RUNTYPE=ENERGY, GRADIENT, and numerical HESSIAN. The necessary modifications to use the EFP gradients while moving on the potential surface are programmed for RUNTYPE=OPTIMIZE, SADPOINT, and IRC (see reference 3), but the other gradient based potential surface explorations such as DRC are not yet available. Finally, RUNTYPE=PROP is also permissible.
2. The ab initio system must be treated with RHF, ROHF, UHF, or the open shell SCF wavefunctions permitted by the GVB code. The correlated methods in GAMESS (MP2 and CI) should not be used, since the available H2OEF2 potential was derived at the RHF level, and therefore does not contain dispersion terms. A correlated computation on the ab initio system without these terms in the EFP will probably lead to unphysical results. MCSCF and GVB-PP represent a gray area, but Mo Krauss has obtained some results for a solute described by an MCSCF wavefunction in which the EFP results agree well with fully ab initio computations (in structures and interaction energies).
3. EFPs can move relative to the ab initio system and relative to each other, but the internal structure of an EFP is frozen.

4. The boundary between the ab initio system and the EFPs must not be placed across a chemical bond.
5. Calculations must be done in C1 symmetry at present. Enter NOSYM=1 in \$CONTRL to accomplish this.
6. Reorientation of the fragments and ab initio system is not well coordinated. If you are giving Cartesian coordinates for the fragments (COORD=CART in \$EFRAG), be sure to use \$CONTRL's COORD=UNIQUE option so that the ab initio molecule is not reoriented.
7. If you need IR intensities, you have to use NVIB=2. The potential surface is usually very soft for EFP motions, and double differenced Hessians should usually be obtained.

\*\*\* Practical hints for using EFPs \*\*\*

At the present time, we have only one EFP suitable for general use. This EFP models water, and its numerical parameters are internally stored, using the fragment name H2OEF2. These numerical parameters are improved values over the H2OEF1 set which were presented and used in reference 2, and they also include the improved EFP-EFP repulsive term defined in reference 3. The H2OEF2 water EFP was derived from RHF/DH(d,p) computations on the water dimer system. When you use it, therefore, the ab initio part of your system should be treated at the SCF level, using a basis set of the same quality (ideally DH(d,p), but probably other DZP sets such as 6-31G(d,p) will give good results as well). Use of better basis sets than DZP with this water EFP has not been tested.

As noted, effective fragments have frozen internal geometries, and therefore only translate and rotate with respect to the ab initio region. An EFP's frozen coordinates are positioned to the desired location(s) in \$EFRAG as follows:

- a) the corresponding points are found in \$FRAGNAME.
- b) Point -1- in \$EFRAG and its FRAGNAME equivalent are made to coincide.
- c) The vector connecting -1- and -2- is aligned with the corresponding vector connecting FRAGNAME points.
- d) The plane defined by -1-, -2-, and -3- is made to coincide with the corresponding FRAGNAME plane.

Therefore the 3 points in \$EFRAG define only the relative position of the EFP, and not its internal structure. So, if the "internal structure" given by points in \$EFRAG differs from the true values in \$FRAGNAME, then the order in which the points are given in \$EFRAG can affect the positioning of the fragment. It may be easier to input water EFPs if you use the Z-matrix style to define them, because then you can ensure you use the actual frozen geometry in your \$EFRAG. Note that the H2OEF2 EFP uses the frozen geometry  $r(\text{OH})=0.9438636$ ,  $a(\text{HOH})=106.70327$ , and the names of its 3 fragment points are ZO1, ZH2, ZH3.

The translations and rotations of EFPs with respect to the ab initio system and one another are automatically quite soft degrees of freedom. After all, the EFP model is meant to handle weak interactions! Therefore the satisfactory location of structures on these flat surfaces will require use of a tight convergence on the gradient: OPTTOL=0.00001 in the \$STATPT group.

### QM/MM across covalent bonds

Recent work by Visvaldas Kairys and Jan Jensen has made it possible to extend the EFP methodology beyond the simple solute/solvent case described above. When there is a covalent bond between the portion of the system to be modeled by quantum mechanics, and the portion which is to be treated by EFP multipole and polarizability terms, an additional layer is needed

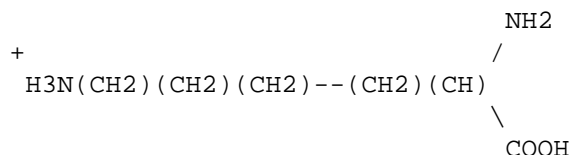
in the model. The covalent linkage is not so simple as the interactions between closed shell solute and solvent molecules. The "buffer zone" between the quantum mechanics and the EFP consists of frozen nuclei, and frozen localized orbitals, so that the quantum mechanical region sees an orbital representation of the closest particles, and multipoles etc. beyond that. Since the orbitals in the buffer zone are frozen, it need extend only over a few atoms in order to keep the orbitals in the fully optimized quantum region within that region.

A preprint describing the methodology exists: "QM/MM boundaries across covalent bonds: a frozen LMO based approach for the Effective Fragment Potential method" V.Kairys, J.H.Jensen J.Phys.Chem.A submitted.

The general outline of this kind of computation is as follows:

- a) a full quantum mechanics computation on a system containing the quantum region, the buffer region, and a few atoms into the EFP region, to obtain the frozen localized orbitals in the buffer zone. This is called the "truncation run".
- b) a full quantum mechanics computation on a system with all quantum region atoms removed, and with the frozen localized orbitals in the buffer zone. The necessary multipole and polarizability data to construct the EFP that will describes the EFP region will be extracted from the wavefunction. This is called the "MAKEFP run". It is possible to use several such runs if the total EFP region is quite large.
- c) The intended QM/MM run(s), after combining the information from these first two types of runs.

As an example, consider a protonated lysine residue which one might want to consider quantum mechanically in a protein whose larger parts are to be treated with an EFP. The protonated lysine is



The bonds which you see drawn show how the molecule is partitioned between the quantum mechanical side chain, a CH<sub>2</sub>CH group in the buffer zone, and eventually two different EFPs may be substituted in the area of the NH<sub>2</sub> and COOH groups to form the protein backbone. The "truncation run" will be on the entire system as you see it, with the 13 atoms in the side chain first in \$DATA, the 5 atoms in the buffer zone next in \$DATA, and the simplified EFP region at the end. This run will compute the full quantum wavefunction by RUNTYP=ENERGY, followed by the calculation of localized orbitals, followed by truncation of the localized orbitals that are found in the buffer zone so that they contain no contribution from AOs outside the buffer zone. The key input groups for this run are

```
$contrl
```

```
$truncn doproj=.true. plain=.true. natab=13 natbf=5 $end
```

This will generate a total of 6 localized molecular orbitals in the buffer zone (one CC, three CH, two 1s inner shells), expanded in terms of atomic orbitals located only on those atoms.

The truncation run prepares template input files for the next run, including adjustments of nuclear charges at boundaries, etc.

The "MAKEFP" run drops all 13 atoms in the quantum region, and uses the frozen orbitals just prepared to obtain a wavefunction for the EFP region. The carbon atom in the buffer zone that is connected to the now absent QM region will have its nuclear charge changed from 6 to 5 to account for a missing electron. The key input for this RUNTYP=MAKEFP job is the six orbitals

in \$VEC, plus the groups

```
$guess guess=huckel insorb=6 $end
$mofrz frz=.true. ifrz(1)=1,2,3,4,5,6 $end
$stone
QMMMbuf
$end
```

which will cause the wavefunction optimization for the remaining atoms to optimize orbitals only in the NH<sub>2</sub> and COOH pieces. After this wavefunction is found, the run extracts the EFP information needed for the QM/MM third run(s). This means running the Stone analysis for distributed multipoles, and obtaining a polarizability tensor for each localized orbital in the EFP region.

The QM/MM run might be RUNTYP=OPTIMIZE, etc. depending on what you want to do with the quantum atoms, and its \$DATA group will contain both the 13 fully optimized atoms, and the 5 buffer atoms, and a basis set will exist on both sets of atoms. The carbon atom in the buffer zone that borders the EFP region will have its nuclear charge set to 4 since now two bonding electrons to the EFP region are lost. \$VEC input will provide the six frozen orbitals in the buffer zone. The EFP atoms are defined in a fragment potential group.

The QM/MM run could use RHF or ROHF wavefunctions, to geometry optimize the locations of the quantum atoms (but not of course the frozen buffer zone or the EFP piece). It could remove the proton to compute the proton affinity at that terminal nitrogen, hunt for transition states, and so on. Presently the gradient for GVB and MCSCF is not quite right, so their use is discouraged.

Input to control the QM/MM preparation is \$TRUNCN and \$MOFRZ groups. There are a number of other parameters in various groups, namely QMMMbuf in \$STONE, MOIDON and POLNUM in \$LOCAL, NBUFFMO in \$EFRAG, and INSORB in \$GUESS that are relevant to this kind of computation. For RUNTYP=MAKEFP, the biggest choices are LOCAL=RUEDENBRG vs. BOYS, and POLNUM in \$LOCAL, otherwise this is pretty much a standard RUNTYP=ENERGY input file.

Source code distributions of GAMESS contain a directory named ~/gamess/tools/efp, which has various tools for EFP manipulation in it, described in file readme.1st. A full input file for the protonated lysine molecule is included, with instructions about how to proceed to the next steps. Tips on more specialized input possibilities are appended to the file readme.1st.

### \*\*\* References \*\*\*

The first of these is more descriptive, and the second has a very detailed derivation of the method. The latest EFP developments are discussed in the 3rd paper.

1. "Effective fragment method for modeling intermolecular hydrogen bonding effects on quantum mechanical calculations" J.H.Jensen, P.N.Day, M.S.Gordon, H.Basch, D.Cohen, D.R.Garmer, M.Krauss, W.J.Stevens in "Modeling the Hydrogen Bond" (D.A. Smith, ed.) ACS Symposium Series 569, 1994 , pp 139-151.
2. "An effective fragment method for modeling solvent effects in quantum mechanical calculations". P.N.Day, J.H.Jensen, M.S.Gordon, S.P.Webb, W.J.Stevens, M.Krauss, D.Garmer, H.Basch, D.Cohen J.Chem.Phys. 105, 1968-1986(1996 ).
3. "The effective fragment model for solvation: internal rotation in formamide" W.Chen and M.S.Gordon, J.Chem.Phys., 105, 11081-90(1996 )

4. "Transphosphorylation catalyzed by ribonuclease A: Computational study using ab initio EFPs" B.D.Wladkowski, M. Krauss, W.J.Stevens, J.Am.Chem.Soc. 117, 10537-10545(1995 ).
5. "A study of aqueous glutamic acid using the effective fragment potential model" P.N.Day, R.Pachter J.Chem.Phys. 107, 2990-9(1997 )
6. "Solvation and the excited states of formamide" M.Krauss, S.P.Webb J.Chem.Phys. 107, 5771-5(1997 )
7. "Study of small water clusters using the effective fragment potential method" G.N.Merrill, M.S.Gordon, J.Phys.Chem.A 102, 2650-7(1998 )
8. "Solvation of the Menshutkin Reaction: A Rigorous test of the Effective Fragment Model" S.P.Webb, M.S.Gordon J.Phys.Chem.A 103, 1265-73(1999)
9. "Solvation of Sodium Chloride: EFP study of  $\text{NaCl}(\text{H}_2\text{O})_n$ " C.P.Petersen, M.S.Gordon J.Phys.Chem.A 103, 4162-6(1999)

## MOPAC calculations within GAMESS

Parts of MOPAC 6.0 have been included in GAMESS so that the GAMESS user has access to three semiempirical wavefunctions: MNDO, AM1 and PM3. These wavefunctions are quantum mechanical in nature but neglect most two electron integrals, a deficiency that is (hopefully) compensated for by introduction of empirical parameters. The quantum mechanical nature of semiempirical theory makes it quite compatible with the ab initio methodology in GAMESS. As a result, very little of MOPAC 6.0 actually is incorporated into GAMESS. The part that did make it in is the code that evaluates

- 1) the one- and two-electron integrals,
- 2) the two-electron part of the Fock matrix,
- 3) the cartesian energy derivatives, and
- 4) the ZDO atomic charges and molecular dipole.

Everything else is actually GAMESS: coordinate input (including point group symmetry), the SCF convergence procedures, the matrix diagonalizer, the geometry searcher, the numerical hessian driver, and so on. Most of the output will look like an ab initio output.

It is extremely simple to perform one of these calculations. All you need to do is specify GBASIS=MNDO, AM1, or PM3 in the \$BASIS group. Note that this not only picks a particular Slater orbital basis, it also selects a particular "hamiltonian", namely a particular parameter set.

MNDO, AM1, and PM3 will not work with every option in GAMESS. Currently the semiempirical wavefunctions support SCFTYP=RHF, UHF, and ROHF in any combination with RUNTYP=ENERGY, GRADIENT, OPTIMIZE, SADPOINT, HESSIAN, and IRC. Note that all hessian runs are numerical finite differencing. The MOPAC CI and half electron methods are not supported.

Because the majority of the implementation is GAMESS rather than MOPAC you will notice a few improvements. Dynamic memory allocation is used, so that GAMESS uses far less memory for a given size of molecule. The starting orbitals for SCF calculations are generated by a Huckel initial guess routine. Spin restricted (high spin) ROHF can be performed. Converged SCF orbitals will be labeled by their symmetry type. Numerical Hessians will make use of point group symmetry, so that only the symmetry unique atoms need to be displaced. Infrared intensities will be calculated at the end of hessian runs. We have not at present used the block diagonalizer during intermediate SCF iterations, so that the run time for a single geometry point in GAMESS is usually longer than in MOPAC. However, the geometry optimizer in GAMESS can frequently optimize the structure in fewer steps than the procedure in MOPAC. Orbitals and Hessians are punched out for convenient reuse in subsequent calculations. Your molecular orbitals can be drawn with the PLTORB graphics program.

To reduce CPU time, only the EXTRAP convergence accelerator is used by the SCF procedures. For difficult cases, the DIIS, RSTRCT, and/or SHIFT options will work, but may add significantly to the run time. With the Huckel guess, most calculations will converge acceptably without these special options.

MOPAC parameters exist for the following elements. The quote means that these elements are treated as "sparkles" rather than as atoms with genuine basis functions. For MNDO:



```

H
Li *           B C N O F
Na' *         Al Si P S Cl
K' * ... Zn * Ge * * Br
Rb' * ... * * Sn * * I
* * ... Hg * Pb *

```

```

                For AM1:
H
* *           B C N O F
Na' *         Al Si P S Cl
K' * ... Zn * Ge * * Br
Rb' * ... * * Sn * * I
* * ... Hg * * *

                For PM3:
H
* Be           * C N O F
Na' Mg         Al Si P S Cl
K' * ... Zn Ga Ge As Se Br
Rb' * ... Cd In Sn Sb Te I
* * ... Hg Tl Pb Bi

```

Semiempirical calculations are very fast. One of the motives for the MOPAC implementation within GAMESS is to take advantage of this speed. Semiempirical models can rapidly provide reasonable starting geometries for ab initio optimizations. Semiempirical hessian matrices are obtained at virtually no computational cost, and may help dramatically with an ab initio geometry optimization. Simply use HESS=READ in \$STATPT to use a MOPAC \$HESS group in an ab initio run.

It is important to exercise caution as semiempirical methods can be dead wrong! The reasons for this are bad parameters (in certain chemical situations), and the underlying minimal basis set. A good question to ask before using MNDO, AM1 or PM3 is "how well is my system modeled with an ab initio minimal basis sets, such as STO-3G?" If the answer is "not very well" there is a good chance that a semiempirical description is equally poor.

### Molecular Properties

These two papers are of general interest:

A.D.Buckingham, J.Chem.Phys. 30, 1580-1585(1959 ).

D.Neumann, J.W.Moskowitz J.Chem.Phys. 49, 2056-2070(1968 ).

All units are derived from the atomic units for distance and the monopole electric charge, as given below.

distance	- 1 au = 5.291771E-09 cm
monopole	- 1 au = 4.803242E-10 esu 1 esu = sqrt(g-cm <sup>3</sup> )/sec
dipole	- 1 au = 2.541766E-18 esu-cm 1 Debye = 1.0E-18 esu-cm
quadrupole	- 1 au = 1.345044E-26 esu-cm <sup>2</sup> 1 Buckingham = 1.0E-26 esu-cm <sup>2</sup>
octupole	- 1 au = 7.117668E-35 esu-cm <sup>3</sup>
electric potential	- 1 au = 9.076814E-02 esu/cm

electric field	- 1 au = 1.715270E+07 esu/cm <sup>2</sup> 1 esu/cm <sup>**2</sup> = 1 dyne/esu
electric field gradient	- 1 au = 3.241390E+15 esu/cm <sup>3</sup>

The atomic unit for electron density is electron/bohr<sup>3</sup> for the total density, and 1/bohr<sup>3</sup> for an orbital density.

The atomic unit for spin density is excess alpha spins per unit volume,  $h/4 \cdot \pi \cdot \text{bohr}^3$ . Only the expectation value is computed, with no constants premultiplying it.

IR intensities are printed in Debye<sup>2</sup>/amu-Angstrom<sup>2</sup>. These can be converted into intensities as defined by Wilson, Decius, and Cross's equation 7.9.25, in km/mole, by multiplying by 42.255. If you prefer 1/atm-cm<sup>2</sup>, use a conversion factor of 171.65 instead. A good reference for deciphering these units is A.Komornicki, R.L.Jaffe J.Chem.Phys. 1979 , 71, 2150-2155. A reference showing how IR intensities change with basis improvements at the HF level is Y.Yamaguchi, M.Frisch, J.Gaw, H.F.Schaefer, J.S.Binkley, J.Chem.Phys. 1986 , 84, 2262-2278.

Raman intensities in A<sup>\*\*4</sup>/amu multiply by 6.0220E-09 for units of cm<sup>\*\*4</sup>/g.

### Localization tips

Three different orbital localization methods are implemented in GAMESS. The energy and dipole based methods normally produce similar results, but see M.W.Schmidt, S.Yabushita, M.S.Gordon in J.Chem.Phys., 1984 , 88, 382-389 for an interesting exception. You can find references to the three methods at the beginning of this chapter.

The method due to Edmiston and Ruedenberg works by maximizing the sum of the orbitals' two electron self repulsion integrals. Most people who think about the different localization criteria end up concluding that this one seems superior. The method requires the two electron integrals, transformed into the molecular orbital basis. Because only the integrals involving the orbitals to be localized are needed, the integral transformation is actually not very time consuming.

The Boys method maximizes the sum of the distances between the orbital centroids, that is the difference in the orbital dipole moments.

The population method due to Pipek and Mezey maximizes a certain sum of gross atomic Mulliken populations. This procedure will not mix sigma and pi bonds, so you will not get localized banana bonds. Hence it is rather easy to find cases where this method give different results than the Ruedenberg or Boys approach.

GAMESS will localize orbitals for any kind of RHF, UHF, ROHF, or MCSCF wavefunctions. The localizations will automatically restrict any rotation that would cause the energy of the wavefunction to be changed (the total wavefunction is left invariant). As discussed below, localizations for GVB or CI functions are not permitted.

The default is to freeze core orbitals. The localized valence orbitals are scarcely changed if the core orbitals are included, and it is usually convenient to leave them out. Therefore, the default localizations are: RHF functions localize all doubly occupied valence orbitals. UHF functions localize all valence alpha, and then all valence beta orbitals. ROHF functions localize all valence doubly occupied orbitals, and all singly occupied orbitals, but do not mix these two

orbital spaces. MCSCF functions localize all valence MCC type orbitals, and localize all active orbitals, but do not mix these two orbital spaces. To recover the invariant MCSCF function, you must be using a FORS=.TRUE. wavefunction, and you must set GROUP=C1 in \$DRT, since the localized orbitals possess no symmetry.

In general, GVB functions are invariant only to localizations of the NCO doubly occupied orbitals. Any pairs must be written in natural form, so pair orbitals cannot be localized. The open shells may be degenerate, so in general these should not be mixed. If for some reason you feel you must localize the doubly occupied space, do a RUNTYP=PROP job. Feed in the GVB orbitals, but tell the program it is SCFTYP=RHF, and enter a negative ICHARG so that GAMESS thinks all orbitals occupied in the GVB are occupied in this fictitious RHF. Use NINA or NOUTA to localize the desired doubly occupied orbitals. Orbital localization is not permitted for CI, because we cannot imagine why you would want to do that anyway.

Boys localization of the core orbitals in molecules having elements from the third or higher row almost never succeeds. Boys localization including the core for second row atoms will often work, since there is only one inner shell on these. The Ruedenberg method should work for any element, although including core orbitals in the integral transformation is more expensive.

The easiest way to do localization is in the run which generates the wavefunction, by selecting LOCAL=xxx in the \$CONTRL group. However, localization may be conveniently done at any time after determination of the wavefunction, by executing a RUNTYP=PROP job. This will require only \$CONTRL, \$BASIS/\$DATA, \$GUESS (pick MOREAD), the converged \$VEC, possibly \$SCF or \$DRT to define your wavefunction, and optionally some \$LOCAL input.

There is an option to restrict all rotations that would mix orbitals of different symmetries. SYMLOC=.TRUE. yields only partially localized orbitals, but these still possess symmetry. They are therefore very useful as starting orbitals for MCSCF or GVB-PP calculations. Because they still have symmetry, these partially localized orbitals run as efficiently as the canonical orbitals. Because it is much easier for a user to pick out the bonds which are to be correlated, a significant number of iterations can be saved, and convergence to false solutions is less likely.

\* \* \*

The most important reason for localizing orbitals is to analyze the wavefunction. A simple example is to make contour plots of the resulting orbitals with the PLTORB graphics codes, or perhaps to read the localized orbitals in during a RUNTYP=PROP job to examine their Mulliken populations. MCSCF localized orbitals can be input to a CI calculation, to generate the corresponding density matrix, which contains much useful information about electron populations and chemical bonding. For example,

J.Am.Chem.Soc., 104, 960-967 (1982 ),  
 J.Am.Chem.Soc., 113, 5231-5243 (1991 ),  
 Theoret.Chim.Acta, 83, 57-68 (1992 ).

In addition, the energy of your molecule can be partitioned over the localized orbitals so that you may be able to understand the origin of barriers, etc. This analysis can be made for the SCF energy, and also the MP2 correction to the SCF energy, which requires two separate runs. An explanation of the method, and application to hydrogen bonding may be found in J.H.Jensen, M.S.Gordon, J.Phys.Chem. 1995 , 99 , 8091-8107.

Analysis of the SCF energy is based on the localized charge distribution (LCD) model: W.England and M.S.Gordon, J.Am.Chem.Soc. 93, 4649-4657 (1971 ). This is implemented for RHF and ROHF wavefunctions, and it requires use of the Ruedenberg localization method, since it needs the two electron integrals to correctly compute energy sums. All orbitals must be included

in the localization, even the cores, so that the total energy is reproduced.

The LCD requires both electronic and nuclear charges to be partitioned. The orbital localization automatically accomplishes the former, but division of the nuclear charge may require some assistance from you. The program attempts to correctly partition the nuclear charge, if you select the MOIDON option, according to the following: a Mulliken type analysis of the localized orbitals is made. This determines if an orbital is a core, lone pair, or bonding MO. Two protons are assigned to the nucleus to which any core or lone pair belongs. One proton is assigned to each of the two nuclei in a bond. When all localized orbitals have been assigned in this manner, the total number of protons which have been assigned to each nucleus should equal the true nuclear charge.

Many interesting systems (three center bonds, back-bonding, aromatic delocalization, and all charged species) may require you to assist the automatic assignment of nuclear charge. First, note that MOIDON reorders the localized orbitals into a consistent order: first comes any core and lone pair orbitals on the 1st atom, then any bonds from atom 1 to atoms 2, 3, ..., then any core and lone pairs on atom 2, then any bonds from atom 2 to 3, 4, ..., and so on. Let us consider a simple case where MOIDON fails, the ion  $\text{NH}_4^+$ . Assuming the nitrogen is the 1st atom, MOIDON generates

```

NNUCMO=1, 2, 2, 2, 2
MOIJ=1, 1, 1, 1, 1
      2, 3, 4, 5
ZIJ=2. 0, 1. 0, 1. 0, 1. 0, 1. 0,
      1. 0, 1. 0, 1. 0, 1. 0

```

The columns (which are LMOs) are allowed to span up to 5 rows (the nuclei), in situations with multicenter bonds. MOIJ shows the Mulliken analysis thinks there are four NH bonds following the nitrogen core. ZIJ shows that since each such bond assigns one proton to nitrogen, the total charge of N is +6. This is incorrect of course, as indeed will always happen to some nucleus in a charged molecule. In order for the energy analysis to correctly reproduce the total energy, we must ensure that the charge of nitrogen is +7. The least arbitrary way to do this is to increase the nitrogen charge assigned to each NH bond by 1/4. Since in our case NNUCMO and MOIJ and much of ZIJ are correct, we need only override a small part of them with \$LOCAL input:

```

IJMO(1)=1, 2, 1, 3, 1, 4, 1, 5
ZIJ(1)=1. 25, 1. 25, 1. 25, 1. 25

```

which changes the charge of the first atom of orbitals 2 through 5 to 5/4, changing ZIJ to

```

ZIJ=2. 0, 1. 25, 1. 25, 1. 25, 1. 25,
      1. 0, 1. 0, 1. 0, 1. 0

```

The purpose of the IJMO sparse matrix pointer is to let you give only the changed parts of ZIJ and/or MOIJ.

Another way to resolve the problem with  $\text{NH}_4^+$  is to change one of the 4 equivalent bond pairs into a "proton". A "proton" orbital AH treats the LMO as if it were a lone pair on A, and so assigns +2 to nucleus A. Use of a "proton" also generates an imaginary orbital, with zero electron occupancy. For example, if we make atom 2 in  $\text{NH}_4^+$  a "proton", by

```

IPROT(1)=2
NNUCMO(2)=1
IJMO(1)=1, 2, 2, 2   MOIJ(1)=1, 0   ZIJ(1)=2. 0, 0. 0

```

the automatic decomposition of the nuclear charges will be

```

NNUCMO=1, 1, 2, 2, 2, 1
MOIJ=1, 1, 1, 1, 1, 2
      3, 4, 5
ZIJ=2. 0, 2. 0, 1. 0, 1. 0, 1. 0, 1. 0
      1. 0, 1. 0, 1. 0

```

The 6th column is just a proton, and the decomposition will not give any electronic energy associated with this "orbital", since it is vacant. Note that the two ways we have dissected the nuclear charges for  $\text{NH}_4^+$  will both yield the correct total energy, but will give very different individual orbital components. Most people will feel that the first assignment is the least arbitrary, since it treats all four NH bonds equivalently.

However you assign the nuclear charges, you must ensure that the sum of all nuclear charges is correct. This is most easily verified by checking that the energy sum equals the total SCF energy of your system.

As another example,  $\text{H}_3\text{PO}$  is studied in EXAM26.INP. Here the MOIDON analysis decides the three equivalent orbitals on oxygen are O lone pairs, assigning +2 to the oxygen nucleus for each orbital. This gives  $Z(\text{O})=9$ , and  $Z(\text{P})=14$ . The least arbitrary way to reduce  $Z(\text{O})$  and increase  $Z(\text{P})$  is to recognize that there is some backbonding in these "lone pairs" to P, and instead assign the nuclear charge of these three orbitals by 1/3 to P, 5/3 to O.

Because you may have to make several runs, looking carefully at the localized orbital output before the correct nuclear assignments are made, there is an option to skip directly to the decomposition when the orbital localization has already been done. Use

```
$CONTRL RUNTYP=PROP
$GUESS GUESS=MOREAD NORB=
$VEC containing the localized orbitals!
$TWOEI
```

The latter group contains the necessary Coulomb and exchange integrals, which are punched by the first localization, and permits the decomposition to begin immediately.

SCF level dipoles can also be analyzed using the DIPDCM flag in \$LOCAL. The theory of the dipole analysis is given in the third paper of the LCD sequence. The following list includes application of the LCD analysis to many problems of chemical interest:

W.England, M.S.Gordon J.Am.Chem.Soc. 93, 4649-4657 (1971 )  
 W.England, M.S.Gordon J.Am.Chem.Soc. 94, 4818-4823 (1972 )  
 M.S.Gordon, W.England J.Am.Chem.Soc. 94, 5168-5178 (1972 )  
 M.S.Gordon, W.England Chem.Phys.Lett. 15, 59-64 (1972 )  
 M.S.Gordon, W.England J.Am.Chem.Soc. 95, 1753-1760 (1973 )  
 M.S.Gordon J.Mol.Struct. 23, 399 (1974 )  
 W.England, M.S.Gordon, K.Ruedenberg, Theoret.Chim.Acta 37, 177-216 (1975 )  
 J.H.Jensen, M.S.Gordon, J.Phys.Chem. 99, 8091-8107(1995 )  
 J.H.Jensen, M.S.Gordon, J.Am.Chem.Soc. 117, 8159-8170(1995 )  
 M.S.Gordon, J.H.Jensen, Acc.Chem.Res. 29, 536-543(1996 )

\* \* \*

It is also possible to analyze the MP2 correlation correction in terms of localized orbitals, for the RHF case. The method is that of G.Peterssen and M.L.Al-Laham, J.Chem.Phys., 94, 6081-6090 (1991 ). Any type of localized orbital may be used, and because the MP2 calculation typically omits cores, the \$LOCAL group will normally include only valence orbitals in the localization. As mentioned already, the analysis of the MP2 correction must be done in a separate run from the SCF analysis, which must include cores in order to sum up to the total SCF energy.

\* \* \*

Typically, the results are most easily interpreted by looking at "the bigger picture" than at

"the details". Plots of kinetic and potential energy, normally as a function of some coordinate such as distance along an IRC, are the most revealing. Once you determine, for example, that the most significant contribution to the total energy is the kinetic energy, you may wish to look further into the minutia, such as the kinetic energies of individual localized orbitals, or groups of LMOs corresponding to an entire functional group.

### Transition Moments and Spin-Orbit Coupling

GAMESS can compute transition moments and oscillator strengths for the radiative transition between two CI wavefunctions. The moments are computed using both the "length (dipole) form" and "velocity form". The two values will be slightly different as the CI wavefunction does not exactly satisfy the Hellmann-Feynman theorem. This basic computation is OPERAT=DM, the other operators are various spin-orbit coupling options.

GAMESS can also compute the "microscopic Breit-Pauli spin orbit operator". This can be done for general spins, even more than two different spin multiplicities at once, for general active spaces. The full Breit-Pauli operator can be computed exactly, or approximated in two ways: complete elimination of the  $2e^-$  term, whose absence can be approximately accounted for by means of effective nuclear charges, or inclusion of only core-active  $2e^-$  terms which typically account for 90% or more of the two electron term, but little of the  $2e^-$  terms CPU time. It is also possible to obtain the dipole transition moment between spin-mixed wavefunctions, which do not have a rigorous S quantum no.

Spin orbit coupling is always performed in a quasi-degenerate perturbative manner. Typically the states close in energy are included into the spin orbit coupling matrix. "Close" has an easily understandable meaning, since in the limit of small coupling the quasi-degenerate treatment is reduced to a second order perturbative treatment, that is, the effect of a state upon the state of primary interest is given by the square of the spin orbit coupling matrix element divided by the difference of the adiabatic energies. This is useful to keep in mind when deciding how many CI states to include in the matrix. The states that are included are treated in a fashion that is equivalent to infinite order perturbation theory (exact) whereas the states that are not included make no contribution.

The orbitals for the CI can be one common set of orbitals used by all CI states. If one set of orbitals is used, the transition moment or spin-orbit coupling can be found for any type of CI wavefunction GAMESS can compute.

Alternatively, two sets of orbitals (obtained from separate MCSCF orbital optimizations) can be used. Two or more separate CIs will be carried out. The two MO sets must share a common set of frozen core orbitals, and the CI -must- be of the complete active space type. These restrictions are needed to leave the CI wavefunctions invariant under the necessary rotation to corresponding orbitals. The non-orthogonal procedure implemented is a GUGA driven equivalent to the method of Lengsfeld, et al. Note that the FOCI and SOCI methods described by these workers are not available in GAMESS.

If you would like to use separate orbitals for the states, use the FCORE option in \$MCSCF in a SCFTYP=MCSCF optimization of the orbitals. Typically you would optimize the ground state completely, and use these MCSCF orbitals in an optimization of the excited state, under the constraint of FCORE=.TRUE. The core orbitals of such a MCSCF optimization should be declared MCC, rather than FZC, even though they are frozen.

In the case of transition moments either one or two CI calculations are performed, necessarily on states of the same multiplicity. If all the states are the same symmetry or you

are using GROUP=C1 to make them all appear at once, only a \$DRT1 is read. However you might use more than one \$DRT to do separate CI computations on specific CI state symmetries. Spin-orbit coupling runs almost always does two or more CI calculations, as the states to be coupled are usually of different multiplicities, and perhaps also of several spatical symmetries. So, spin-orbit runs might read only \$DRT1, but normally read more: \$DRT1, \$DRT2, ... The first CI calculation, defined by \$DRT1, must be for the state of lower spin multiplicity, with \$DRT2, \$DRT3, ... being successively higher multiplicities. The limit is 64 separate CI computations, \$DRT64.

The choice between HSO2 and HSO2FF is very often in favour of the former. HSO2 computes the matrix elements in CSF basis and then contracts them with CI coefficients, whereas HSO2FF uses a generalised density in AO basis computed for each pair of states, thus HSO2 is much more efficient in case of multiple states given in IROOTS. HSO2FF takes less memory for integral storage, thus it can be superior in case of small active spaces and large basis sets. The numerical results with HSO2 and HSO2FF should be identical within machine and algorithmic accuracy.

You can use spatial symmetry in the various \$DRTx computations to decrease the size of the CI. Since the corresponding orbital transformation permutes the active orbitals in an uncontrolled way, use ISTSYM to define the desired state symmetry. Do two separate CI computations in the case of being interested in including two states of the same multiplicity but of different spatial symmetry.

The spin-orbit operator contains a one electron term arising from Pauli's reduction of the hydrogenic Dirac equation to one-component form, and a two electron term added by Breit. The only practical limitation on the computation of the Breit term is that HSO2FF is limited to 10 active orbitals on 32 bit machines, and to about 26 active orbitals on 64 bit machines. The spin-orbit matrix elements vanish for  $\Delta S > 1$ , but it is possible to include three or more spins in the computation. Since singlets interact with triplets, and triplets interact with pentuplets, inclusion of S=0,1,2 simultaneously lets you pick up the indirect interaction between singlets and pentuplets that the intermediate triplets afford.

As an approximation, the nuclear charge appearing in the one electron term can be regarded as an empirical scale factor, compensating for the omission of the two electron operator. In addition, these effective charges are often used to compensate for missing nodes in valence orbitals of ECP runs, in which case the ZEFF are typically very far from the two nuclear charges. ZEFTYP selects some built in values obtained by S.Koseki et al, but if you have some favorite parameters, they can be read in as the ZEFF input array. Effective charges may be used for any OPERAT, but are most often used with HSO1.

Various symmetries are used to avoid computing zero spin-orbit matrix elements. NOSYM in \$TRANST allows some control over this: NOSYM=1 gives up point group symmetry completely, while 2 turns off additional symmetries such as spin selection rules. HSO1,2,2P compute all matrix elements in a group (i.e. between two \$DRTx groups with fixed Ms(ket)-Ms(bra)) if at least one of them does not vanish by symmetry, and HSO2PP actually avoids computation for each pair of states if forbidden by symmetry. Setting NOSYM=2 will cause HSO2FF to consider the elements between two singlets, which are always calculated for HSO1,2,2P when transition dipoles are requested as well.

The users are encouraged to specify full symmetry in their \$DATA input even though they may choose to set the symmetry in \$DRT to C1. The CI states will be labelled in the group given in \$DATA. The use of non-Abelian symmetry> is limited by the absence of non-Abelian CI in GAMESS. In this case the users can choose between setting full non-Abelian symmetry in \$DATA and C1 in \$DRT or else an Abelian subgroup in both \$DATA and \$DRT. The latter choice

appears to be most efficient at present.

SYMTOL has a dramatic effect on the run speed. This cutoff is applied to CSF coefficients, their products, and these products times CSF orbital overlaps. The value permits a tradeoff of accuracy for run time, and since the error in the spin-orbit properties approaches SYMTOL mainly for SOCI functions, it may be useful to increase SYMTOL to save time for CAS or FO CI functions. Some experimenting will tell you what you can get away with. SYMTOL is also used during CI state symmetry assignment, for NOIRR=-1 in \$DRT.

In case if you do not provide enough storage for the form factors sorting then some extra disk space will be used; the extra disk space can be eliminated if you set SAVDSK=.TRUE. (the amount of savings depends on the active space and memory provided, in some cases it can decrease the disk space up to one order of magnitude). The form factors are in binary format, and so can be transferred between computers only if they have compatible binary files. There is a built-in check for consistency of a restart file DAFL30 with the current run parameters.

\* \* \*

The transition moment and spin orbit coupling driver is a rather restricted path through GAMESS, you should

- 1) Give SCFTYP=NONE. \$GUESS is not read, as the program expects to MOREAD the orbitals in a \$VEC1 group, and perhaps a \$VEC2. It is not possible to reorder MOs.
- 2) \$CIINP is not read. The CI is hardwired to consist of CI DRT generation, integral transformation/sorting, Hamiltonian generation, and diagonalization. This means \$DRT1 (and maybe \$DRT2,...), \$TRANS, \$CISORT, \$GUGEM, and \$GUGDIA input is read, and acted upon.
- 3) The density matrices are not generated, and so no properties (other than the transition moment or the spin-orbit coupling) are computed.
- 4) There is no restart capability provided,, except for saving some form-factor information.
- 5) CI DRT input is given in \$DRT1, \$DRT2, ....., \$DRT64. These must go from lowest multiplicity to highest.
- 6) IROOTS will determine the number of CI states in each CI for which the properties are calculated. Use NSTATE to specify the number of CI states for the CI Hamiltonian diagonalisation. Sometimes the CI convergence is assisted by requesting more roots to be found in the diagonalization than you want to include in the property calculation.
- 7) Compared to older versions, the basis set has been fully generalized to allow any s, p, d, f, g, or l functions.

Reference for separate orbital optimization:

1. B.H.Lengsfeld, III, J.A.Jafri, D.H.Phillips, C.W.Bauschlicher, Jr. J.Chem.Phys. 74,6849-6856(1981 )

References for transition moments:

2. F.Weinhold, J.Chem.Phys. 54,1874-1881(1970 )
3. C.W.Bauschlicher, S.R.Langhoff Theoret.Chim.Acta 79:93-103(1991 )
4. "Intramediate Quantum Mechanics, 3rd Ed." Hans A. Bethe, Roman Jackiw



- Benjamin/Cummings Publishing, Menlo Park, CA (1986), chapters 10 and 11.
5. S.Koseki, M.S.Gordon J.Mol.Spectrosc. 123, 392-404(1987)

References for Zeff spin-orbit coupling, and ZEFTYP values:

6. S.Koseki, M.W.Schmidt, M.S.Gordon J.Phys.Chem. 96, 10768-10772 (1992)
7. S.Koseki, M.S.Gordon, M.W.Schmidt, N.Matsunaga J.Phys.Chem. 99, 12764-12772 (1995)
8. N.Matsunaga, S.Koseki, M.S.Gordon J.Chem.Phys. 104, 7988-7996 (1996)
9. S.Koseki, M.W.Schmidt, M.S.Gordon J.Phys.Chem.A 102, 10430-10435 (1998)

References for full Breit-Pauli spin-orbit coupling:

- 10.T.R.Furlani, H.F.King J.Chem.Phys. 82, 5577-5583 (1985)
- 11.H.F.King, T.R.Furlani J.Comput.Chem. 9, 771-778 (1988)
12. D.G.Fedorov, M.S.Gordon J.Chem.Phys. 112, 5611-5623 (2000)
- with the latter including information on the partial two electron operator method.

A recent application:

13. D.G.Fedorov, M.Evans, Y.Song, M.S.Gordon, C.Y.Ng J.Chem.Phys. 111, 6413(1999)

\* \* \*

Special thanks to Bob Cave and Dave Feller for their assistance in performing check spin-orbit coupling runs with the MELDF programs. Special thanks to Tom Furlani for contributing his 2e- spin-orbit code and answering many questions about its interface.

\* \* \*

We end with an example. Note that you must know what you are doing with term symbols, J quantum numbers, point group symmetry, and so on in order to make skillful use of this part of the program. Seeing your final degeneracies turn out like a text book says it should is beautiful!

```
! Compute the splitting of the famous sodium D line.
!
! The two SCF energies below give an excitation energy
! of 16,044 cm-1 to the 2-P term. The computed spin-orbit
! levels are at RELATIVE E=-10.269 and 5.135 cm-1, which
! means the 2-P level interval is 15.404 cm-1.
!
! Charlotte Moore's Atomic Energy Levels, volume 1, gives
! the experimental 2-P interval as 17.1963, the levels are
! at 2-S-1/2= 0.0, 2-P-1/2= 16,956.183, 2-P-3/2= 16,973.379
!
```

1. generate ground state 2-S orbitals by conventional ROHF. the energy of the ground state is -161.8413919816

```
--- $contrl scftyp=rohft mult=2 $end
--- $system kdiag=3 memory=300000 $end
--- $guess guess=huckel $end
```

2. generate excited state 2-P orbitals, using a state-averaged SCF wavefunction to ensure radial degeneracy of the 3p shell is preserved. The open shell SCF energy is -161.7682895801. The computation is both spin and space restricted open shell SCF on the 2-P Russell-Saunders term. Starting orbitals are reordered orbitals from step 1.

```
--- $contrl scftyp=gvt mult=2 $end
```

```

--- $system kdiag=3 memory=300000 $end
--- $guess guess=moread norb=13 norder=1 iorder(6)=7,8,9,6 $end
--- $scf nco=5 nseto=1 no(1)=3 rstrct=.true. couple=.true.
---         f(1)= 1.0 0.166666666666667
---         alpha(1)= 2.0 0.333333333333333 0.0
---         beta(1)= -1.0 -0.166666666666667 0.0 $end

```

3. compute spin orbit coupling in the 2-P term. The use of C1 symmetry in \$DRT1 ensures that all three spatial CSFs are kept in the CI function. In the preliminary CI, the spin function is just the alpha spin doublet, and all three roots should be degenerate, and furthermore equal to the GVB energy at step 2. The spin-orbit coupling code uses both doublet spin functions with each of the three spatial wavefunctions, so the spin-orbit Hamiltonian is a 6x6 matrix. The two lowest roots of the full 6x6 spin-orbit Hamiltonian are the doubly degenerate 2-P-1/2 level, while the other four roots are the degenerate 2-P-3/2 level.

```

$contrl scftyp=none cityp=guga runtyp=transitn mult=2 $end
$system memory=2000000 $end
$basis gbasis=n31 ngauss=6 $end
$gugdia nstate=3 $end
$stranst operat=hsol numvec=1 numci=1 nfzc=5 nocc=8
        iroots=3 zeff=10.04 $end
$drt1 group=c1 fors=.true. nfzc=5 nalp=1 nval=2 $end

$data
Na atom...2-P excited state...6-31G basis
Dnh 2

Na 11.0
$end

--- GVB ORBITALS --- GENERATED AT 7:46:08 CST 30-MAY-1996
Na atom...2-P excited state
E(GVB)= -161.7682895801, E(NUC)= .0000000000, 5 ITERS
$VEC1
1 1 9.97912679E-01 8.83038094E-03 0.00000000E+00...
... orbitals from step 2 go here ...
13 3-1.10674398E+00 0.00000000E+00 0.00000000E+00
$END

```