ACES II Release 2.5.0
User Manual
DRAFT COPY

Quantum Theory Project
P.O. Box 118435
University of Florida
Gainesville, FL 32611

January 29, 2006

Contents

1 Authors

1.1 Official ACES II citation

The users of ACES II must give the following citation:

> ACES II is a program product of the Quantum Theory Project, University of Florida. Authors: J.F. Stanton, J. Gauss, S.A. Perera, J.D. Watts, A.D. Yau, M. Nooijen, N. Oliphant, P.G. Szalay, W.J. Lauderdale, S.R. Gwaltney, S. Beck, A. Balková, D.E. Bernholdt, K.K. Baeck, P. Rozyczko, H. Sekino, C. Huber, J. Pittner, W. Cencek, D. Taylor, and R.J. Bartlett. Integral packages included are VMOL (J. Almlöf and P.R. Taylor); VPROPS (P. Taylor); ABACUS (T. Helgaker, H.J. Aa. Jensen, P. Jørgensen, J. Olsen, and P.R. Taylor); HONDO/GAMESS (M.W. Schmidt, K.K. Baldridge, 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).

The basis set distributed with the ACES II program system is obtained through Pacific Northwest National Laboratory (PNNL) and any calculation that uses basis sets from this catalog must give the following citation *in addition* to the ACES II citation.

> Basis sets were obtained from the Extensible Computational Chemistry Environment Basis Set Database, Version 1.0, as developed and distributed by the Molecular Science Computing Facility, Environmental and Molecular Sciences Laboratory which is part of the Pacific Northwest Laboratory, P.O. Box 999, Richland, Washington 99352, USA, and funded by the U.S. Department of Energy. The Pacific Northwest Laboratory is a multi-program laboratory operated by Battelle Memorial Institute for the U.S. Department Energy under contract DE-AC06-76RLO 1830. Contact Karen Schuchardt for further information.

1.2 Specific authors

- TD-CCSD energies. P.G. Szalay and A. Balkova.
- TD-CCSD analytical derivatives. P.G. Szalay.
- Dropped molecular orbitals in analytical derivative calculations for RHF, UHF, and ROHF references. K.-K. Baeck.
- Equation-of-motion CCSD calculation of dynamic polarizabilities (including partitioned scheme). J.F. Stanton, S.A. Perera, and M. Nooijen.

- Equation-of-motion CCSD calculation of NMR spin-spin coupling constants (including partitioned scheme). S.A. Perera and M. Nooijen.
- Partitioned equation-of-motion CCSD calculations of excitation energies. S.R. Gwaltney and M. Nooijen.
- Equation-of-motion CCSD gradient calculations for excited states. J.F. Stanton and J. Gauss

2 Preface

This manual is maintained as a set of LaTeX documents under CVS control along with the ACES II source code. Every version of the code has an accompanying manual, and while the user interface is relatively stable, it is not guaranteed that the manual of one version is compatible with binary executables of a different version. Any errors in the manual should be reported to `aces2@qtp.ufl.edu`

The goal of this manual is to provide beginners and experts alike with enough information to run any calculation that the software allows. A reference section is provided for further reading on the theories and algorithms implemented in the program.

3 Introduction

ACES II is a set of programs that performs *ab initio* quantum chemistry calculations. The package has a high degree of flexibility and supports many kinds of calculations at a number of levels of theory. The major strength of the program system is using "many-body" methods to treat electron correlation. These approaches, broadly categorized as many-body perturbation theory (MBPT) and the coupled-cluster (CC) approximation, offer a reliable treatment of correlation and have the attractive property of size-extensivity, which means the energies scale properly with the size of the system. As a result of this property, MBPT and CC methods are ideally suited for the study of chemical reactions. While ACES II can perform Hartree-Fock Self-Consistent-Field (HF-SCF) and Kohn-Sham Density Functional Theory (KS-DFT) calculations, the ACES II program system is not intended for large scale HF-SCF or KS-DFT calculations.

Two important features of the ACES II program system are its effective use of molecular symmetry, particularly in MBPT and CC calculations, and the sophisticated gradient methods which are included in the program. Even if a few elements of symmetry are present in a molecular system, differences in execution times required for calculations with symmetry and without can be dramatic. The implementation of symmetry currently is limited to D_{2h} and its subgroups, and the expected speedup due to symmetry utilization will be on the order of the square of the order of the computational point group for all steps except integral and integral derivative generation and integral sorts, in which the speedup can be no greater than the order of the group.

Gradient techniques are implemented for SCF and the following correlated levels of theory: MBPT(2), MBPT(3), MBPT(4), CCD, QCISD, CCSD, QCISD(T), and CCSD(T) for both restricted and unrestricted Hartree-Fock (RHF and UHF, respectively) reference functions. In addition, for the MBPT(2), MBPT(3), CCSD, and CCSD(T) methods, gradients are available for restricted open-shell Hartree-Fock (ROHF) reference functions. They are also available for certain CCSD calculations based on quasi-restricted Hartree-Fock (QRHF) reference functions, namely those for high-spin doublet cases and two-determinant CCSD (TD-CCSD) calculations for open-shell singlet states, in which the open-shell orbitals have different symmetries.

Efficient algorithms for geometry optimization and transition state searching have also been included, and may be used at all levels of theory. The analytical gradients employed during geometry optimizations and vibrational frequency calculations depend on their availability. When analytical gradients are not available, automated finite differencing procedures can be used to compute the derivatives. Analytic second derivatives have been implemented for SCF using RHF, UHF, and ROHF reference functions. In addition, analytically evaluated NMR chemical shift tensors are available at the SCF and MBPT(2) levels using

gauge-including atomic orbitals (GIAOs) to ensure exact gauge-invariance. Other features include the direct calculation of electronic excitation energies using the Tamm-Dancoff (or configuration interaction singles) model (CIS), the random-phase approximation (RPA), the equation-of-motion coupled-cluster approach (EOM-CC) and similarity transform-equation-of-motion (STEOM), and molecular ionization potentials and electron affinities with EOM, STEOM, and Fock-space coupled-cluster methods. Transition moments between ground and excited states can be calculated for all of the methods, as well as selected excited state properties. The excited state geometry optimization and frequency calculations employ the analytical gradients capabilities available for the EOM and STEOM methods.

The programs collectively known as ACES II began development in early 1990, and the first version of the code was written by J.F. Stanton, J. Gauss, J.D. Watts, W.J. Lauderdale, and R.J. Bartlett. Program development is continuing, and the capabilities as well as the contributors to the development of the ACES II program system are continually increasing. At present, there are more than 30 member executables, each of which performs a well-defined function and communicates with the rest of ACES II through stored files. The ACES II program system has been interfaced with external programs such as MOLCAS and GAMESS. The primary function of the MOLCAS and GAMESS interfaces is to provide integral and integral derivative programs that are more efficient and have direct capabilities to complement the functionalities of the locally modified version of VMOL, VPROPS, and VDINT programs. A complete replacement of these member executables is not yet feasible since the specialized integrals such as Gauge-origin-independent (GIAO) integrals and NMR spin-spin coupling operator integrals are only available in VMOL, VPROPS, and VDINT. ACES II can also generate interfaces to graphical programs such as MOLDEN and gOpenMol, wave function analysis codes such as Natural Bond Orbital (NBO), and semi-emperical programs such as HyperChem, NDDO, and MOPAC. The ACES II distribution comes with these capabilities; however, ACES II developers are not responsible for distributing or maintaining the external programs. Having independent licenses for external programs along with ACES II will allow users to take full advantage of this functionality.

Since ACES II is the product of academic research group, and not a software company, we are unable to guarantee that all results obtained with it are correct. Although we have made great progress in removing serious errors from the codes, problems may still occur and should be reported to `aces2@qtp.ufl.edu`. Any suggestions for improving the input or output, "wish-lists" for features, or other comments may also be sent to this address.

3.1 Overview of capabilities of ACES II

The general capabilities of ACES II to determine single point energies, analytical gradients, and analytical hessians are as follows:

Single point energy calculations:

- Independent particle models include RHF, UHF, and ROHF.
- Correlation methods utilizing RHF and UHF reference determinants include MBPT(2), MBPT(3), SDQ-MBPT(4), MBPT(4), CCD, CCSD, CCSD(T), CCSD+TQ*(CCSD), CCSD(TQ), CCSDT-1 CCSDT-2, CCSDT-3, QCISD, QCISD(T), QCISD(TQ), UCCS(4), UCCSD(4), CID, and CISD.
- Correlation methods that can use ROHF reference determinants include MBPT(2), CCSD, CCSDT, CCSD(T), CCSDT-1, CCSDT-2, and CCSDT-3.
- Correlation methods that can use QRHF or Brueckner orbital reference determinants include CCSD, CCSDT, CCSD(T), CCSDT-1, CCSDT-2, and CCSDT-3.
- Two-determinant CCSD calculations for open-shell singlet state.
- Equation-of-motion CCSD calculation of dynamic polarizabilities (including partitioned scheme)
- Equation-of-motion CCSD calculation of NMR spin-spin coupling constants (including partitioned scheme).
- Partitioned equation-of-motion CCSD calculations of excitation energies.
- Kohn-Sham DFT methods combined with a wide selection of density functionals.

Analytical gradients:

- Independent particle models include RHF, UHF, and ROHF.
- Correlation methods utilizing RHF and UHF reference determinants include MBPT(2), MBPT(3), SDQ-MBPT(4), MBPT(4), CCD, CCSD, CCSD+T(CCSD), CCSD(T), CCSDT-1, CCSDT-2, CCSDT-3, QCISD, QCISD(T), UCC(4), UCCSD(4), CID, and CISD.
- Correlation methods that can also utilize ROHF reference determinants include MBPT(2), CCSD, and CCSD(T).
- Correlation methods that can also utilize QRHF reference determinants include CCSD.
- Two-determinant CCSD calculations for open-shell singlet state based on QRHF orbitals.

- EOM-CCSD analytical gradients for excited states.
- TD-CCSD analytical derivatives.
- Dropped core and/or virtual orbitals in analytical derivative calculations for RHF, UHF, and ROHF references.

Analytical Hessians:

- Independent particle models include RHF, UHF, and ROHF.

4 Quickstart Guide

At the bare minimum, the user must provide an input file named `ZMAT` and a basis set file named `GENBAS`. The main executable `xaces2` is a driver program for other batch-based member executables such as `xjoda`, `xvscf`, and `xvcc`. Since `xaces2` uses the `system()` function to run these programs, the executable path must be set by the user's regular shell environment. For example, if `xaces2` does not appear in the default login environment, then running `xaces2` directly will likely result in error since the first attempt to run `xjoda` will fail with "`xjoda: not found`".

The ACES II program system uses many storage files, and the developers initially recommend running the program in a directory that contains only `ZMAT` and `GENBAS`. As experience with the program increases, users will become comfortable with naming files that do not conflict with those used by ACES II.

```
> ls
GENBAS
> cat <<EOF >ZMAT
H2
H
H 1 R

R=0.7356

*ACES2(BASIS=DZP,CALC=SCF)

EOF
> xaces2 > out
```

The file named `out` now contains the RHF SCF results of H_2 in the DZP basis set (provided `GENBAS` contains the DZP basis set for H). Since SCF is the default calculation level, the CALC keyword is not necessary. The minimum requirements for `ZMAT` are:

- 1-line title
- molecular system in internal or Cartesian coordinates
- blank line
- `*ACES2` namelist declaring a basis set to use
- blank line

For more information on `ZMAT`, `GENBAS`, and most of the other files used and created by ACES II, Section 6 (page 22) describes the overall list of files used by the program and Section 7 (page 26) shows the input formats for certain user files.

5 Program Structure

The ACES II program system is a collection of programs that work together to perform the user's calculation. An ACES Member Executable (AME) is referenced by the name of its source code (e.g., JODA) and the name of its binary executable (e.g., `xjoda`). Most users will only interact with the driver program `xaces2`, but it is strongly recommended that users familiarize themselves with `xjoda` since that program reads the input file and initializes the ACES II file set.

5.1 aces2 and p_aces2

`xaces2` is the main program that drives the ACES II program system. After an initial call to `xjoda`, it determines the proper calling sequence of programs based on the calculation level and various other keywords. In principle, this program is not necessary if the user knows the exact calling sequence of member executables (and the calculation does not involve dropped MO gradients).

`xp_aces2` is a parallel version of `xaces2` that should be used *ONLY* for calculations that perform numerical finite differences (geometry optimizations with GRAD_CALC= NUMERICAL or vibrational frequencies with VIB=FINDIF). See the examples of parallel calculations in Section 10.3 (page 96) for more information.

5.2 joda

Along with parsing `ZMAT`, building the keyword environment, and initializing the ACES II file set, JODA is responsible for everything between single point calculations. Geometry optimizations, numerical finite differences, and restart capabilities are all handled by this program.

5.3 mopac

ACES II has a modified version of MOPAC (version 5) that can generate an initial guess Hessian for geometry optimizations.

5.4 vmol

`xvmol` calculates the one- and two-electron AO integrals over Gaussian basis functions. VMOL was written by J. Amlöf and P.R. Taylor and was modified to include an option for effective core potentials.

5.5 vmol2ja

`xvmol2ja` creates most of the transformation matrices needed to switch between internal (VMOL) and external (`ZMAT`) ordering of atoms and atomic orbitals. It also creates the Cartesian-Spherical orbital transformations.

5.6 vprops

`xvprops` evaluates one-electron integrals needed for the calculation of various first-order properties such as dipole moment, quadrupole moment, electrical field gradients, or spin densities. It originates from POLYATOM and was interfaced to the VMOL integral program by P.R. Taylor.

5.7 nddo

`xnddo` calculates the NDDO density, which can be used as an initial guess for the SCF algorithm.

5.8 vscf, p_vscf, vscf_ks, intgrt, and intpack

These programs are responsible for generating the Hartree-Fock (`xvscf`) and Kohn-Sham (`xvscf_ks`) SCF reference wavefunctions. For KS-SCF, the numerical integrator `xintgrt` is used to calculate the functional energy and `xintpack` is used for OEP calculations.

The parallel HF-SCF program `xp_vscf` should be used for GAMESS direct integrals (FOCK=AO, DIRECT=ON, INTEGRALS=GAMESS).

5.9 dirmp2 and p_dirmp2

`xdirmp2` calculates the MBPT(2) energy using the GAMESS direct integral package. `xp_dirmp2` does this in parallel.

5.10 vtran

`xvtran` is responsible for the 4-index AO→MO integral transformations after the SCF calculation.

5.11 tdhf

`xtdhf` performs time-dependent Hartree-Fock calculations.

5.12 intprc

`xintprc` sorts the two-electron integrals into five basic types: OOOO, OOOV, OOVV, OVVV, and VVVV, in which O and V stand for occupied and virtual orbitals, respectively. It also calculates the MBPT(2) energy.

5.13 vcc, vcc5t, and vcc5q

These programs calculate the CC energy by solving the T amplitude equations and calculating all non-iterative contributions. `xvcc` also calculates finite-order perturbation theory energies by manipulating the CC iteration logic.

5.14 mrcc

The `xmrcc` program uses a different programming environment than the rest of ACES II. This program implements many EOM-related excited-state theories like IP-EOM, DIP-EOM, STEOM, etc.

5.15 fno

`xfno` rotates the orbitals of the reference wavefunction to frozen natural orbitals for the correlation corrections.

5.16 lambda

`xlambda` solves the Λ equations to determine the response of the CC amplitudes to a given perturbation.

5.17 vea and vee

`xvea` calculates electron attachment energies by the EOM-CC method. `xvee` calculates excitation energies, transition moments, and excited state density matrices for TDA EOM-CC methods. Unlike MRCC, they both use the standard ACES II programming environment.

5.18 vcceh

`xvcceh` calculates EOM-CCSD polarizability and NMR spin-spin coupling constants.

5.19 dens

`xdens` calculates the one- and two-particle correlated density matrices in the MO basis.

5.20 props

`xprops` computes all of the first-order properties (dipole moments, electric field gradients, electric quadrupole moments, electrostatic potentials, spin densities for open-shell molecules, etc.). It also computes the scalar relativistic corrections and the Mulliken population analysis.

5.21 anti

`xanti` sorts and de-antisymmetrizes the two-particle density matrix.

5.22 bcktrn

`xbcktrn` performs the MO→AO transformation of the density matrices for direct contraction with the integral derivatives in the AO basis.

5.23 vdint, vksdint, scfgrd, and p_scfgrd

VDINT is a heavily modified version of the integral derivative program ABACUS written by T. Helgaker, P. Jørgensen, H. Aa. Jensen, and P.R. Taylor, suitable for CC/MBPT gradient calculations. In addition to integral derivatives with respect to geometrical perturbations it calculates one- and two-electron integrals required for chemical shift calculations within the GIAO scheme. For the most part, `xvdint` calculates the gradient in ACES II. For KS reference wavefunctions, `xvksdint` calculates the contribution from the functional derivative. `xscfgrd` calculates the HF-SCF gradient using the GAMESS direct integral-derivative package, and `xp_scfgrd` does this in parallel.

5.24 cphf

`xcphf` solves the coupled-perturbed Hartree-Fock equations either for geometric displacements or for electric or magnetic field components as perturbations.

5.25 nmr

`xnmr` calculates the paramagnetic contribution to NMR chemical shifts at correlated levels.

5.26 asv

An ACES State Variable (ASV) is a runtime variable that controls the calculation, and users can affect ASV initialization with keywords in the `*ACES2` namelist. `xasv` is not a member executable that other AMEs use but rather a tool for the user to examine the validity of keyword/value pairs. For example, if a user wants to check if CALC=LCCD is valid, then the user would have to create a valid `ZMAT` file and run `xjoda` to guarantee it passes keyword parsing. Alternatively, the user can run "`xasv CALC=LCCD`" at the shell prompt without the hassle of managing files.

5.27 a2proc

This program was initially created to reduce the clutter of member executables in the ACES II program system. Its two main purposes are to gather many small, single-use programs and to provide interfaces to external programs like Molden and HyperChem. "`xa2proc help`" will show the list of available modules and the arguments that each one expects.

5.27.1 clrdirty

During an optimization or frequency calculation with RESTART=ON (the default), the ACES II file set is tagged with a dirty flag. Immediately before a call to `xjoda`, `xaces2` will clear the dirty flag thus signaling `xjoda` to backup the files. If the dirty flag is not clear, then `xjoda` will assume the calculation has crashed and restore the previous file set instead of saving the current set. Users *must* clear the dirty flag manually with "`xa2proc clrdirty`" if they are running each AME separately; otherwise, ACES II will loop over the same geometry forever (it will not even increment the step counter and stop after a certain number of "steps").

5.27.2 mem

A user can alter the MEMORY_SIZE state variable of a *STATIC* ACES II file set with the MEM module. If no AMEs are using the `JOBARC` and `JAINDX` files, then "`xa2proc mem` ***`amount`***" will change the value that each AME uses to allocate memory. This change will remain in effect until the next run of `xjoda`, which will reset it to whatever value is in the `ZMAT` file. *amount* is a double-precision number optionally followed by a unit. Valid, case-insensitive units are B, KB, MB, GB, W, KW, MW, and GW. The number and units must be one string (no spaces).

For example, a user might be nursing a large calculation by running each executable by hand (and backing up the files between them). If he or she discovers ACES needs more memory, then the user changes the MEMSIZE value in `ZMAT` and runs `xa2proc mem` on the backup files.

5.27.3 zerorec

This module flushes records in the `JOBARC` file with zeroes.

5.27.4 rmfiles

The RMFILES module will delete the five list storage files `MOINTS`, `GAMLAM`, `MOABCD`, `DERINT`, and `DERGAM`. More importantly, it will reset the appropriate pointers and counters so the next AME that attempts to initialize the I/O subsystem will not crash.

5.27.5 parfd

The PARFD module is used to export and import finite difference information. It was created as a proof-of-concept program to demonstrate JODA's ability to operate in a parallel finite difference calculation. Its main capabilities are incrementing the displacement data (`parfd update`), dumping the data to standard output (`parfd dump`), and importing data from other calculations (`parfd load` *`file`*). An example of manual parallel finite differences can be found in Section 10.3.2 (page 97).

5.27.6 molden and hyperchem

These modules create Molden and HyperChem input files, respectively, that contain geometry, wavefunction, and vibrational frequency information.

5.27.7 jasum and iosum

These modules print summary information of the `JOBARC` records and storage lists, respectively.

5.27.8 jarec

JAREC (and its quiet variant JAREQ) will show the formatted contents of a `JOBARC` record. It requires three arguments: data type, record name (case sensitive), and dimensions. Data types can be i (integer), d (double), f (float), r (real), ad (array of doubles), and ai (array of integers). The first four types are all one-dimensional vectors, and ad and ai are two-dimensional arrays. For the arrays, the dimension string can be of the form "R,C", "RxC",

and "RXC", in which R is the number of rows and C is the number of columns. The following results were taken after an H_2O_2 DZP SCF geometry optimization:

```
> xa2proc jareq i NATOMS 1
 4
> xa2proc jareq d TOTENERG 1
 -0.150816196754E+03
> xa2proc jareq ad GRADIENT 3x4
    -0.000079292458     0.000079292458     0.000034723378    -0.000034723378
     0.000006762679    -0.000006762679     0.000088780697    -0.000088780697
     0.000000000000     0.000000000000     0.000000000000     0.000000000000
```

5.27.9 xyz

This module prints the Cartesian coordinates of the current geometry. This could be used in a script that automatically runs a vibrational frequency calculation after a geometry optimization.

5.27.10 test

The TEST module is used by the automated regression test suite. Its use is beyond the scope of this manual, but interested users are encouraged to examine the files in the ACES II test directory.

5.28 gemini

`xgemini` is used to manage scratch directories in parallel calculations. Parallel AMEs `xp_aces2` (finite differences), `xp_vscf`, `xp_scfgrd` (SCF energies and gradients), and `xp_dirmp2` (MBPT(2) energies) all operate under the premise that each MPI task has its own ACES II file set to modify. To prevent the tasks from clobbering each other's files, `xgemini` can create, destroy, and manipulate private scratch directories for each task.

6 File Structure

6.1 ZMAT

ZMAT is the primary user interface to ACES II, and it must exist in the run directory.

6.2 GENBAS/ZMAT.BAS

The files named GENBAS and ZMAT.BAS contain the basis set definitions that the program can use. In practice, GENBAS is a large file, and xjoda can spend most of its time scanning the file for the basis set definitions. ZMAT.BAS is created by xjoda to cache the relevant basis sets from GENBAS; in other words, if xjoda sees ZMAT.BAS, then it will try to read the basis information from there. If a definition is missing from ZMAT.BAS, then xjoda will crash just as if GENBAS was missing the definition. Appendix B.1 (page 115) lists the contents of the standard GENBAS file.

6.3 ECPDATA

This file contains the data for effective core potentials. Standard sets can be found in Appendix B.2 (page 131).

6.4 GUESS

The GUESS file is used to control the "placement" of electrons in the SCF initial guess and can be used only with GUESS=READ_SO_MOS (i.e., initial orbitals are read from OLDMOS).

6.5 NEWMOS/OLDMOS

These files contain the MOs (in the symmetry-adapted AO basis) of the SCF wavefunction. NEWMOS is created by the SCF program after the last iteration, and the user can copy this file to OLDMOS to initialize the MOs of a later calculation on the same molecule and basis set. The example in Section 9.1.6 (page 72) illustrates this capability.

6.6 AOBASMOS/OLDAOMOS

These files contain the MOs in the AO basis before symmetry adaptation. AOBASMOS and OLDAOMOS work the same as NEWMOS and OLDMOS, except that OLDAOMOS can be used to initialize SCF orbitals in vibrational frequency calculations, in which the point group symmetry could change for each displacement.

6.7 FCMINT, FCM, FCMSCR, and FCMFINAL

FCMINT contains the full internal coordinate force constant matrix. The other files, FCM, FCMSCR, and FCMFINAL, correspond to the symmetrized, mass-weighted, and analytical force constant matrices, respectively. The example in Section 9.2.6 (page 88) shows how FCMINT can be used to initialize the Hessian matrix in a geometry search.

6.8 frequency

This file is a simple ASCII free-format file that specifies the frequency or frequencies at which dynamic polarizabilities are computed.

6.9 ISOMASS

Vibrational frequencies can be calculated with standard atomic masses or user-supplied masses—usually of isotopes. If a file named ISOMASS is found, then the vibrational frequency logic in JODA replaces the atomic masses with those found in the file. The content is free format ASCII, and the order of the masses must match the non-dummy centers in ZMAT.

6.10 System files

6.10.1 JOBARC, JAINDX

The JOBARC file stores records (named arrays) for the ACES II program system. The accompanying file JAINDX stores metadata about the records.

6.10.2 MOINTS, GAMLAM, MOABCD, DERINT, DERGAM

These files store lists of double-precision arrays used by all of the post-SCF member executables. *They will be VERY large for big molecules and basis sets.*

6.10.3 MOL, IIII, IIJJ, IJIJ, IJKL

The MOL file is created by xjoda and stores the molecular system and basis set information used by xvmol and any AME that uses GAMESS integrals. The IIII file stores the one-electron integrals and all totally symmetric two-electron integrals in the AO basis. *This file will be very large for big basis sets.* The other three files, IIJJ, IJIJ, and IJKL store two-electron AO integrals that are not totally symmetric.

6.10.4 OPTARC/OPTARCBK

The iteration history of geometry optimizations is stored in OPTARC. OPTARCBK is a backup of the true OPTARC file, which gets clobbered during geometry optimizations with numerical gradients.

6.10.5 DIPOL, DIPDER, POLAR, POLDER

DIPOL and POLAR contain the dipole moments and polarizabilities, respectively, and DIPDER and POLDER contain their derivatives.

6.10.6 GRD

This file was intended for programs to extract gradients from ACES II. With the functionality in the ACESCORE library and xa2proc, this file interface is obsolete.

6.10.7 HF2, HF2AA, HF2AB, HF2BB

These files are created by xvtran and store partially-transformed two-electron integrals for use in xintprc. Usually they are deleted by xintprc unless a particular post-SCF option requires them (such as ABCDTYPE=MULTIPASS).

6.10.8 IUHF

This file contains the RHF/UHF flag. Its use is limited and it should disappear in a future release.

6.10.9 TGUESS, LGUESS

These files store the coupled-cluster T and Λ amplitudes, respectively, for restart purposes.

6.10.10 VPOUT

This file contains the first-order property integrals.

6.10.11 GAMESS.LOG, MP2.LOG, DIRGRD.LOG

All of these files are generated by the GAMESS interface in VSCF, DIRMP2, and SCFGRD. They are strictly log files and might contain error messages if a program crashes.

6.10.12 `OUT.000, DUMP.000, 1ELGRAD.000`

All of these files are generated by the GAMESS interface in VSCF, DIRMP2, and SCFGRD and are tagged with the MPI rank of each process. They are strictly output files and might contain error messages if a program crashes.

7 File Formats

7.1 ZMAT

7.1.1 File anatomy

This specification pertains to a particular version of JODA. Older versions might require more rigid input formats, but the general structure of `ZMAT` should never change. Every line in `ZMAT` can be at most 80 characters long. The parser does not check the length of each line; therefore, there are no guarantees that the rules of `ZMAT` parsing will apply once the line length has been exceeded.

1. Header

   Three types of lines are allowed in the header:

   - blank space (consisting of spaces and/or tabs)
   - comments (first non-blank character is a hash mark, #)
   - file directives (first non-blank character is a percent sign, %)

   The header can have an arbitrary number of lines, but the first line that does not qualify as one of these three will be read as the job title.

2. Job title

   The first non-blank, non-comment, non-directive line is the job title. It may consist of any character and may be at most 80 characters long.

3. Molecular system

   The coordinate matrices must immediately follow the job title. Every line must describe one atom (dummy or otherwise). The first blank or comment line terminates the coordinate matrix. Atom descriptions may have hash-delimited comments and the coordinate strings may be separated with spaces or tabs. There are two types of coordinate matrices: internal (Z matrix) and Cartesian.

   Internal coordinate matrices have two parts separated by a blank line: the Z matrix and the parameter definitions.

   Cartesian coordinate matrices are the standard X Y Z format with no blank lines.

   For LST and QST geometry search algorithms, multiple XYZ and Z matrix *parameter* matrices can be supplied in the following order: INITIAL-[TRANSITION]-FINAL. The transition geometry is only used for QST. Supplying a transition geometry for

LST will yield incorrect results since the parser will read the second set of coordinates as the final geometry.

4. Namelists

   Any number of blank lines may pad the namelists. In general, a namelist is delimited by an asterisk immediately followed by the case-sensitive name of the namelist ('`*ACES2`', '`*INTGRT`', ...). Only the `*ACES2` namelist is required for every calculation. Some programs or features recognize other namelists, but the rules for parsing the strings are specific for each list. Rules for parsing the `*ACES2` namelist can be found further down.

5. Line-item basis set and ECP data assignments

   If either BASIS=SPECIAL (the default) or ECP=ON (off by default), then there must be one blank line between the `*ACES2` namelist and the assignment block. If both blocks are required, then there must be one blank line between the `*ACES2` namelist and the basis set block followed by another blank line and the ECP block.

6. Footer

   Any unrecognized text (non-namelist) following the internal coordinate definitions is ignored. If there is no footer, then ***ZMAT must terminate with a blank line!***

7.1.2 Examples

```
Line 1   RHF CCSD property calc of C6 in DZP
Line 2   X
Line 3   X 1 RX
Line 4   C 2 R 1 A
Line 5   C 2 R 1 A 3 T
Line 6   C 2 R 1 A 4 T
Line 7   C 2 R 1 A 5 T
Line 8   C 2 R 1 A 6 T
Line 9   C 2 R 1 A 7 T
Line 10
Line 11  RX =  1.0
Line 12  R  =  1.33
Line 13  A  = 90.
Line 14  T  = 60.
Line 15
Line 16  *ACES2(CALC=CCSD,PROPS=FIRST_ORDER,BASIS=DZP)
Line 17
```

```
Line 1  RHF MBPT(2) property calc of Formaldehyde in DZP
Line 2  O 0.0      0.0          1.22
Line 3  C 0.0      0.0          0.0
Line 4  H 0.0      0.873489539  0.545816842
Line 5  H 0.0     -0.873489539 -0.545816842
Line 6
Line 7  *ACES2(CALC=MBPT(2),PROPS=FIRST_ORDER,BASIS=DZP)
Line 8
```

7.1.3 File directives

Users can control the locations of files with file directives in the header. By default, all files used by ACES II (`JOBARC`, `JAINDX`, `IIII`, `MOINTS`, etc.) are kept in the directory in which the `xaces2` program is invoked. Any file except `ZMAT` may be relocated with:

```
% FILE=/some/absolute/path/to/the/FILE
```

in which `FILE` is the name of the particular file. For example, if a user wants to calculate isotopic shifts after a vibrational frequency calculation, it is useful to keep `JOBARC` and `JAINDX` in a safe place (not in the current directory). For water, the following input could be used:

```
Line 1   %JOBARC=/u/usr/safe/JOBARC
Line 2   %JAINDX=/u/usr/safe/JAINDX
Line 3
Line 4   RHF SCF vib freq calc for Water in DZP
Line 5   H
Line 6   O 1 R
Line 7   H 2 R 1 A
Line 8
Line 9   R=0.957
Line 10  A=104.51
Line 11
Line 12  *ACES2
Line 13  VIB=EXACT,BASIS=DZP
Line 14
```

After this job executes, the files `JOBARC` and `JAINDX` are stored in the directory `/u/usr/safe`. Then the user can simply change to this directory, create the appropriate `ISOMASS` file, and run `xjoda` directly.

Coarse-grain restart capabilities depend on a "file" named `SAVEDIR`. The path variable is the full name of the directory to use for archiving.

```
% SAVEDIR = /u/home/yau/RESTART
```

This *MUST* be unique for all jobs running simultaneously. If multiple jobs are using the same save directory, then they are clobbering each other's files and the last job to archive

its files is the one that wins. The default action is to create a directory named `SAVEDIR` in the run directory.

7.1.4 Molecular orientation

The orientation of the molecule in Cartesian space is related to its point group. Two orientations are used extensively in the ACES II program system: the "standard" or "computational" orientation, which is a standard orientation for the computational point group, and a "canonical" orientation, which is the standard orientation for the full point group. Note that in some cases, the two orientations are identical. All calculations are performed in the computational orientation; therefore, orbital symmetries should be specified according to this Cartesian axis system. For the most part, the canonical orientation is used internally for tasks such as determining irreducible representations or other properties related to the full point group. The standard orientation for each point group follows

C_N**:** Rotation axis along z.

D_N**:** Rotation axes coincident with Cartesian axes and the highest order axis along z.

C_s**:** Plane of symmetry is xy.

S_N**:** S_N axis along z.

C_{Nv}**:** C_N axis along z, xz is σ_v.

C_{Nh}**:** C_N axis along z, xy is σ_h.

D_{Nh}**:** C_N axis along z, one C_2 axis along x.

D_{Nd}**:** S_{2N} axis along z, one C_2 along x.

T**:** C_3 axis along (q, q, q).

T_d**:** S_4 axis along z.

T_h**:** C_3 axis along (q, q, q), symmetry planes are xy, xz, and yz.

O**:** C_4 axes along x, y, and z.

O_h**:** C_4 axes along x, y, and z.

I**:** C_5 axis along z, one C_3 lies in the xz plane.

I_h**:** C_5 axis along z, xz is a symmetry plane.

For groups with ambiguities (D_2, D_{2h}, C_{2v}), there is no standard orientation at present, and users might want to run `xjoda` once to show which orientation is used before assigning orbital symmetries. For example, water belongs to the C_{2v} point group and the symmetry plane containing the hydrogen atoms might be assigned to either the xz or yz planes, leading to an ambiguity between the b_1 and b_2 irreducible representations. Eventually, some criterion could be established that can define a standard orientation for these groups thereby alleviating this issue.

7.1.5 Dummy and ghost atoms

Dummy atoms, represented with "X", are only useful for internal coordinates and define points in space. They do not have basis functions and do not affect the symmetry of the molecule. They are necessary to break angles of 180° and are used to define highly symmetric molecules with no atom at the center of mass.

Ghost atoms, which are specified by the symbol "GH", have zero nuclear charge. However, while dummy atoms are "invisible" to the program outside the coordinate generator, ghost atoms serve as a center for basis functions. This feature is particularly useful for calculations that determine the basis set superposition error (BSSE) and has several other applications, such as describing "lone-pair" electrons of a molecule by functions that are not centered at any of the molecular nuclei. Symmetry can be used in such calculations but is restricted to the symmetry of the "supermolecule" comprised of the real and ghost atoms. The additional basis functions do not necessarily form a complete set of symmetry-adapted functions within the point group of the supermolecule. This is different from the use of dummy atoms, which do not affect the symmetry of the calculation.

Currently, only single point energy calculations are possible with ghost atoms. In addition, the basis set must be supplied explicitly with BASIS=SPECIAL and the line-item basis set definitions after the `*ACES2` namelist.

7.1.6 Cartesian coordinates

The format is straightforward. Each line defines one atom with the atomic symbol and the values of the x, y, and z coordinates in free format. The coordinates may be given in either atomic units or Ångströms.

Older versions of `xjoda` require COORDINATE=CARTESIAN for this to work, but any version *after* 2.5.0 attempts to figure it out automatically (since the first line of a Z matrix has only one word). If the Cartesian coordinates are specified in atomic units, then the keyword UNITS=BOHR must be used.

7.1.7 Internal coordinates

The specification by internal coordinates is known as the Z matrix. Centers of the nuclei are expressed relative to previously defined centers by means of distances and angles. The specification includes a length, a bond angle, and a dihedral angle. The number associated with each atom is governed by its position in the Z matrix. The essentials of Z matrix construction can be illustrated by considering a Z matrix for a system of four atoms ABCD, not linked in any particular order.

```
Arbitrary ABCD molecule
A
B 1 AB
C 1 AC 2 CAB
D 3 CD 2 DCB 1 TAU
```

The first line in the Z matrix contains the atomic symbol of one of the atoms, say A. The second line specifies the position of a second atom, say B, relative to the first atom. Suppose that B is a distance AB from A. The second line then contains the atomic symbol B, followed by the number 1 (A is atom number 1), and a parameter label, AB, ("B 1 AB"). For the specification of the third atom, a distance and an angle are needed. We may use the distance between atoms A and C and the angle CAB, or we may use the distance between atoms B and C and the angle CBA. In the first case the third line would have the form "C 1 AC 2 CAB", while in the second case it would have the form "C 2 BC 1 CBA". Finally, there is a line specifying the position of D relative to the other atoms. This line must contain a distance, a bond angle, and a dihedral angle and could have the form "D 3 CD 2 DCB 1 TAU", with TAU being the angle between the BCD and ABC planes.

For a system with more than four atoms, the fifth and subsequent lines follow the same pattern as the fourth line of the example given above (i.e., they also contain a length, angle, and dihedral angle and the numbers of three previously specified centers). It should be emphasized that this is a somewhat simplified description that would work for a tetra-atomic molecule such as hydrogen peroxide but would not be satisfactory for acetylene. The latter requires "dummy" atoms. These and several other tips for forming Z matrices are discussed below.

A more formal description of a line in the Z matrix input is as follows. Each line may have as many as seven entries. We consider the I^{th} line. The contents are the I^{th} element of the ZSYM array, the $3 * I^{th}$, $(3 * I - 1)^{th}$, and $(3 * I - 2)^{th}$ elements of the NCON and PARNAM arrays. The ZSYM array is of length N, where N is the number of lines in the Z matrix (this includes those for any dummy and ghost atoms), and contains the chemical symbols of all the atoms in the Z matrix. The NCON and PARNAM arrays are of length 3*N. NCON contains the numbers of atoms relative to which each atom is specified. The PARNAM array contains the names of the lengths, angles, and dihedral angles contained in

the Z matrix. Positions $3*I-2$ are for lengths, $3*I-1$ for angles, and $3*I$ are for dihedral angles (I=1,2,...,N). It should be clear that elements 1, 2, 3, and 5, 6, and 9 of NCON and PARNAM are not defined. The I^{th} line (I=1,2,...,N) then has the general form:

ZSYM The atomic symbol of the atom (dummy atom is "X", ghost is "GH").

NCON(3*I-2) The number of the atomic center to which the atom is formally linked. *Note that this need only be a formal link, the two atoms need not be chemically bonded.* (Only for I>1).

PARNAM(3*I-2) A character variable name corresponding to the distance between atom NCON(3*I-2) and atom I.

NCON(3*I-1) The third member of the triangle formed by atoms I and NCON(3*I-2). (Only for I>2).

PARNAM(3*I-1) A character variable corresponding to the associated angle.

NCON(3*I) The fourth member of the dihedral angle formed by atoms I, NCON(3*I-2), and NCON(3*I-1). (Only for I>3).

PARNAM(3*I) A character variable name corresponding to a dihedral angle determined as follows: In the plane perpendicular to the NCON(3*I-2)↔NCON(3*I-1) axis, the angle is that needed to rotate the projection of the [I←NCON(3*I-2)] vector into the projection of the [NCON(3*I)←NCON(3*I-1)] vector. Clockwise is taken to be positive. Values must be restricted from -180 to 180°.

All variable names (PARNAM array) are limited to *five* characters. An asterisk (∗) immediately after the variable name implies an optimization. *No numbers are allowed inside the Z matrix—all internal coordinates must be given a symbol, even if the value is not going to be optimized.*

Z matrix Parameters

After a blank line following the Z matrix, the values of all *unique* internal coordinates (those with different names) are specified as follows:

PNM=Value

where PNM is one of the unique members of the PARNAM array (see above), and Value is the value assigned to that coordinate. The first non-blank string after the equal sign is passed to a number parser, so trailing text (like a comment) is ignored.

Angles must be entered in degrees (°), and bond angles (as distinct from dihedral angles) of 0° and 180° are not allowed since these lead to a singularity in the transformation between

Cartesian and internal coordinates (and do not allow dihedral angles to be defined). This, of course, does not mean that ACES II is unable to handle linear molecules such as carbon dioxide. Rather, for linear molecules, "dummy atoms" must be used in the Z matrix to avoid problematic bond angles.

To facilitate construction of Z matrices for highly symmetric molecules, certain variable names have been reserved for specific values. To use these parameters, which are listed below, the user must specify a value in the parameter input section, but it need not be correct since it will be converted to the exact value internally.

TDA Specifies the tetrahedral angle, $\mathrm{Cos}^{-1}\left(\frac{-1}{3}\right)$ (i.e., 109.4712...).

IHA Specifies the "icosahedral" angle $\mathrm{Cos}^{-1}\left(\sqrt{\frac{1}{5}}\right)$ (i.e., 63.4349...).

7.1.8 Z matrix analyzer

A unique feature of ACES II is the Z matrix analyzer, which is capable of detecting subtle and obvious deficiencies in the definition of internal coordinates. This is particularly important for geometry optimizations, in which the construction of the Z matrix and the choice of parameters to be optimized are of vital importance. The analyzer inspects the internal coordinates and carries out a number of checks. These include determination of whether coordinates given the same name are actually equivalent or coordinates having different names are equivalent, whether a non-zero gradient is possible with respect to modes which are not being optimized, etc. In addition, it determines the number of degrees of freedom within the totally symmetric subspaces of nuclear configurations and compares this value with the number of independent coordinates which are being optimized. If they are not equal, a warning message is printed out.

For most Z matrices with poorly defined internal coordinates, the analyzer prints out a number of warning messages but does not halt the ACES II execution sequence. However, for Z matrices which are particularly bad, it will terminate the job. All users are encouraged to carefully inspect the output of the analyzer and to check that the full molecular point group (printed out below the output from the analyzer) is the one intended. If warning messages are printed or if the symmetry is not what the user expects, then reconstruct as necessary.

A rule of thumb for Z matrix construction is that each internal coordinate included in the Z matrix must be accompanied by all others which are equivalent to it by the symmetry of the molecule. For example, in water, it is best to specify the molecular geometry by the two O–H distances and the H–O–H bond angle, rather than by the O–H distance, the H–H distance, and the H–H–O angle. Although most users would definitely use the first Z matrix, the idea of using just chemical bonds as internuclear distances can be dangerous.

For example, in a regular hexagonal ring, a little reflection will show that one cannot include *all six* inter-vertex distances in the Z matrix. If only five are specified, such as in the Z matrix below (the 1–6 distance is missing), the internal coordinate gradient cannot have the full symmetry of the molecule, and the first step of a geometry optimization will break the molecular symmetry. This results in extremely slow convergence and significantly increased CPU time due to the reduced molecular symmetry.

```
A poor Z matrix for hexagonal C6
C
C 1 R
C 2 R 1 A
C 3 R 2 A 1 T
C 4 R 3 A 2 T
C 5 R 4 A 3 T

R=1.33
A=120.
T=0.
```

In these situations, it is always best to use dummy atoms, which are merely mechanisms to reference a point in space for other coordinates. One or more dummy atoms are needed in essentially all Z matrices for molecules with high symmetry. As an example of their use, a "good" Z matrix for the C_6 ring is shown below. Don't be afraid to use dummy atoms!

```
A better Z matrix for hexagonal C6
X
X 1 RX
C 2 R 1 A
C 2 R 1 A 3 T
C 2 R 1 A 4 T
C 2 R 1 A 5 T
C 2 R 1 A 6 T
C 2 R 1 A 7 T

RX=1.0
R=1.33
A=90.
T=60.
```

7.1.9 *ACES2 namelist

The *ACES2 namelist is a keyword/value pair listing with a few tweaks. Every keyword has an internally declared type, which controls what form the value word(s) may take. Currently, the known types are: handle, string, long integer, and double.

Handles are strings that map to an integer. An example of this is the CALC state variable. When the parser reads 'CALC=SCF', it scans the CALC lookup table for the case-insensitive string that matches 'SCF'. Upon finding a match, the value of CALC is set to the offset of 'SCF' in the table. In this case, CALC is set to 0 (zero) since SCF is

the first element. Alternatively, the namelist could have read 'CALC=0' and the effect would be the same.

Some keys take switch values (ON and OFF). For these cases, the keyword may be specified without a value and the parser will assume the value is 'ON'. Similarly, the negation operator '!' may be used to turn the value to 'OFF'. An example is 'sym,!ecp' meaning 'SYM=ON,ECP=OFF'.

Strings are character arrays that are treated differently depending on the keyword they define. There are two types: plain text strings and array strings.

plain text strings are only used by the BASIS keyword (currently). The exact, case-sensitive value string is used to find the basis set definition in `GENBAS`.

array strings

matrices are used by the keywords OCCUPATION, IP_SYM, EA_SYM, etc. They are loosely defined as irrep-by-spin. This means the parser expects to find 'spin' columns of 'nirrep' rows. The row delimiter is a dash and the column delimiter is a forward slash.
Here are some examples:

- 4 irreps, 2 spins: 'occ=1-1-2-2/1-1-2-2'
- 2 irreps, 1 spin: 'ip_sym=1-0'
- 8 irreps, 3 spin pairs:
  'ee_sym=1-1-1-1-0-0-0-0/1-1-1-1-0-0-0-0/1-1-1-1-0-0-0-0'

Sets are merely one-dimensional arrays of values. The set delimiters are the same as the matrix ones except that the dash specifies a range of values and the forward slash separates single values. Currently, the only keywords that accept this type of string are: DROPMO, FD_IRREPS, ESTATE_SYM, QRHF_GEN, QRHF_ORB, and QRHF_SPIN.
Here are some examples:

- drop orbitals 1, 2, 3, 10, 11, and 12: 'dropmo=1-3/10-12'
- compute frequencies of modes that transform as irreps 1, 3, and 4:
  'fd_irreps=1/3/4'
- add an electron to the third lowest virtual of irrep 2 and remove an electron from the highest occupied of irrep 4: 'qrhf_gen=2/(-4),qrhf_orb=3/1'

NOTE: This syntax is *very* different from previous versions. Some value strings may be allowed by any version but might mean entirely different things. For example, 'DROPMO=1-31' used to mean dropping orbitals 1

and 31 from the correlated calculation. If that string was parsed by the new `xjoda`, VTRAN would attempt to remove every orbital from 1 to 31.

(long) Integers are parsed in a fairly straightforward manner. For example, 'print=1' and 'charge=-1'.

There is one special state variable that recognizes units appended to the value and that is MEMORY. Recognized units are (b)ytes and (w)ords with SI prefixes (k)ilo, (m)ega, and (g)iga. The scaling factor is 1024, not 1000.

Doubles are not used currently even though we have this capability, there are no keywords with values of this type. Examples would be: 'double1=1.', 'double2=-2.d0', 'double3=3.5e-6'.

Version 2.3 introduced environment variable awareness for keyword values. It is now possible to enter a value as "${*VARNAME*}" and have `xjoda` pull the value from the shell environment. The most practical application of this would be to loop over variables in a shell script that define various keywords. Here is an example:

```
> cat <<EOF >ZMAT
envvar test job
H
H 1 R

R=0.7

*ACES2
calc=${CALC}
basis=${BASIS}

EOF
> for CALC in scf ccsd
> do export CALC
>    for BASIS in DZP TZP TZ2P
>    do export BASIS
>       clean # or other cleaning script
>       xaces2 > $CALC.$BASIS.out
>    done
> done
> clean
```

7.1.10 Line-item basis/ECP definitions

If the BASIS keyword is set to SPECIAL (the default), then the basis set specification will be read directly after the `*ACES2` keyword list. One blank line must separate the last line of the keyword list from the beginning of the basis set input section. Each entry must be placed on an individual line, and the ordering of atoms must follow the Z matrix ordering *exactly*. The names will then be used to find the definitions in the basis set file (either `GENBAS` or `ZMAT.BAS`) in the current directory. If a basis set is not found, ACES II exits with an error condition. ACES II does not check to make sure that the atom to which the basis set belongs corresponds to the atomic designation in the corresponding row of the Z matrix. No entries are made for dummy atoms.

The format of the basis set names in `GENBAS` is: "XX:BASNAM", where XX is the atomic symbol of the atom (in capital letters), and BASNAM is the name of the basis. For new users, it is probably best to search `GENBAS` for "XX:" (XX being the atomic symbol), since this will show all of the available basis sets for that atom. A description of the format of `GENBAS` and its contents are given in the next section.

7.2 GENBAS/ZMAT.BAS

The following fixed format is used to store basis sets in the `GENBAS` file. Note that lines 5, 7, and 8 define numbers of shells (`NS`), contractions (`NC`), and exponents (`NE`).

Line	Fortran format	Description
1	A80	blank line
2	A80	name of the basis set
3	A80	comment line
4	A80	blank line
5	I3	the number of shells in the basis set (NS)
6	NS(I5)	angular momentum for each shell (L)
7	NS(I5)	number of contracted basis functions for each shell (NC)
8	NS(I5)	number of exponents for that shell (NE)
9	A80	blank line
10	NE(F14.7)	exponents for the first shell
11	A80	blank line
12	NC(F10.7,1X)	contraction coefficients for the first shell
13	A80	blank line

It is *necessary* that the shells are grouped by angular momentum, with the *s* shell(s) first, followed by *p* shell(s), etc.; otherwise, the input file written for the VMOL integral program will be incorrect. Lines 10 through 13 repeat NS number of times. An example of the boron PVTZ basis set entry is included below.

```
B:PVTZ
JFS DUNNING CORRELATION CONSISTENT BASIS FROM FTP

  4
    0    1    2    3
    4    3    2    1
   10    5    2    1

  5473.0000000   820.9000000   186.8000000    52.8300000    17.0800000
     5.9990000     2.2080000     0.5879000     0.2415000     0.0861000

 0.0005550 -0.0001120  0.0000000  0.0000000
 0.0042910 -0.0008680  0.0000000  0.0000000
 0.0219490 -0.0044840  0.0000000  0.0000000
 0.0844410 -0.0176830  0.0000000  0.0000000
 0.2385570 -0.0536390  0.0000000  0.0000000
 0.4350720 -0.1190050  0.0000000  0.0000000
 0.3419550 -0.1658240  0.0000000  0.0000000
 0.0368560  0.1201070  1.0000000  0.0000000
-0.0095450  0.5959810  0.0000000  0.0000000
 0.0023680  0.4110210  0.0000000  1.0000000

    12.0500000     2.6130000     0.7475000     0.2385000     0.0769800

 0.0131180  0.0000000  0.0000000
 0.0798960  0.0000000  0.0000000
 0.2772750  0.0000000  0.0000000
 0.5042700  1.0000000  0.0000000
 0.3536800  0.0000000  1.0000000

     0.6610000     0.1990000

 1.0000000  0.0000000
 0.0000000  1.0000000
```

```
     0.4900000

 1.0000000
```

7.3 ECPDATA

The parameters of the effective core potentials are specified in a format similar to GENBAS. For example, the entry for copper is given below.

```
*
CU:ECP-10-SK
# ECP BY STEVENS/KRAUSS FOR CU - 10 CORE ELECTRONS - LMAX = 2
*
    NCORE = 10    LMAX = 2
d
  -10.00000000    1  511.9951763
  -72.55482820    2   93.2801074
  -12.74502310    2   23.2206669
s-d
    3.00000000    0  173.1180854
   23.83518250    1  185.2419886
  473.89304880    2   73.1517847
  157.63458230    2   14.6884157
p-d
    5.00000000    0  100.7191369
    6.49909360    1  130.8345665
  351.46053950    2   53.8683720
   85.50160360    2   14.0989469
*
CU:ECP-18-SK
# ECP BY STEVENS/KRAUSS FOR CU - 18 CORE ELECTRONS - LMAX = 3
*
    NCORE = 18    LMAX = 3
f
  -18.00000000    1  359.2137111
 -119.92593970    2   67.5347369
  -29.55328670    2   14.7222923
  -10.28924330    2    3.9975558
    -.78363630    2    1.1889410
s-f
    3.00000000    0   19.6202650
   20.15792750    1    5.1604389
   34.50019060    2    1.2306099
  -18.98120030    2    1.0850105
p-f
    5.00000000    0   31.9385762
   20.60853280    1   14.9202125
   56.00168880    2   15.6835232
   57.21701070    2    4.9311614
    7.71778780    2    1.0622167
d-f
     .25986160    2    5.1159991
    -.46216800    2     .7396784
*
```

The first line contains a single star. The name of the data group starts with the element symbol followed by the ECP nickname. It is useful to give a comment introduced by the

hash symbol (#) in the next line to indicate the origin of the ECP. The actual ECP data is given in between two lines with a '$*$'. The first line specifies the number of core electrons described by the ECP (NCORE) and the maximum angular momentum number of the projector operators (LMAX) in integers (s=0, p=1, d=2, ...). These are followed by the description of the effective core potential which consists of the angular momentum numbers and by the analytical representation of the operator. The latter includes the coefficient c_m, the exponent N_m of r and the exponent α_m of the gaussian:

$$U_l(r) = \sum_m c_m e^{\alpha_m r^2} r^{N_m}.$$

For a detailed description, see L.R. Kahn, P. Baybutt, and D.G. Truhlar, *J. Chem. Phys.* **65**, 3826 (1976).

7.4 GUESS

The GUESS file is used to control the "placement" of electrons in the SCF initial guess and can be used only with GUESS=READ_SO_MOS (orbitals come from OLDMOS). *All options in the GUESS file must be specified—there are no defaults!*

The following is an example of the GUESS file for a state of the water cation.

```
Line 1   H2O. TEST.
Line 2     3  1  1  0  alpha occ
Line 3     3  0  1  0  beta occ
Line 4     0  0  0  0  alpha pairs to be swapped (irrep 1)
Line 5     0  0  0  0  alpha irrep 2
Line 6     0  0  0  0  alpha irrep 3
Line 7     0  0  0  0  alpha irrep 4
Line 8     0  0  0  0  beta pairs to be swapped (irrep 1)
Line 9     0  0  0  0  beta irrep 2
Line 10    0  0  0  0  beta irrep 3
Line 11    0  0  0  0  beta irrep 4
Line 12    0  0  0  0  alpha locking within each irrep
Line 13    0  0  0  0  beta  locking within each irrep
Line 14    0  0  0  0  alpha printing of initial guess
Line 15    0  0  0  0  beta  printing of initial guess
Line 16    0  0        stopping parameters
Line 17    1           read from OLDMOS
Line 18    1           beta orbitals are copied from alpha orbitals
Line 19    0           reuse GUESS for every SCF calculation
```

This is a formatted file. The trailing text on each line is not parsed as input (as long as each line is at most 80 characters), but it is usually included as a reminder of the options. The input is as follows: (*NIRREP* is the order of the computational point group or 4 in the present case, and *NSPIN* is the number of spins: 1 for RHF, 2 for UHF and ROHF.)

Line 1: `A80`

A title.

Next NSPIN: `NIRREP(I3)`

The alpha (beta) occupation vector.

Next NIRREP*NSPIN: `4(I3)`

Pairs of orbitals to be swapped in each spatial symmetry block (for each spin symmetry). Two numbers are needed to specify each pair; therefore, no more than two interchanges may be made for a given symmetry block and spin.

Next NSPIN: `NIRREP(I3)`

Symmetry block occupation lock flags. Orbital occupation proceeds in the direction of "minimum change" by monitoring $C_{OLD}^{T} * S * C$. Zero is unlocked, a positive integer is locked.

Next NSPIN: `NIRREP(I3)`

Print flags for alpha (beta) initial guess. A positive integer prints the guess for that symmetry block, while a zero does not.

Next 1: `2(I3)`

Stopping parameters. Set the first value to a positive integer to stop the SCF after computing the initial guess.

Next 1: `I3`

I/O parameter. If set to a positive integer, then the initial guess MOS are read from `OLDMOS`.

Next 1: `I3`

UHF creation parameter which copies the alpha MOs to the beta MOs. This is only meaningful if the guess is read from `OLDMOS`. This allows a user, for example, to start a UHF calculation with an RHF closed-shell set of orbitals. A positive integer reads only the alpha orbitals, and a zero reads both sets.

Next 1: `I3`

A flag that forces `xvscf` to read `GUESS` every time an SCF calculation is performed. This applies only to calculations that run multiple SCF calculations (geometry optimizations, HF stability analyses, etc.). Set to 0 for just the first time; otherwise, set to a positive integer. If it is set to 0, ***`GUESS`*** *is deleted after it has been read.*

8 Keywords

8.1 *ACES2 namelist

The user can control the behavior of an ACES II job through the use of keywords in the *ACES2 namelist. In some cases, the value for a keyword can be specified by an integer or by a character string. In our opinion the latter is preferable as it makes the input file more readable.

All possible keywords in the *ACES2 namelist are discussed below. As there are a lot of keywords, we have grouped them according to the general flow of a calculation.

Keyword Conventions

Keywords in the *ACES2 namelist are matched to an initial substring of the actual keyword in xjoda. For example, the full keyword is CALCLEVEL, but the unique substring is CALC, so CALC, CALCULATION, and CALCLEVEL may all be used to set the calculation level. The underlined substring in the following keyword definitions is what is used to match the keyword. As another example, the memory keyword is defined as MEMORY_SIZE, but any keyword in the *ACES2 namelist starting with MEM will be used to set the memory size.

Some keywords should not be set by the user without careful consideration of what the program will do. Either the keyword controls experimental pathways through the program, or the program uses considerable logic to determine the correct default behavior. These keywords are listed in *italics* instead of **bold**.

Value Conventions

Value strings are parsed differently depending on how the corresponding keyword is defined in JODA. Currently, there are handles, strings, integers, and reals. For clarity, this manual also mentions switches and tols (tolerances). A switch is a handle with only two values (ON/OFF), and a tol is an integer N that corresponds to a value of 10^{-N}.

In ACES II, handles are character strings that map onto integers. For example, the CALC keyword has 41 possible values. The code might do one thing if CALC=0 (SCF) and another if CALC=10 (CCSD). Keywords of type handle can accept the handle string or the integer as a value, so CALC=CCSD does the same thing as CALC=10 in the namelist. Ultimately, the internal integer is irrelevant to the user and could change any time; therefore, all keywords are described according to their handles only.

Some keywords only recognize switch-like handles (TRUE/FALSE, ON/OFF, 1/0, etc.) and are listed as type switch. If these keywords appear in the namelist without a value string, then they will be set to ON. Similarly, they can be negated by prefixing them with an exclamation mark. For example *ACES2(RESTART,!NONHF) will register as RESTART=ON

and NONHF=OFF.

Keywords of type string are self-explanatory; however, they are used to read in fixed-format arrays until we define a general expression for array notation. Hopefully the formats of keywords that accept arrays are clearly described in the manual.

Integers are also self-explanatory and can be used in place of handles for those keywords of such type. As mentioned before, a tol type is an integer that corresponds to the negative power of 10. Some keywords are defined with unusual units, and these are appropriately identified in the manual.

Currently, reals apply only to memory-related keywords. These are identified as type "special" because the parser will scale the real by units suffixed to the value string.

8.1.1 System: general

CALCLEVEL = (handle) SCF

Specifies the general level of theory to be used throughout the program. Acceptable values are: SCF, MBPT(2), MBPT(3), SDQ-MBPT(4), MBPT(4), LCCD, LCCSD, UCCSD(4), CCD, UCC(4), CCSD, CCSD[T], CCSD+TQ*, CCSDT-1, CCSDT-1b, CCSDT-2, CCSDT-3, CCSDT-4, CCSDT, *LCCSDT*, *CCD+ST(CCD)*, QCISD(T), CCSD(T), QCISD, CID, CISD, QCISD(TQ), CCSD(TQ), *CCSD+TQ*, *CCSDT+Q**, *CCSDT+Q*, CC5SD(T), CCSD-T, CC3, CCSDT-T1T2, CCSDTQ-1, CCSDTQF-1, CCSDTQ-2, CCSDTQ-3, CCSDTQ, ACCSD, and HFDFT. Emphasized calculation levels (in italics) are not implemented currently but are planned for future versions of the program. CALC=HFDFT has been obsoleted by SCF_TYPE=KS.

MEMORY_SIZE = (special) 15000000W

Sets the "maximum" memory to allocate. Most member executables will not allocate more than the amount specified, but some will attempt to allocate extra bits (on the order of a few megabytes) that are of little consequence compared to the main memory heap.

There is special parsing logic for the corresponding value string. The tail of the string is checked for case-insensitive units and optional prefixes. Units can be W (integer words) or B (bytes), and prefixes can be K, M, or G (kilo, mega, and giga, respectively). The number part is read with the `atof()` C function.

[NOTE: The prefixes scale the units by 2^{10} (1024), not 1000, and the default unit is integer words (4 bytes for 32-bit binaries and 8 bytes for 64-bit binaries).]

RESTART = (switch) ON

Controls the creation, updating, and reading of a checkpoint directory. Currently, only coarse-grain restarts are available, which means geometry optimizations and finite difference calculations can be restarted from the last checkpoint. Fine-grain restarts are not available yet but will also checkpoint SCF, T, and Lambda iterations.

GRAD_CALC = (handle) AUTO

Specifies whether to calculate a gradient analytically or numerically. This should be used only to override analytical gradients since the program will attempt to use them whenever possible.

DERIV_LEV = (handle) -1

Specifies whether or not derivatives of the energy are to be calculated and if so then whether first or second. =NONE Derivatives not calculated, =FIRST First derivatives to be calculated, =SECOND Second derivatives to be calculated. This is automatically set.

8.1.2 System: molecular control

SYMMETRY = (handle) ON

Specifies which subgroup ("computational" point group) of the full point group is to be used in the energy and/or gradient calculation. =OFF forces a no symmetry run (in C_1), =ON runs the calculation in the largest self-adjoint subgroup (D_{2h} and its subgroups), and =FULL uses the full point group. Currently, ACES II does not support groups with degenerate representations, so the FULL option has no value unless JODA is used to make input files for another program package.

SUBGROUP = (handle) DEFAULT

Specifies a lower computational point group symmetry to use (provided it is a subgroup of the full group). Acceptable values are: DEFAULT, C1, C2, CS, CI, C2V, C2H, D2, and D2H. SUBGROUP=C1 is equivalent to SYMMETRY=OFF. The DEFAULT option uses the highest order Abelian subgroup (including the full group).

SUBGRPAXIS = (handle) X

This is a somewhat complicated keyword to use. The value can be X, Y, or Z. The use of the keyword is best described by example: Suppose one is running a calculation on water, and wishes to run it in the C_s point group with the "special" plane being the one that bisects the H-O-H bond angle. SUBGRPAXIS specifies which Cartesian direction in the C_{2v} frame becomes the special direction in the C_s frame. Normally,

JODA will orient water in the yz plane. In this example, the y axis in the C_{2v} frame should be the z axis in the C_s frame, so SUBGRPAXIS=Z.

When the true Abelian subgroup is either C_{2v} or D_{2h}, the orientation is not well defined, and it might be necessary to run `xjoda` twice. If SUBGROUP=0 in the first pass, then the reference orientation for the true Abelian subgroup can be determined and the appropriate value of SUBGRPAXIS can be selected.

NOREORI = (switch) OFF

Forces the program not to change the internal orientation of the molecule specified by the input Cartesian coordinates. To function correctly, this keyword also requires turning off symmetry (SYMMETRY=OFF).

8.1.3 System: debug

PRINT = (integer) 0

Controls the amount of printing in all member executables except JODA. A value of 1 will produce a modest amount of additional output over the default value, which includes some useful information such as SCF eigenvectors.

JODA_PRINT = (integer) 0

Controls the amount of debug printing performed by JODA. The higher the value, the more information is printed. Values of 25 or higher generally do not produce anything of interest to the general user, and values greater than 999 will dump the core vector to disk.

8.1.4 I/O subsystem

FILE_RECSIZ = (special) -1W

Sets the length of the physical records used in direct access file I/O. This value should always be chosen as a multiple of 512 bytes and is parsed like the MEMORY_SIZE keyword. The default value (-1) determines a size based on the value of MEMORY_SIZE.

CACHE_RECS = (integer) -1

Sets the number of physical records (of length FILE_RECSIZE) held in the I/O cache. The maximum number of records is 128, and the default value (-1) determines a number based on the value of MEMORY_SIZE.

INCORE = (handle) NONE

This keyword can be used to reduce disk I/O (usually by a significant amount). Acceptable values are: NONE, NOABCD, NOABCI, NOABIJ, ALL, and T. =NONE and =ALL load no and all lists into core memory, respectively. =NOABCD, =NOABCI, and =NOABIJ load all lists that are at most the size of $\langle Ab|Ci\rangle$, $\langle Ab|Ij\rangle$, and $\langle Ij|Ka\rangle$ molecular integral lists, respectively. These options are implemented in VCC, LAMBDA, DENS, ANTI, BCKTRN, and VEE. The =T option only applies to VCC, which will load T_1 and T_2 amplitudes into memory. All of the other executables will treat =T as =NONE (including LAMBDA, although this should change soon).

8.1.5 Chemical system

COORDINATES = (handle) AUTO

Specifies whether the molecular system is defined in INTERNAL (Z matrix) or CARTESIAN (xyz) coordinates.

UNITS = (handle) ANGSTROM

Specifies the units used for molecular geometries. The value can be ANGSTROM or BOHR.

CHARGE = (integer) 0

Sets the molecular charge.

MULTIPLICTY = (integer) 1

Sets the spin multiplicity $(2S + 1)$.

8.1.6 Basis set

BASIS = (string) SPECIAL

Defines the name of the basis set to use for all atoms in the molecule. If BASIS=SPECIAL, then each atom (in coordinate order) must be specified after the `*ACES2` namelist.

SPHERICAL = (switch) OFF

Controls whether (=ON) spherical harmonic (5d, 7f, 9g, ...) or (=OFF) Cartesian (6d, 10f, 15g, ...) basis functions are used.

LINDEP_TOL = (tol) 5

Sets the tolerance for linear dependencies in the basis set. The basis set is considered linearly dependent and eigenvectors of the overlap matrix are neglected if the associated eigenvalues are less than 10^{-N}.

GENBAS_1 = (integer) 0

This keyword applies to first-row elements (H and HE) and specifies the number of contracted Gaussian functions per shell. There is usually no need to use this keyword, but it can be useful for using a subset of the functions in a particular entry in the `GENBAS` file, particularly for generally contracted basis sets. For example, if entry H:BASIS in the `GENBAS` file contains 7 contracted s functions, 4 p functions, and one d function, then setting GENBAS_1=730 would eliminate the last p function and the d function. The default for this keyword is to use the unaltered entry in `GENBAS`.

GENBAS_2 = (integer) 0

This keyword performs the same function as GENBAS_1 but applies only to second-row atoms.

GENBAS_3 = (integer) 0

This keyword performs the same function as GENBAS_1 and GENBAS_2, but applies only to third-row atoms.

CONTRACTION = (handle) GENERAL

Specifies the contraction scheme used by the integral and integral derivative programs. This can be set to SEGMENTED or GENERAL.

[NOTE: Even for truly segmented basis sets, both integral and integral derivative programs run significantly faster in GENERAL mode.]

ECP = (switch) OFF

Controls whether effective core potentials (a kind of pseudopotential) are used (=ON) or not (=OFF). ECP=ON requires BASIS=SPECIAL and specification of the ECP data sets (in a file named ECPDATA).

8.1.7 Integrals

INTEGRALS = (handle) VMOL

Specifies which integral package to use. This is not a very robust keyword and should always be set to VMOL unless otherwise directed to change it. Other values include SEWARD and GAMESS, but these options will not work unless the binaries were linked to the appropriate libraries, which require third-party license agreements.

INTGRL_TOL = (tol) 14

Sets the tolerance for storing the two-electron integrals such that integrals having an absolute value greater than 10^{-N} will be stored on disk.

DIRECT = (switch) OFF

Controls whether two-particle AO integrals are calculated on-the-fly or calculated once and stored in a file. This cannot be used with VMOL integrals.

8.1.8 Reference

REFERENCE = (handle) RHF

Specifies the type of SCF reference to use. Supported references include spin-restricted Hartree-Fock (=RHF), spin-unrestricted Hartree-Fock (=UHF), and spin-restricted open-shell Hartree-Fock (=ROHF). These apply to Kohn-Sham DFT references as well (e.g., RHF is spin-restricted Kohn-Sham). REF=TWODET (two-determinant reference for open-shell singlet CC calculations) is available, but the orbitals are obtained from a closed-shell state and are *not* optimum open-shell singlet (or low spin triplet) SCF orbitals. The orbital occupancy for subsequent CCSD calculations is specified by the various QRHF keywords.

NONHF = (switch) OFF

Signals the correlation energy code if the reference wavefunction satisfies the Hartree-Fock equations ($F_{ai} = 0$). Usually there is no need to set this keyword since standard non-HF reference functions (QRHF and ROHF) set this flag internally; however, if you input a set of orbitals to the correlation energy code directly, then it might be necessary to set NONHF=ON.

ORBITALS = (handle) -1

Specifies semicanonical orbitals ($F_{i\neq j} = F_{a\neq b} = 0$) in non-HF calculations. "Semicanonical" orbitals are obtained by diagonalizing the occupied-occupied (ij) and virtual-virtual (ab) blocks of the spin-orbital Fock matrix and can be advantageously exploited in certain post-SCF calculations. (This is particularly true for ROHF-MBPT and non-iterative triple excitation corrections.) There is no default value for this parameter, and considerable logic is used to determine the orbital type in post-SCF non-HF calculations (if the keyword is not defined). It is strongly recommended that this keyword not be used by anyone who is not thoroughly familiar with non-HF CC/MBPT methods, since the logic used to set the default value is sound. =STANDARD uses the orbitals obtained in the reference function calculation without modification, and =SEMICANONICAL forces a transformation to semicanonical orbitals.

PERT_ORB = (handle) UNKNOWN

Specifies whether the gradient formulation assumes that the perturbed orbitals are not those in which the Fock matrix is diagonal (=STANDARD). =CANONICAL means that the perturbed orbitals are assumed to be canonical. This keyword must be set to CANONICAL in gradient calculations with methods that include triple excitations (i.e., MBPT(4), CCSD[T], CCSD(T), and QCISD(T)).

8.1.9 SCF: general

SCF_TYPE = (handle) HF

Switches the SCF code between the standard Hartree-Fock program (=HF) and the Kohn-Sham DFT program (=KS). If SCF_TYPE=KS, then the KS-DFT program generates the SCF reference and requires separate namelists (viz. *VSCF and *INTGRT).

FOCK = (handle) PK

Specifies the algorithm for constructing the Fock matrix in SCF calculations. =PK uses the PK-supermatrix approach while =AO constructs the matrix directly from the AO integrals. In general, PK is faster but results in *considerable* use of disk space when out-of-core algorithms are required.

CHECK_SYM = (handle) OVERRIDE

Specifies the action taken when the density matrix does not transform as the totally symmetric irreducible representation of the full molecular point group. =NORMAL terminates the run if the molecule is nonlinear, while =OVERRIDE allows the job to continue but prints a warning message. Nonsymmetric density matrices result from calculations on electronically degenerate states or from broken-symmetry SCF solutions. Often, such calculations do not give meaningful results and inexperienced users are encouraged to use CHECK_SYM=NORMAL in their calculations on nonlinear molecules. For Π, Δ, Φ, ... states of linear molecules, however, meaningful calculations can still be performed even though the density matrix is not symmetric.

SCF_PRINT = (integer) 0

[This keyword is currently not used but exists for future compatibility.]

8.1.10 SCF: orbital control

GUESS = (handle) MOREAD

Specifies where the initial SCF eigenvectors are read from. The HF-SCF executable checks multiple places for pre-existing orbitals, but this keyword can override that logic.

=MOREAD → JOBARC file

=CORE → core Hamiltonian

=READ_SO_MOS → OLDMOS file (from NEWMOS)

=READ_AO_MOS → OLDAOMOS file (from AOBASMOS)

Other options include =NDDO, =WALT_PRJDEN, =MIN_BASIS, and =HUCKEL, although some of these are still experimental and not fully supported.

OCCUPATION = (string of irrep-by-spin array)

Specifies the orbital occupancy of the reference wavefunction in terms of the occupation numbers per irrep per spin. The occupancy is specified by NIRREP (or 2*NIRREP) integers that define the number of occupied orbitals of each symmetry type. (NIRREP is the number of irreducible representations in the computational point group.) If there are no orbitals of a particular symmetry type, then a zero must be entered. If the reference wavefunction is for an open-shell system, then two strings of NIRREP occupation numbers separated by a forward-slash (/) define the α and β sets of orbitals.

An example of the use of the OCCUPATION keyword for the water molecule would be OCCUPATION=3-1-1-0. For the $^2A_1$ water cation (an open-shell system), the keyword would be specified by OCCUPATION=3-1-1-0/2-1-1-0.

It should be noted that the VMOL integral program orders the irreducible representations in a strange way, which most users do not perceive to be a logical order. Hence, it is advised to run a single-point SCF calculation to determine the initial number and ordering of the irreducible representations. The occupation keyword may be omitted, in which case an initial orbital occupancy is determined by diagonalizing the core Hamiltonian. In many cases, SCF calculations that use the core Hamiltonian guess will converge to the lowest energy SCF solution, but this is not guaranteed.

LOCK_ORBOCC = (switch) OFF

Controls orbital occupancies among symmetry blocks in the SCF iterations. =ON locks the occupation to the OCCUPATION value array (or the initial guess if OCCUPATION is undefined); =OFF permits the occupation to change.

[NOTE: If the OCCUPATION keyword is defined, then LOCK_ORBOCC is switched on automatically.]

NEWVRT = (switch) OFF

Signals if the SCF virtual orbitals are to be replaced by a set determined from an $N-1$ potential. This is an orthogonal transformation within the virtual space. As long as appropriate f_{ij} and f_{ab} are included, the energies of standard single-reference methods are unchanged. However, for TD-CC methods, this keyword mixes one of the occupied orbitals with the virtual space and the results are changed. It is anticipated that TD-CCSD calculations will be improved by the use of this keyword, but this has yet to be demonstrated. NEWVRT is useful for interpreting results of EOM-EE calculations since excitations tend to be more easily identified as involving one occupied and one virtual orbital. =OFF does not rotate the virtual space, and =ON rotates the virtual space.

8.1.11 SCF: iteration control

SCF_CONV = (tol) 7

Sets the convergence criterion for the SCF equations. Equations are considered converged when the maximum change in density matrix elements is less than 10^{-N}.

SCF_MAXCYC = (integer) 150

Sets the maximum number of SCF iterations.

DAMP_TYP = (handle) NONE

Specifies what type of damping is used during the SCF iterations. The value can be NONE (no damping), DAVIDSON (Davidson's empirical dynamical damping scheme), and STATIC (a fixed damping parameter). Note that RPP convergence extrapolation is not turned on until the damp factor has gone below DAMP_TOL and the energy change is below 0.05 a.u. DAMP_TYP=DAVIDSON is recommended even though NONE is the default.

DAMP_TOL = (integer) 10

If DAMP_TYP=DAVIDSON, then the cutoff is defined to be $N * 0.01$, so the default cutoff is 0.1.

DAMPSCF = (integer) 20

If DAMP_TYP=STATIC, then the static damping factor used in the SCF iterations is $N * 0.01$, so the default damping factor is 0.2.

SCF_EXTRAP *(previously RPP)* = (switch) ON

Controls whether or not the reduced partitioning procedure is to be used to accelerate convergence of the SCF equations.

SCF_EXPORDER *(previously RPP_ORDER)* = (integer) 6

Sets the number of density matrices to be used in the RPP convergence acceleration procedure.

SCF_EXPSTAR *(previously RPP_LATEST)* = (integer) 8

Sets the latest iteration for initiation of the RPP convergence acceleration procedure. RPP is switched on when the error falls below a certain threshold, but in difficult cases, in which the iterations are oscillatory, it is necessary to force it on when the error is still large. In these cases, the RPP will begin on the iteration number specified by this parameter.

LSHF_A1 = (integer) 0

Sets the doubles-singles level shifting parameter α, in which $\alpha = N*0.01$. This keyword is currently only meaningful in ROHF calculations.

LSHF_B1 = (integer) 0

Sets the singles-virtuals level shifting parameter β, in which $\beta = N*0.01$. This keyword is currently only meaningful in ROHF calculations.

8.1.12 SCF reference adjustments

DROPMO = (string of 1D array) 0

Lists which molecular orbitals will be dropped from the post-SCF calculation. The orbitals are ordered by eigenvalue from the most stable (negative energy) to the most unstable (largest positive energy) regardless of irrep. The delimiter is a forward-slash (/) that separates single orbitals and orbital ranges (x-y inclusive). For example, the string "1-5/55/62-64" will cause VTRAN to drop orbitals 1, 2, 3, 4, 5, 55, 62, 63, and 64. For UHF calculations, the appropriate orbitals are deleted from both spin cases.

HFSTABILITY = (handle) OFF

Checks the stability of RHF and UHF wavefunctions and optionally rotates the orbitals to a lower SCF solution. There are three possible options for this keyword. =OFF does nothing and =ON performs a stability check and returns the number of negative eigenvalues in the orbital rotation Hessian. =FOLLOW performs the stability check and then proceeds to rotate the SCF orbitals in the direction of a particular negative eigenvalue of the orbital rotation Hessian (see the explanation of the ROT_EVEC keyword below), after which the SCF is rerun.

ROT_EVEC = (integer) 0

Defines which eigenvector of the orbital rotation Hessian will be used to rotate the original SCF orbitals. By default, it will use the eigenvector with the lowest eigenvalue of irrep 1 (the totally symmetric part of the block-factored Hessian). This choice often leads to the lowest energy SCF solution. For RHF stability checks, only those instabilities which correspond to RHF solutions will be considered. It is important to understand that following non-symmetric eigenvectors lowers the symmetry of the wavefunction and that following RHF→UHF stabilities leads to a UHF solution. To converge the SCF roots associated with such instabilities, one must run the calculation in reduced symmetry and as a closed-shell UHF case, respectively. If not set to 0, the program follows the vector with the N^{th} lowest eigenvalue having the proper symmetry (totally symmetric) and spin (RHF→RHF or UHF→UHF) properties.

BRUECKNER = (switch) OFF

Controls whether Brueckner orbitals are to be determined for the specified CC method.

BRUCK_CONV = (tol) 4

Sets the convergence criterion in Brueckner iterations. The calculation is considered converged when the largest single excitation amplitude falls below 10^{-N}.

QRHF_GENERAL = (string of 1D array) 0

The presence of this keyword turns on QRHF logic, and the value of each element specifies which irrep is created (positive) or annihilated (negative). In the QRHF scheme, an RHF (closed-shell) SCF calculation is performed with the OCCUPATION keyword or the CHARGE and MULTIPLICITY keywords. The correlated calculation is then performed on an open-shell reference generated from this closed-shell state by removing, adding, or exciting electrons. Any number of electrons may be added and all of the electrons may be removed from the SCF reference. The array elements associated with this keyword must be separated by forward-slashes (/) and be positive or parenthesized negative. The absolute value of each element specifies the symmetry block involved in the addition or removal of electrons. The numerical ordering of the symmetry blocks is consistent with then OCCUPATION keyword (the symmetries of the integrals).

By default, the electrons of each irrep are added to the lowest unoccupied α orbital in the symmetry block and removed from the highest occupied β orbital. Different orbitals and spins can be specified with the QRHF_ORBITAL and QRHF_SPIN keywords.

[NOTE: Gradients and property calculations are currently available only *for cases involving addition or removal of electrons. Mixed cases involving both processes are not available. Gradients and properties are available for open-shell singlet CCSD wavefunctions for the case that the open-shell orbitals have different symmetries.] ???*

QRHF_ORBITAL = (string of 1D array) 1

By default, in QRHF calculations, electrons are removed from the highest occupied β orbital in a symmetry block (symmetry block HOMO), while electrons are added to the lowest unoccupied α orbital within a symmetry block (symmetry block LUMO). The purpose of the QRHF_ORBITAL keyword is to allow additional flexibility in choosing which orbitals will have their occupation numbers altered. The value of this keyword gives the index with respect to the default orbital for the orbital which will be populated or depopulated. For calculations involving more than one removal or addition of electrons, values are separated by forward-slashes and correspond one-for-one to the QRHF_GENERAL value array. For example, specifying QRHF_GENERAL=2/(-4) and QRHF_ORBITAL=3/2 means that an electron will be added to the third lowest virtual in symmetry block 2 and another will be removed from the second highest occupied orbital in symmetry block 4. Examples in Sections 9.1.4 and **??** (pages 71 and **??**) further illustrate the QRHF input options.

QRHF_SPIN = (string of 1D array)

Specifies the spins of QRHF occupations by overriding the remove-from-β and add-to-α behavior. An element value of 1 means α spin and 2 means β spin.

[NOTE: This option allows one to construct low-spin determinants, which generally are unsuitable for single-reference coupled-cluster calculations. An important exception is the open-shell singlet coupled-cluster method (REFERENCE=TWODET).]

UNO_REF = (switch) OFF

Controls the use of the new UNO reference state. The ACES II calculation is initialized by an open-shell SCF calculation specified in the usual way (either UHF or RHF). The resulting α and β density matrices from the SCF calculation are then added into a spatial density matrix and this is diagonalized to yield unrestricted natural orbitals (Pulay's UNOs). These orbitals are used to create a new reference determinant. The UNOs can be used in conjunction with QRHF and NEWVRT keywords to create more complicated vacuum states for post-Hartree-Fock calculations.

[NOTE: The UNO keywords together with MAKERHF can be used to create a reference determinant that is used primarily in connection to certain types of multireference calculations run in MRCC.]

UNO_CHARGE = (integer) 0

Sets the charge of the final reference state.

UNO_MULT = (integer) 1

Sets the spin-multiplicity of the final reference state. The orbitals are occupied in order of their natural occupation number. Often this keyword is used to create a closed shell-reference (vital if the MRCC module is used). In this case, the program can proceed as an RHF calculation.

MAKERHF = (switch) OFF

MAKERHF=ON instructs the post-SCF logic to proceed as an RHF calculation and must be specified in an MRCC calculation.

8.1.13 Post-SCF file options

SINGLE_STORE = (switch) OFF

Controls storing permutations of commonly resorted two-particle quantities like MO integrals and T amplitudes. If a calculation is running out of disk space, then setting SINGLE_STORE=ON might allow the calculation to finish.

ABCDTYPE = (handle) STANDARD

Specifies the way that VVVV molecular orbital integrals (having four virtual MOs) are handled in post-SCF calculations. =STANDARD uses a technique that minimizes CPU time but makes liberal use of disk storage (particularly during the integral processing program INTPRC). =MULTIPASS avoids creating intermediate sort files during INTPRC by reading the HF2 files (created by VTRAN) multiple times (note that this option requires more CPU processing time and the setting of HF2_FILE=SAVE). =AOBASIS uses an AO-based algorithm to evaluate all terms involving the VVVV MO integrals. Again, use of this option results in considerably longer CPU times but significantly reduces the amount of disk storage (more so for UHF and ROHF references than RHF).

AO_LADDERS = (handle) SINGLEPASS

Specifies the algorithm used by ACES II when ABCDTYPE=AOBASIS. =MULTIPASS uses an approach where the AO integral file is read a number of times to maximize vectorization and is usually the optimal strategy on vector supercomputers. =SINGLEPASS determines the contributions with only a single pass through the AO integral files at the cost of significantly reduced vectorization.

SAVE_INTS = (switch) OFF

Controls deletion of the AO integrals file(s) after the AO to MO integral transformation. If SAVE_INTS=ON, then these files are not deleted. Any method that requires AO integrals during a post-SCF calculation will automatically switch on this keyword.

VTRAN = (handle) FULL/PARTIAL

Specifies what type of integral transformation is to be performed in the transformation program VTRAN. =FULL/PARTIAL allows the program to choose the appropriate type of transformation, while =FULL forces a full integral transformation, and =PARTIAL skips the VVVV integrals. This keyword is set automatically based on other relevant keywords.

HF2_FILE = (handle) USE

Specifies whether the HF2 file series (including HF2AA, HF2BB, and HF2AB) is deleted after the first stage of integral processing. The default is to delete these files; however, when ABCDTYPE=MULTIPASS, these files must not be deleted and the program sets HF2_FILE=SAVE.

ABCDFULL = (handle) UNKNOWN

This is a debug aid and is concerned with the storage of VVVV integrals and effective Hamiltonian elements. It is only relevant to RHF calculations, and it should never be necessary for users to set this keyword. Possible values are: =UNKNOWN (the program will choose an appropriate value), =ON (full storage), and =OFF (reduced storage).

HBARABCD = (handle) UNKNOWN

Controls the formation (and storage) of the VVVV block of the effective Hamiltonian. It should never be necessary for users to set this keyword. Possible values are: =UNKNOWN (the program will choose an appropriate value), =ON (full storage), and =OFF (reduced storage).

HBARABCI = (handle) UNKNOWN

Controls the formation (and storage) of the VVVO block of the effective Hamiltonian. It should never be necessary for users to set this keyword. Possible values are: =UNKNOWN (the program will choose an appropriate value), =ON (full storage), and =OFF (reduced storage).

GAMMA_ABCD = (handle) DISK

Controls the evaluation of the two-particle density elements when all four indices correspond to virtual orbitals. This works in conjunction with the ABCDTYPE keyword. When ABCDTYPE=AOBASIS, the ABCD integrals are not stored on disk and GAMMA_ABCD is set to DIRECT. When ABCD integrals are stored to disk, GAMMA_ABCD must be set to DISK.

8.1.14 Post-SCF calculations

CC_CONV = (tol) 7

Sets the convergence criterion for the CC and Lambda equations. Equations are considered converged when the maximum change in amplitudes is less than 10^{-N}.

CC_MAXCYC = (integer) 50

Sets the maximum number of CC or Lambda iterations.

CC_EXTRAPOL *(previously RLE)* = (handle) DIIS

Specifies the type of convergence acceleration used during the CC iterations. =STANDARD uses the RLE method of Bartlett and Purvis with periodic extrapolation of the solution vector, =DIIS uses the DIIS approach of Pulay, =NOJACOBI uses the RLE method with continuous extrapolation, and =OFF uses no convergence acceleration. In general, DIIS works well for a wide range of molecular systems while RLE can be better for certain cases. NOJACOBI might offer advantages in cases where the reduced subspace becomes singular too rapidly. NOJACOBI requires some additional disk storage, which might be disadvantageous for very large calculations. =OFF is generally a bad idea for CC calculations but might be preferred by some CI calculations.

CC_EXPORDER *(previously ORDER_RLE)* = (integer) 5

Sets the maximum number of iterates to include in the R matrix used by RLE and DIIS. The maximum value allowed is 25.

XFORM_TOL = (tol) 11

Sets the tolerance for storing transformed integrals. Integrals less than 10^{-N} are neglected and not stored on disk.

NTOP_TAMP = (integer) 15

Sets the N largest T amplitudes to print for each spin case and excitation level.

TAMP_SUM = (integer) 0

Sets how often the largest T amplitudes are printed. =0 only prints amplitudes at the beginning and end of the run, =1 prints amplitudes after every iteration, =2 prints amplitudes after every other iteration, etc.

DENSITY = (handle) RELAXED

Specifies whether the relaxed density matrix is computed for correlated wavefunctions. =RESPONSE skips the orbital relaxation terms that contribute to the density matrix. This keyword should only be used by advanced users and ACES II developers who are testing new theories.

DOHBAR = (switch) OFF

Controls what action is taken by the linear response program. =ON calculates and saves the full effective Hamiltonian. =OFF solves the "lambda" linear response equations.

8.1.15 Excited states: general

EXCITE = (handle) NONE

Specifies the type of excited-state calculation to perform. Available values are: NONE, TDA, RPA, EOMEE, CIS, CIS(D), P-EOMEE, EOM-BWPT2, and STEOM. *This keyword needs a lot of attention...*

ESTATE_TOL = (tol) 5

Sets the tolerance used in converging EOM-CC excited state calculations. The iterative diagonalization continues until the RMS residual falls below 10^{-N}.

ESTATE_MAXC = (integer) 20

Sets the maximum number of iterative diagonalization steps for each root in excited state calculations.

EOM_MAXCYC = (integer) 50

Sets the number of iterations in EOM excited state calculations. If it has the value N and $NROOT$ roots have been requested for a given symmetry, then the program will allow up to $N*NROOT$ iterations to find all of the requested roots for that symmetry.

This keyword is only used by the iterative solution of the HF Stability analysis and has nothing to do with EOM.

PROGRAM = (handle) DEFAULT

EOMREF = (handle) NONE

IMEM_SIZE = (special) 3000000W

EOM_PRJCT = (handle) NONE

NT3EOMEE = (handle) NONE

ACC_SYM = (handle) NONE

8.1.16 Excited states: properties

ESTATE_PROP = (handle) OFF

Specifies whether any excited state one-electron properties are to be calculated, but it only applies to EOM-CC calculations. Proper use of this keyword might require some fairly advanced knowledge of quantum chemistry and the available options are discussed here. The values are:

=OFF → no properties or transition moments are calculated

=EXPECTATION → transition moments and dipole strengths are calculated along with selected one-electron properties which are evaluated as expectation values

=UNRELAXED → selected one-electron properties are calculated in an approximation that neglects relaxation of molecular orbitals

=RESPONSE → selected one-electron properties are calculated as analytic first derivatives of the energy

Except for EOMCC calculations on two-electron systems (which are exact), properties obtained by the three approaches will not be equivalent. The default value for this keyword is slightly complicated. For TDA calculations, the default is EXPECTATION since the evaluation of transition moments involves only a negligible amount of additional computation relative to the evaluation of the excitation energies. For EOMCC, the default is OFF since evaluation of any transition moments or properties requires approximately twice the computational time. ESTATE_PROP=RESPONSE is not available for EOMCC calculations. Transition moments and dipole strengths are evaluated by default for all values of ESTATE_PROP other than OFF.

8.1.17 Excited states: affinities

EA_CALC = (handle) NONE

Specifies the method for calculating electron affinities of a closed-shell parent state. The values are: =NONE, =MBPT(2) (strict second-order perturbation theory), =SO_DYSON (second-order Green's Function, iterating the MBPT(2) formula), =OVGF, =P_EOMEA (partitioned EA-EOM, the 2p1h-2p1h block is treated as diagonal), =EA_EOMCC (H-bar is diagonalized over 1p and 2p1h configurations), =COMBO, =OS_CCSD (a state-selective multireference CC method, in which both orbitals and cluster amplitudes are optimized for the final state of interest). This last method requires additional input in the `mrcc_gen` namelist. Analytical gradients are available for EA-EOMCC and P-EOMEA, and require additional input though the `mrcc_gen` namelist. Properties and transition properties of the EA states can be requested by additional input in MRCC namelists (`*ea_calc` section).

EA_SYM = (string of 1D array) 0

Specifies the number of states to be calculated by the EA_CALC calculation. A string (e.g., 4-2-2-0) has to be provided that indicates the number of doublet states in each symmetry block. The example string requests 4 doublet states of A1 symmetry, 2 doublet states in symmetry blocks 2 and 3, and 0 in block 4. The number of EA states to be calculated can also be specified by listing an energy range using ea_low and ea_high keywords in the `mrcc_gen` namelist. This can be particularly relevant in vibrational frequency calculations in which the symmetry changes at different geometries.

DEA_CALC = (handle) NONE

Specifies the method for calculating double electron affinities after calculation of the closed-shell parent state. This type of calculation can be very useful to describe certain multireference situations like the C atom or the N_2O_2 molecule. The values are: =NONE, =TDA (the analogue of CIS, in which a CI calculation is performed over 2-particle states), =EOMCC, =STEOM (Similarity Transformed Equation of Motion), =OS_CCSD, =SS_STEOM. DEA-STEOM is only meaningful if EA_CALC=EA_EOMCC. OS-CCSD and SS-STEOM are multireference CC methods, which are still in an experimental stage. Analytical gradients are available for TDA and STEOM and require additional input though the `mrcc_gen` namelist. Properties and transition properties of the DEA states can be requested by additional input in MRCC namelists (`*dea_calc` section).

DEA_SYM = (string of 1D array) 0

Specifies the number of states to be calculated by a DEA calculation. A string (e.g., 4-2-2-0/2-1-1-1) has to be provided that indicates the number of singlet states in each symmetry block, followed by the number of triplet states in each symmetry block. The example string requests 4 singlet states of A1 symmetry, 2 singlet states in symmetry blocks 2 and 3, and 0 in block 4. In addition, 2 triplet states will be calculated in block 1 and 1 triplet in blocks 2, 3, and 4 each. The triplet vector and the forward-slash are optional.

8.1.18 Excited states: electronic (absorption)

EE_SYM = (string of 1D array) 0

Specifies the number of states to be calculated by an EE calculation. A string (e.g., 4-2-2-0) has to be provided that indicates the number of singlet states in each symmetry block. The example string requests 4 singlet states of A1 symmetry, 2 states in symmetry blocks 2 and 3, and 0 in block 4. The number of EE states to calculate can also be specified by listing an energy range using ee_low and ee_high keywords in the `mrcc_gen` namelist. This can be particularly relevant in vibrational frequency calculations in which the symmetry changes at different geometries. RHF is limited to singlets, but UHF can do singlets and triplets using the N-N/N-N notation like DEA_SYM.

EE_SEARCH = (handle) LOWEST

Specifies the character of the states to calculate. If EE_SYM is not specified, the program attempts to determine all states of the given character; otherwise, it uses the symmetry constraints imposed by EE_SYM. The values are: =LOWEST (find the lowest energy roots regardless of character), =CORE (find excitations from core orbitals using guess vectors from a projected TDA matrix), =LUMO (find excitations to the LUMO), and =HOMO (find excitations to the HOMO).

ESTATE_SYM = (string of 1D array) 0

Specifies the number of excited states which are to be determined in each irreducible representation of the computational subgroup. The program attempts to find all of the lowest roots, but this is not guaranteed since the eigenvalue problem is not solved by direct matrix diagonalization, rather by an iterative (modified Davidson) algorithm. For excited state gradient calculations (TDA only), only one root can be specified, so only one non-zero entry in the string is allowed and that entry must be set to one. The format used for this keyword is identical to that used in the OCCUPATION keyword. For example, for a computational subgroup having four symmetry species, the string

ESTATE_SYM=3-1-0-2 specifies that 6 total roots should be searched for, three in the first block, one in the second block, and two in the fourth block. *This keyword is the same as EE_SYM, but EE_SYM is read when PROGRAM=ACES3 and ESTATE_SYM is read when PROGRAM=ACES2.*

8.1.19 Excited states: ionizations

IP_CALC = (handle) NONE

Specifies the method for calculating ionization potentials of the closed-shell parent state. The values are: =NONE, =MBPT(2) (strict second-order perturbation theory), =SO_DYSON (second-order Green's Function, iterating the MBPT(2) formula), =OVGF, =P_EOMIP (partitioned IP-EOM, the 2h1p-2h1p block is treated as diagonal), =IP_EOMCC (H-bar is diagonalized over 1h and 2h1p configurations), =COMBO, and =OS_CCSD (a state-selective multireference CC method, in which both orbitals and cluster amplitudes are optimized for the final ionized state of interest). This last method requires additional input in the `mrcc_gen` namelist. Analytical gradients are available for IP-EOMCC and P-EOMIP and require additional input though the `mrcc_gen` namelist. Properties and transition properties of the IP states can be requested by additional input in MRCC namelists (`*ip_calc` section).

IP_SYM = (string of 1D array) 0

Specifies the number of states to calculate by an IP calculation. A string (e.g., 4-2-2-0) has to be provided that indicates the number of doublet states in each symmetry block. The example string requests 4 doublet states of A1 symmetry, 2 doublet states in symmetry blocks 2 and 3, and 0 in block 4. The number of IP states to be calculated can also be specified by listing an energy range using ip_low and ip_high keywords in the `mrcc_gen` namelist. This can be particularly relevant in vibrational frequency calculations, in which the symmetry changes at different geometries.

IP_SEARCH = (handle) VALENCE

Specifies the character of the IP states to zoom in on. If IP_SYM is not specified, the program attempts to determine all IP's of the given character; otherwise, it uses the symmetry constraints imposed by IP_SYM. The values are: =VALENCE (all valence IP's), =LOWEST (only the ground state of the cation), =COREIP (only the core ionization potentials—IPs larger than about 100 eV), =SHAKEUP (shake-up states), and =KOOPMANS (all principle IP's having predominantly 1h character). KOOPMANS type can be very difficult for inner valence ionization potentials, and SHAKEUP is not implemented yet.

DIP_CALC = (handle) NONE

Specifies the method for calculating double IP states of the closed-shell parent state. This type of calculation can be very useful to describe certain multireference situations like biradicals. The values are: =NONE, =TDA (the analogue of CIS, in which a CI calculation is performed over 2-hole states), =EOMCC (H-bar is diagonalized over 2h and 3h1p calculations), =STEOM (Similarity Transformed Equation of Motion), =OS_CCSD, and =SS_STEOM. To run a meaningful DIP-STEOM calculation one must also set IP_CALC=IP_EOMCC. OS-CCSD and SS-STEOM are multireference CC methods, which are still in an experimental stage. Analytical gradients are available for TDA and STEOM and require additional input though the `mrcc_gen` namelist. Properties and transition properties of the DIP states can be requested by additional input in MRCC namelists (`*dip_calc` section).

DIP_SYM = (string of 1D array) 0

Specifies the number of states to be calculated by a DIP calculation. A string (e.g., 4-2-2-0/2-1-1-1) has to be provided that indicates the number of singlet states in each symmetry block, followed by the number of triplet states in each symmetry block. The example string requests 4 singlet states of A1 symmetry, 2 singlet states in symmetry blocks 2 and 3, and 0 in block 4. In addition 2 triplet states will be calculated in block 1 and 1 in blocks 2, 3, and 4 each. The triplet vector and the forward-slash are optional.

8.1.20 Excited states: gradients

ZETA_TYPE = (handle) DIIS

Specifies the algorithm used to solve linear equations (zeta equations) in excited state gradient theory. =POPLE uses Pople's orthogonal subspace approach, and =DIIS uses Pulay's DIIS approach.

ZETA_CONV = (tol) 12

Sets the convergence criterion for the iterative solution of the Zeta equations and Z-vector equations. The solutions are considered to be converged when the error falls below 10^{-N}.

ZETA_MAXCYC = (integer) 50

Sets the maximum number of cycles allowed for the solution of the Zeta equations.

RESRAMAN = (switch) OFF

8.1.21 Properties

PROPS = (handle) OFF

Specifies whether properties other than the energy, geometry, and vibrational frequencies are calculated. Acceptable values are:

=OFF → no properties are calculated

=FIRST_ORDER → dipole moment, quadrupole moment, electrical field gradients, spin densities, etc. and the approximate perturbative relativistic correction to the energy by Cowan-Griffin

=SECOND_ORDER → frequency-independent polarizabilities, commonly known as the CPHF polarizabilities, provided CALC=SCF

=EOM_NLO → both frequency-dependent and -independent polarizabilities at the CCSD level using EOM-CC (see EOMPROP)

=J_SO → paramagnetic and diamagnetic spin-orbit contributions to the NMR spin coupling constant

=J_SD → spin-dipole contribution to the NMR spin coupling constant

=J_FC → Fermi-contact contribution to the NMR spin coupling constant

=JSC_ALL → all four contributions

=NMR → NMR chemical shifts (limited to SCF and MBPT(2) with SPHERICAL=OFF)

Note that options J_SO, J_SD, J_FC, and JSC_ALL require CALC=CCSD and that **options J_SO, J_FC, and JSC_ALL require REF=UHF**. (see EOMPROP)

EOMPROP = (handle) CILIKE

Specifies the method of calculating EOM-CCSD second-order properties (polarizabilities and spin-spin coupling constants). =CILIKE uses a CI-like formula, which is not rigorously size-extensive, =LINEAR removes unlinked diagrams from the CI-like formula, =QUADRATIC uses a size-extensive quadratic formula, and =COMBO computes all approximations. This keyword only has meaning when PROP =EOM_NLO, =J_SO, =J_FC, =J_SD, or =JSC_ALL.

TDHF = (switch) OFF

Controls whether a time-dependent Hartree-Fock calculation of nonlinear optical properties is to be performed. This keyword can only be used for closed-shell SCF calculations with no dropped MOs. The nonlinear properties which are to be calculated are specified by a namelist which is put at the end of the `ZMAT` file. A description of this namelist and an example can be found in Section 9.1.10 (page 77).

CPHF_CONVER = (tol) 12

Sets the convergence criterion for the iterative solution of the CPHF and Z-vector equations. The solutions are considered to be converged when the error falls below 10^{-N}.

CPHF_MAXCYC = (integer) 64

Sets the maximum number of cycles allowed for the solution of the CPHF and Z-vector equations.

TREAT_PERT = (handle) SIMULTANEOUS

This keyword is used for certain types of correlated second-derivative calculations [presently only GIAO NMR shift calculations] and directs ACES II to either treat all perturbations at once or treat them sequentially. The latter approach results in less demand for physical disk space at the cost of increased cpu time. Available options are SIMULTANEOUS and SEQUENTIAL.

XFIELD = (integer) 0

Sets the X-component of an external electric field. The value must be specified as an integer and the field used by the program will be $N * 10^{-6}$. This allows field strengths $|\mathcal{E}| > 10^{-6}$ to be used.

YFIELD = (integer) 0

Sets the Y-component of an external electric field. See above.

ZFIELD = (integer) 0

Sets the Z-component of an external electric field. See above.

PRP_INTS = (handle) PARTIAL

Specifies the types of property integrals that are computed. Setting this to FULL or PARTIAL computes the full set or sub set, respectively. This works in conjunction with the PROPS keyword and the defaults are set automatically.

8.1.22 Geometry optimization: general

GEOM_OPT = (handle) NONE

Specifies the scope of coordinate optimizations. This keyword is automatically set to PARTIAL if a geometry optimization is implied with asterisks in the internal coordinate Z matrix. Setting GEOM_OPT=FULL will optimize all coordinates regardless of asterisks in the Z matrix (COORD=INTERNAL) and will optimize a Cartesian input geometry using Redundant Internal Coordinates (RICs).

CURVILINEAR = (handle) OFF

Specifies whether or not the Hessian matrix is transformed (nonlinearly) to curvilinear internal coordinates. =OFF turns the transformation off if analytical force constants are not available, while it is always performed if CURVILINEAR=ON. =NO unconditionally turns the transformation off. This keyword is set automatically.

8.1.23 Geometry optimization: stepping algorithm

OPT_METHOD = (handle) AUTO

Specifies the geometry optimization strategy. =AUTO, =NR uses a straightforward Newton-Raphson minimum energy search, =RFA uses the Rational Function Approximation minimum energy search (and can be used when the initial structure is in a region where the number of negative Hessian eigenvalues is nonzero), =MANR uses a Morse-adjusted Newton-Raphson minimum energy search (and is very efficient if the Hessian is available), =EVFTS uses Cerjan-Miller eigenvector following for finding a transition state (can be started in a region where the Hessian index, the number of negative Hessian eigenvalues, is not equal to one), =MAEVFTS uses a Morse-adjusted eigenvector following for finding a transition state.

MAX_STEP = (integer) 300

Sets the largest step size in millibohr.

STP_SIZ_CTL = (handle) TRUST_RADIUS

Controls how the step size is scaled. =TRUST_RADIUS uses the dynamic scaling by Fletcher, =NORM uses the absolute step length, and =MAXIMUM uses the largest individual step in the internal coordinate space.

EIGENVECTOR = (integer) 1

Sets which eigenvector of the totally symmetric part of the block-factored Hessian is to be followed *uphill* in a transition state search. Eigenvectors are indexed by their

eigenvalues (the lowest eigenvalue is 1, the next lowest is 2, etc.). The default should *always* be used if you are not looking for a specific transition state which you know corresponds to motion along a different mode. The value of EIGENVECTOR has no meaning if OPT_METHOD is not set to EVFTS or MAEVFTS.

NEGEVAL = (handle) RFA

Specifies the behavior if negative eigenvalues are encountered in the totally symmetric Hessian during an NR or MANR search. If NEGEVAL=ABORT, then the job will terminate with an error message. If NEGEVAL=ABSVAL, then the program will just switch the eigenvalue to its absolute value and keep plugging away (this is strongly discouraged). If NEGEVAL=RFA, then OPT_METHOD is switched to RFA internally and the optimization is continued.

8.1.24 Geometry optimization: iteration control

CONVERGENCE = (tol) 4

Sets the convergence criterion for geometry optimizations. The optimization terminates when the RMS gradient is below 10^{-N} Hartree/Bohr.

OPT_MAXCYC = (integer) 50

Sets the maximum number of optimization cycles.

INIT_HESSIAN = (handle) SPECIAL

Specifies the initial Hessian for geometry optimizations. =SPECIAL generates an approximate Hessian internally, =FCMINT imports an approximate Hessian from the FCMINT file, =MOPAC generates the FCMINT file by running MOPAC, and =EXACT uses analytical second derivatives to generate an exact Hessian (limited to SCF).

HESS_UPDATE = (handle) POWELL (or BOFILL)

Specifies the algorithm used to update the Hessian. =NONE uses the same Hessian throughout the optimization, =POWELL uses the Powell update, =BFGS uses the Broyden-Fletcher-Goldfarb-Shanno update, =MS uses the Murtagh-Sargent update, =BOFILL uses the Bofill mixture of Powell and Murtagh-Saregent, and =PSB uses the Powell symmetric Broyden update. =POWELL and =BOFILL are the defaults for minimum energy and transition state searches, respectively.

EVAL_HESS = (integer) 0

Sets the cycle interval for recomputing the Hessian. For correlated calculations, the Hessian is evaluated *only at the SCF level*.

8.1.25 Geometry optimization: integral derivatives

TRANS_INV = (handle) USE

Specifies whether or not translational invariance is exploited in derivative calculations. =USE uses translational invariance, while =IGNORE does not.

8.1.26 Frequencies and other 2^{nd}-order properties

VIBRATION = (handle) NO

Specifies the method for calculating vibrational frequencies. =EXACT (SCF only) performs normal mode analysis on an analytic force constant matrix and computes rotationally projected frequencies and infrared intensities. =FINDIF signals ACES II to compute the force constant matrix by finite difference of analytically computed gradients or energies using symmetry-adapted mass-weighted Cartesian coordinates.

RAMAN = (switch) OFF

Controls raman intensities and depolarization ratios of numerical vibrational frequency calculations (VIB=FINDIF). The Raman intensities are limited to the SCF and CCSD level (via EOM-CCSD), and as a result RAMAN keyword must be used in conjunction with CALC=SCF or CALC=CCSD. We recommend setting EOMPROP=QUADRATIC for raman intensities at the CCSD level.

[NOTE: This method requires an extra input file named `frequency`, which is described in Section 6 (page 22).]

8.1.27 Finite displacements

(for numerical gradients and Hessians)

FD_STEPSIZE = (integer) 50

Sets the step length (in 10^{-4} amu$^{1/2}$ Bohr) used in generating the force constant matrix by finite difference of Cartesian gradients. (The default is 0.005 amu$^{1/2}$ Bohr.)

FD_IRREPS = (string of 1D array) 0

Lists the symmetry types to be evaluated in a VIBRATION=FINDIF calculation. The numbers of the irreducible representations for which vibrational analysis is to be performed are separated by forward-slashes. For example, FD_IRREP=1/3/4 computes the frequencies of modes transforming as the first, third, and fourth irreducible representations. If a symmetry is specified for which there are no vibrational modes, the program will terminate. The labels of the irreducible representations for this keyword

are *not* usually the same as those used in the symmetry-adapted integrals. Moreover, for some point groups (like those of linear molecules), the two sets of labels refer to different subgroups. There is still no straightforward way to determine what they will be without starting a calculation. One JODA run will list the relevant irreducible representations. If all vibrational frequencies are desired, this keyword need not be included. The default is to calculate all irreps.

FD_PROJECT = (switch) ON

Controls whether or not rotational degrees of freedom are projected out of symmetry-adapted coordinates. =ON uses rotationally projected coordinates, while =OFF retains the rotational degrees of freedom. At a stationary point on the potential surface, both options will give equivalent harmonic force fields, but =OFF should be used at non-stationary points.

FD_USEGROUP = (handle) FULL

Specifies the point group to be used in generating the symmetry-adapted vibrational coordinates. =FULL specifies the full molecular point group, =COMP specifies the Abelian subgroup used in the electronic structure calculation.

POINTS = (handle) DOUBLE

Specifies either single- (=SINGLE) or double-sided (=DOUBLE) numerical differentiation in the finite difference evaluation of the Hessian. Two-sided numerical differentiation is considerably more accurate than the single-sided method, and its use is *strongly recommended* for production work.

8.1.28 External interfaces

EXTERNAL = (handle) NONE

Specifies what type of external file interface for third-party programs to create. Current interfaces include =HYPERCHEM and =MOLDEN.

A2PROC *help*

This is not a keyword *per se* but is used by ACES II to generate the external interfaces, and it can be used to generate such interfaces after a calculation has finished.

9 Examples

9.1 Single-point calculations

9.1.1 RHF CCSD(T) energy

```
H2O. CCSD(T) energy calculation. DZP basis set.
H
O 1 ROH
H 2 ROH 1 HOH

ROH=0.95
HOH=104.5

*ACES2(CALC=CCSD(T),BASIS=DZP)
```

This illustrates a single-point CCSD(T) energy calculation on the water molecule with the DZP basis set. This input is particularly simple since all of the keywords default to the closed-shell singlet state of the neutral molecule. In other words, CHARGE=0, MULTIPLICITY=1, and REFERENCE=RHF can all be omitted. The OCCUPATION string is not specified, which means the program will determine this itself. By default all electrons are correlated.

9.1.2 UHF CCSD(T) energy

```
H2O+. UHF-CCSD(T) energy calculation. DZP basis set.
H
O 1 ROH
H 2 ROH 1 HOH

ROH=0.95
HOH=104.5

*ACES2
REF=UHF,CALC=CCSD(T)
OCCUPATION=3-1-1-0/2-1-1-0

H:DZ
O:DZP
H:DZ
```

This illustrates a single-point CCSD(T) energy calculation on the $^2A_1$ state of the water cation with a special basis set. For this system, one cannot rely on the defaults that were used in the previous example. It is necessary to specify REFERENCE and either OCCUPATION or both CHARGE and MULTIPLICITY. Since we want a particular state, we use the OCCUPATION keyword.

The `*ACES2` namelist is valid even without the initial and final parentheses; however, if an open parenthesis is used to start the namelist then it must terminate with a close parenthesis.

9.1.3 ROHF CCSD(T) energy

```
H2O+. ROHF-CCSD(T) energy calculation. DZP basis set.
H
O 1 ROH
H 2 ROH 1 HOH

ROH=0.95
HOH=104.5

*ACES2
REF=ROHF,CALC=CCSD(T),BASIS=DZP
OCCUPATION=3-1-1-0/2-1-1-0
```

This is essentially identical to the previous example. The difference is that the user requests a restricted open-shell Hartree-Fock reference. The program automatically turns on the NON-HF option so the appropriate "non-Hartree-Fock" terms are included in the coupled-cluster equations. It also automatically sets ORBITAL=SEMICANONICAL, which is needed to evaluate triple excitations non-iteratively, as in the CCSD(T) method.

9.1.4 QRHF CCSD(T) energy

```
H2O+. QRHF-CCSD(T) energy calculation. DZP basis set.
H
O 1 ROH
H 2 ROH 1 HOH

ROH=0.95
HOH=104.5

*ACES2
REF=RHF,CALC=CCSD(T),BASIS=DZP
QRHF_GENERAL=-1
ORBITAL=SEMICANONICAL
OCCUPATION=3-1-1-0/3-1-1-0
```

This is another way in which one can calculate the energy of an open-shell molecule. Here, the energy of the $^2A_1$ state of the water cation is being calculated by the QRHF-CCSD(T) method. The orbitals, however, are not from an SCF calculation on H_2O^+. Rather, they are orbitals from the neutral molecule. In general, QRHF means orbitals are taken from a convenient closed-shell system and are then used in an open-shell system.

An important difference between this example and the previous examples is that the CHARGE, MULT, REF, and OCCUPATION keywords *do not* refer to the system being studied (H_2O^+), but instead refer to the system from which the orbitals are obtained (H_2O). QRHF_GENERAL=-1 (along with the default values for QRHF_ORBITAL and QRHF_SPIN) means the reference function for the correlated calculation is to be formed by removing a β-spin electron from the highest occupied orbital of symmetry 1 (A_1).

As in the ROHF example, the program automatically turns on the NON-HF option so that the appropriate "non-Hartree-Fock" terms are included in the coupled-cluster equations.

9.1.5 Effective core potentials

```
CRF6 SINGLE POINT ENERGY CALCULATION USING AN ECP FOR CR
CR
F 1 RMC
F 1 RMC 2 W1
F 1 RMC 3 W1 2 T1
F 1 RMC 4 W1 3 T2
F 1 RMC 5 W1 4 T1
F 1 RMC 6 W1 5 T2

RMC=1.676125
W1=90.
T1=90.
T2=270.

*ACES2(BASIS=SPECIAL,ECP=ON,CALCLEVEL=SCF
OCC=10-6-6-2-6-2-2-0,SPHERICAL=ON)

CR:SBKJC
F:DZP
F:DZP
F:DZP
F:DZP
F:DZP
F:DZP

CR:SBKJC
F:NONE
F:NONE
F:NONE
F:NONE
F:NONE
F:NONE
```

The use of effective core potentials is specified in the `ZMAT` file by the keyword ECP=ON. The keyword BASIS must be set to BASIS=SPECIAL. The ecp nicknames are listed below the last basis set (separated by a blank line) using the same format as the non-standard basis set specification, XX:ECPNAM. As for the `GENBAS` file, the `ECPDATA` file may be searched for XX, where XX is the atomic symbol, to show what ECPs are available for that atom. Atoms included in the `ZMAT` file without an ecp parameter set are marked by the nickname NONE.

9.1.6 Initial guessing with `OLDMOS`

Suppose the user wishes to run an ROHF calculation, but the default ROHF procedure does not converge or converges to the wrong electronic state. The user could try to start the ROHF calculation from a converged set of UHF orbitals.

First, run a UHF calculation. `xvscf` writes the UHF orbitals to a small formatted file called `NEWMOS`. These are expressed in terms of the symmetry-adapted AO basis functions

and are printed by spin and symmetry block. To run an ROHF calculation starting from these UHF orbitals:

1. copy NEWMOS to OLDMOS in a new directory
2. copy in ZMAT and change REF=UHF to REF=ROHF
3. add GUESS=READ_SO_MOS to the *ACES2 namelist
4. copy in ZMAT.BAS (or GENBAS)
5. run xaces2

The UHF job:

```
cd /usr/var/tmp/yau/scr
rm *
cp ~yau/nh2-uhf.zmt ZMAT
cp ~yau/GENBAS GENBAS
xaces2 > ~yau/nh2-uhf.out
cp NEWMOS ~yau/nh2-uhf.mos
```

The ROHF job:

```
cd /usr/var/tmp/yau/scr
rm *
cp ~yau/nh2-rohf.zmt ZMAT
cp ~yau/nh2-uhf.mos OLDMOS
cp ~yau/GENBAS GENBAS
xaces2 > ~yau/nh2-rohf.out
cp NEWMOS ~yau/nh2-rohf.mos
```

9.1.7 Initial guessing with OLDAOMOS

The main use of this option is in finite-difference vibrational frequency calculations. In these calculations, it is not possible to use the OCCUPATION keyword since the symmetry can change at various displacements. Therefore, the SCF program is not always able to converge to the correct electronic state at each geometry, and sometimes the frequency calculation cannot be completed. The GUESS=READ_AO_MOS option is intended to solve this problem.

Users begin by performing a single-point SCF calculation at the geometry at which vibrational frequencies are to be calculated. In this calculation, the occupation can be set to obtain the correct electronic state. After the SCF, xvscf creates a formatted file called AOBASMOS. Before the frequency calculation, the OCCUPATION keyword must be removed from *ACES2, the option GUESS=READ_AO_MOS must be set, and the AOBASMOS file must be renamed OLDAOMOS. At each point in the frequency calculation, the MOs are read from

OLDAOMOS and transformed to the current symmetry. The occupation is determined in the current point group, and since the displacements from the "reference geometry" are small, the initial guess is usually very good.

It is also possible to use the final AOBASMOS file from a geometry optimization (or transition state search) as the OLDAOMOS file in a frequency calculation. The GUESS=READ_AO_MOS option can be used in other situations, but it might not work well when the geometry changes significantly. The reason for this is that a transformation matrix relating the different geometries must be calculated, and this is only approximated well by a rotation if the geometries are close.

9.1.8 Improving SCF convergence

VSCF contains a number of options for accelerating and controlling the SCF convergence. By default the first few iterations proceed by repeated diagonalization of appropriate Fock matrices. Once either a certain number of iterations have been performed (specified by RPP_LATEST) or an initial convergence criterion has been met, the DIIS convergence extrapolation procedure of Pulay begins. If convergence difficulties are experienced with the default scheme, the user has a number of options. In RHF closed-shell and UHF open-shell calculations, difficult cases will often converge with the use of a dynamical damping algorithm due to E.R. Davidson. This is achieved through the option DAMP_TYP=DAVIDSON. Damping serves to prevent excessively large oscillations in the early iterations. Once the SCF convergence appears to be sufficiently smooth (the damp factor is smaller than DAMP_TOL and the energy difference is sufficiently small), the program reverts to repeated diagonalization and DIIS extrapolation. For ROHF calculations, one can also use the damping algorithm, but in addition the level-shifting technique (M.F. Guest and V.R. Saunders, *Mol. Phys.* **28**, 819 (1974)) is particularly useful. In this scheme one adds a positive number α to all diagonal elements of the singly occupied orbital block of the Fock matrix and a positive number $(\alpha + \beta)$ to the diagonal elements of the virtual orbital block of the MO basis Fock matrix. α and β are set by the LSHF_A1 and LSHF_B1 keywords. If one wishes to use a value 0.2 a.u. for the level-shifters, LSHF_A1 and LSHF_B1 should be set to 20. This is a reasonable value for most systems. Larger values of level shifters are sometimes necessary for transition metal systems, especially when the singly occupied orbitals lie below some of the double occupied orbitals. An example is provided by the following $FeCl_4^-$ input: (excitation energies of this system were studied by N. Oliphant and R.J. Bartlett, *J. Am. Chem. Soc.* **116**, 4091 (1994))

```
FeCl4 sextet
FE  0.0000  0.0000  0.0000
CL  0.0000  3.4826  2.2358
CL  0.0000 -3.4826  2.2358
```

```
CL  3.4826  0.0000 -2.2358
CL -3.4826  0.0000 -2.2358

*ACES2
REF=ROHF,PRINT=1,CALC=SCF,MULT=6,CHARGE=-1
BASIS=SPECIAL,UNITS=BOHR,COORDINATES=CARTESIAN
SPHERICAL=ON,ECP=ON
DAMP_TYP=DAVIDSON,DAMP_TOL=5,LSHF_A1=50,LSHF_B1=50
OCCUPATION=7-6-6-6/5-5-5-5

FE:SBKJC
CL:SBKJC
CL:SBKJC
CL:SBKJC
CL:SBKJC

FE:SBKJC
CL:SBKJC
CL:SBKJC
CL:SBKJC
CL:SBKJC
```

Another tip for ROHF convergence, which the developers *strongly* recommend, is starting the SCF from UHF orbitals from a closely related closed-shell system. (UHF is necessary since ROHF reads both α and β orbitals.)

9.1.9 Hartree-Fock stability analysis

The Hartree-Fock procedure, at convergence, guarantees the resulting wavefunction is a stationary point in the space of orbital rotations (mixing of occupied and virtual orbitals). Though in the majority of cases this stationary point is also a minimum (all orbital rotations increase the energy), it may not always be so. In some cases, the second derivative of the energy with respect to one or more orbital rotations may be zero or negative, indicating rotations which will leave the energy unchanged or lower it. The ACES II program system has the ability to test RHF and UHF wavefunctions for some of the most common instabilities, controlled by the HFSTABILITY and ROT_EVEC keywords.

Using HFSTABILITY=ON will perform a stability analysis after the two-electron integral transformation and processing. (Use of HFSTABILITY=ON is compatible with continuing on to correlated calculations in the same job.) Stability analysis is accomplished by forming the orbital rotation hessian and diagonalizing it. The eigenvalues of this matrix, and their associated eigenvectors indicate the number and type of instabilities present.

Each negative eigenvalue indicates an instability, and their magnitude indicates the severity. Analysis of the eigenvector corresponding to each instability reveals its nature. The direct product of the symmetry irreps of the orbitals involved in the rotation determines the symmetry of the instability. Only for instabilities whose direct product is the totally symmetric irrep (irrep 1) will the wavefunction maintain the symmetry of the molecular

framework. Any other result means the instability leads, at least initially, to a symmetry broken wavefunction. In some cases, symmetry breaking instabilities arise from the presence of lower energy electronic states (different occupations), and the rotation specified by the corresponding eigenvector will correspond to changing the occupation to relieve the instability. For an RHF wavefunction, the UHF orbital rotation hessian is constructed, which allows detection of instabilities in which the wavefunction would prefer to be UHF by comparison of the α and β spin eigenvectors.

Zero eigenvalues will occur for degenerate electronic states and merely indicate the equivalence of occupations within the computational point group. Small numerical inaccuracies frequently result in nominally zero eigenvalues having small non-zero values. The sign of such eigenvalues will determine whether or not they are reported as instabilities. Thus the program might not show the expected number of zero (small) eigenvalues.

Other instabilities, such as the wavefunction becoming complex are not tested since complex wavefunctions are not presently supported in ACES II.

In the program output, stability analysis is headed by the label "RHFSTAB" or "UHFSTAB". The number of instabilities in each irrep is given along with their eigenvalue and classification. For example,

```
@RHFSTAB, Performing stability analysis of RHF wavefunction.
Orbital rotation parameters will be evaluated for each symmetry block.
There are   0 instabilities within irrep   1.
There are   1 instabilities within irrep   2.
 Eigenvalue =   -0.1411211526:
 Instability classification :  RHF ->  UHF with broken symmetry
There are   1 instabilities within irrep   3.
 Eigenvalue =   -0.1411211526:
 Instability classification :  RHF ->  UHF with broken symmetry
There are   0 instabilities within irrep   4.
There are   0 instabilities within irrep   5.
There are   2 instabilities within irrep   6.
 Eigenvalue =   -0.3586792381:
 Instability classification :  RHF ->  UHF with broken symmetry
 Eigenvalue =   -0.2166464610:
 Instability classification :  RHF ->  RHF with broken symmetry
There are   2 instabilities within irrep   7.
 Eigenvalue =   -0.3586792381:
 Instability classification :  RHF ->  UHF with broken symmetry
 Eigenvalue =   -0.2166464610:
 Instability classification :  RHF ->  RHF with broken symmetry
There are   0 instabilities within irrep   8.
```

It is sometimes desirable to obtain solutions corresponding to following instabilities to lower energy stationary points in the orbital rotation space. The keywords HFSTABILITY=FOLLOW and ROT_EVEC are provided to assist with this.

When HFSTABILITY=FOLLOW is set, the stability analysis is performed, as for HFSTABILITY=ON, then the orbital rotation corresponding to the chosen instability is applied

to the SCF eigenvectors and the SCF calculation is repeated with these rotated vectors as a starting guess. This is not strictly eigenvector following, nor direct minimization SCF, but in practice, the procedure is quite effective. By default, the lowest eigenvalue of the totally symmetric irrep is followed. Others can be followed by explicitly specifying them with the ROT_EVEC parameter.

Because instabilities in other than the totally symmetric irrep reduce the symmetry of the wavefunction, only those in irrep 1 can be followed. Following other instabilities requires performing the calculation in reduced symmetry. Note that in C_1 symmetry, *all* instabilities will be in the totally symmetric irrep. Likewise, instabilities which take RHF wavefunctions into UHF ones must be followed using a UHF calculation.

Following instabilities often leads to solutions which are lower in energy but lack the symmetry of the molecular framework, are heavily spin contaminated (in the UHF case) or are otherwise non-physical. Instabilities in Hartree-Fock wavefunctions are a very complex problem and must be treated with great care in order to obtain any meaningful information.

Experience has shown that, in general, molecules with instabilities will have more than one and that solutions obtained by following an instability may have further instabilities. It is impossible to predict exactly what will be uncovered by a stability analysis, and the current drivers for ACES II are configured to handle only the simplest cases. Examining a "tree" of instabilities or other specialized procedures will require the user to create an appropriate driver of their own, which is usually specialized to the system under study.

A suprising number of molecules will exhibit instabilities under various conditions. If you encounter one, don't panic. Take a deep breath, go for a walk, get a cup of coffee. Not all instabilities indicate pathological cases or intractable problems.

9.1.10 Time-dependent Hartree-Fock

In order to provide properties such as frequency dependent polarizabilities, ACES II provides time-depenent Hartree-Fock (TDHF) calculations. The TDHF code can solve the general-order TDHF problem for closed-shell RHF wavefunctions. The TDHF keyword must be set to ON, and additional parameters controlling the TDHF calculation are included in a namelist which is located at the end of the `ZMAT` file. The variables in the namelist are as follows :
Printing options :

- IOPDA controls density matrix printing (0 means no printing (default) and 1 means print the matrix)
- IOPEV controls MO coefficient printing (0 means no printing (default) and 1 means print the matrix)

- IOPU controls U matrix printing (0 means no printing (default) and 1 means print the matrix)
- IOPFE controls Fock matrix printing (0 means no printing (default) and 1 means print the matrix)
- IOPPR controls property integral printing (0 means no printing (default) and 1 means print the matrix)

For subsequent options, 0 means do not do this type of calculation, otherwise do it. *The default is to do the calculation, so if a parameter is not specified, the program will perform that calculation.*
For the polarizability α :

- IDALPH specifies if α is to be calculated

For the first hyperpolarizability β :

- IOR specifies if optical rectification calculation is to be performed
- IEOPE specifies if electro-optical Pockels effects calculation is to be performed
- ISHG specifies if second harmonic generation calculation is to be performed

For the second hyperpolarizability γ :

- IOKE specifies if an optical Kerr effect calculation is to be performed
- IDCOR specifies if a DC electric field induced OR calculation is to be performed
- IIDRI specifies if am intensity dependent refractive index calculation is to be performed
- IDCSHG specifies if DC electric field SHG calculation is to be performed
- ITHG specifies if third harmonic generation calculation is to be performed

The method of solution of the TDHF equations is controlled by the parameter NITER (default value is 20). If NITER is 0 an iterative method of solution is used; if it is 1 a non-iterative method of solution is used; if it is greater than 1 the reduced linear equation method will be used.

When NITER=1 more than one frequency can be solved for. For the other methods of solution, only one frequency can be considered in a single calculation. The number of

nonzero frequencies is specified by NFREQ. The NFREQ frequencies are listed one per line following the INPUTP namelist.

Static calculations are also performed along with the dynamic calculations. To obtain only static results, all parameters should be set to 0.

The following is an example of a TDHF calculation on N_2 :

```
N2. TDHF. TEST. Sadlej basis set.
N
N 1 R

R=2.07434

*ACES2(UNITS=BOHR,BASIS=SPECIAL,TDHF=ON)

N:PBS
N:PBS

 $INPUTP IOPU=0,IOPFE=0,IOPEV=0,IOPPR=0,NITER=1,NFREQ=4,IDCSHG=1,IOKE=1,
IDCOR=1,IIDRI=1,ITHG=1,IWRPA=1 $END
 0.072
 0.0886
 0.0934
 0.0995
```

WARNING: There does not seem to be a portable format for Fortran namelists. The code that parses `ZMAT` contains an error handler that shows the expected format in the event a read error occurs. For TDHF calculations in particular, the developers suggest running a small test calculation to verify the proper formats are used.

9.1.11 EOM-CCSD excitation energy

```
EOM-CCSD excitation energies and transition moments for water
H
O 1 R
H 2 R 1 A

R=0.957
A=104.5

*ACES2(BASIS=TZ2P,CALC=CCSD,EXCITE=EOMEE,ESTATE_SYM=1/1/1/1
ESTATE_PROP=EXPECTATION)
```

This example specifies an equation of motion coupled-cluster excitation energy calculation for the water molecule using the TZ2P basis set. The program will attempt to find the lowest root in each of the four symmetry species of the C_{2v} point group, and oscillator strengths will be evaluated for all excited states.

9.1.12 EOM-CCSD electron attachment energy

EOM-CC electron attachment calculations yield energy differences between an N-electron reference state and one or more electronic states of the (N+1) electron system obtained by

adding an electron. The keywords such as REFERENCE, CHARGE, MULTIPLICITY, and OCCUPATION define the electronic state of the N electron system. If EA_CALC is set to EA_EOMCC, then energies of (N+1) electron states are calculated. The states are specified by the EA_SYM keyword as a string of NIRREP (REFERENCE=RHF) or 2*NIRREP (REFERENCE=UHF or ROHF) integers, where NIRREP is the number of irreducible representations in the computational point group. The string of numbers specifies the numbers of (N+1) electron states of a given symmetry and the spin of the additional electron in each (N+1) electron state. For closed-shell systems only the alpha-roots have to be specified. EA_SYM=3-2-0-2, for example. For open-shell systems one can either attach an electron of alpha spin or one of beta spin, leading to different states of the (N+1)-electron system. The different spin blocks are separated by a slash ('/') as in EA_SYM=3-2-0-2/0-1-0-4. This keyword does not have to be specified in an EA-EOMCC calculation. If EA_SYM is not specified but EA_CALC=EA_EOMCC, the program tries to find the ground state of the (N+1)-electron system internally.

We can recommend three types of applications of the EA-EOMCC program:

1. The calculation of electron affinities. Only EA_CALC=EA_EOMCC needs to be specified. If it is known what the symmetry of the ground state is for the (N+1)-electron system, specify also EA_SYM. The following input yields the electron affinity of the sodium atom.

   ```
   NA atom
   NA

   *ACES2(REFERENCE=UHF,CALC=CCSD,BASIS=DZP
   MULTIPLICITY=2,SPHERICAL=ON,EA_CALC=EA_EOMCC)
   ```

   The keyword EA_SYM=0-0-0-0-0-0-0-0/1-0-0-0-0-0-0-0 might have been specified such that only the closed-shell $3s^2$ state of the sodium anion is calculated, and no other possibilities are considered for the symmetry of the anion ground state.

2. The calculation of excitation spectra for systems with an odd number of electrons. Take as a reference a closed-shell configuration of the system with one less electron. Specify EA_SYM to obtain a number of roots of desired symmetry. The excitation spectrum is then calculated.

   The following example specifies the input for calculation of the excitation spectrum of MgF. The two core-orbitals are excluded from the correlation treatment (Both CCSD and EA_EOMCC). 5 roots are found in symmetry block 1 (Σ and Δ symmetries), 3 in block 2 (Π symmetry) and one in block 4.

```
MgF Excitation Spectrum
MG
F 1 R

R = 1.752

*ACES2(REFERENCE=RHF,CALC=CCSD,BASIS=TZ2P,CHARGE=1
MULT=1,SPHERICAL=ON,DROPMO=1-2,EA_CALC=EA_EOMCC EA_SYM=5-3-0-1)
```

3. The calculation of high spin triplet states for systems with a closed-shell ground state. Take as a CC reference the high spin doublet ground state of the positive ion, and add an extra alpha electron by specifying EA_SYM (e.g., EA_SYM=4-3-0-2/0-0-0-0). This yields high spin triplet excited states of the neutral. In addition the closed-shell ground state can be obtained by adding a beta electron to the proper symmetry-block (e.g., EA_SYM=4-3-0-2/1-0-0-0). This has the advantage that the proper excitation energies of the system are tabulated by the program. We note that singlet and low spin triplet excited state energies can also be obtained by adding a beta electron. However such calculations do not yield satisfactory results, due to spin contamination of the resulting EA_EOMCC states.

   The following input yields triplet excited states for the beryllium atom. The SCF calculation is on the closed-shell neutral system, while the QRHF option is used to create the positive Be ion.

```
BE Atom Excitation spectrum, QRHF Reference
BE

*ACES2(REFERENCE=RHF,CALC=CCSD,BASIS=SPECIAL
SPHERICAL=ON,QRHF_G=(-1),EA_CALC=EA_EOMCC
EA_SYM=8-4-0-4-0-0-0-0/0-0-0-0-0-0-0-0)

BE:WMR
```

   WARNING: In all EOMCC calculations it is highly recommended that the reference state transforms according to a one-dimensional representation of the true molecular point group. Otherwise, the likely outcome is inaccurate results, which in addition are very hard to interpret due to a breaking of the symmetry in such a calculation.

9.1.13 NMR chemical shifts

NMR spectroscopy is a very important analytical tool for the identification and characterization of molecules. However, since there is no simple correlation between the measured chemical shifts and structural parameters, the interpretation of experimental NMR spectra is not trivial and can be in many cases quite involved. The ability to calculate NMR chemical shifts *ab initio* is therefore a very important advancement in quantum chemistry. The

calculation of chemical shifts can provide in many cases the necessary information for the correct interpretation of experimental NMR spectra. Therefore, methods for the computation of NMR chemical shifts at SCF and correlated levels (currently limited to MBPT(2)) have been added in the last years to the functionalities of the ACES II program system.

However, before discussing the details of the calculation of NMR chemical shifts, a few general remarks are required. The main problem in all calculations of magnetic properties (i.e., NMR chemical shifts and magnetizabilities) using finite basis sets (as they are usually employed in quantum chemical calculations) is the gauge-invariance problem. This simply means that the results of such a calculation depend on the chosen gauge-origin and are not invariant with respect to gauge transformations as required by exact theory. A trivial solution to the gauge-invariance problem would be the use of very large basis sets in order to minimize the gauge error, but this approach, due to large computational costs, is limited to small molecules. More satisfying solutions are offered by approaches which introduce local gauge-origins to define the vector potential such as the IGLO method of Kutzelnigg and Schindler, the LORG method of Hansen and Bouman and the GIAO-approach of Ditchfield. The latter originates in London's work on molecular diamagnetism in the thirties and was first used by Hameka during the sixties to calculate magnetizabilities and chemical shieldings. ACES II incorporates the GIAO-SCF and GIAO-MBPT(2) method for calculating chemical shifts, since in our opinion, the GIAO approach is the most elegant solution to the gauge-invariance problem and in contrast to the IGLO method, is easily extended to correlated approaches (as for example in the GIAO-MBPT(2) method).

The calculation of NMR chemical shifts are invoked via the keyword PROP=NMR together with the appropriate specification of the quantum chemical method (CALC=SCF gives then GIAO-SCF, CALC=MBPT(2) gives GIAO-MBPT(2)). In principle, no other option is required to run calculations of NMR chemical shifts. However, to ensure the success of GIAO-MBPT(2) calculations and in particular large-scale calculations with the GIAO-MBPT(2) method, the computational requirements of such calculations should be kept in mind. While CPU requirements are of less interest (a GIAO-MBPT(2) calculation is in terms of the CPU usually less expensive than the corresponding MBPT(2) geometry optimization), memory and disk space requirements are of special concern. The memory requirements are for the `xnmr` module approximately $2 * n^2N^2/h^2$ eight byte words with n denoting the number of occupied orbitals, N denoting the number of virtual orbitals, and h specifying the order of the molecular point group. However, the current memory bottleneck is the integral sorting in the module **xintprc** which requires roughly $2*n^2N^2/h$ eight byte words and which therefore is more demanding. The disk space requirement of a GIAO-MBPT(2) calculation is for most parts determined by the fact that the current version depends on the storage of the GIAO integrals. To summarize the necessary resources, a GIAO-MBPT(2) calcula-

tions requires the storage of the AO two-electron integrals (files IIII, IIJJ, IJIJ, and IJKL depending on symmetry and with a total size of approximately $1.5 * N_{basis}^4/(8h)$ eight byte words (note that the CRAY version requires $2 * N_{basis}^4/(8h)$ words), the storage of the MO integrals (file MOINTS, the largest portion is here given by the integrals $\langle ab||ci\rangle$ with a total size nN^3/h eight byte words), the storage of the GIAO integrals (files IIIIX, IIJJX, IJIJX, IJKLX in case of a symmetric perturbation B_x or IIIJX, IJIKX, IJKLX for a non-symmetric perturbation B_x. The files for the perturbations B_y and B_x are named using the same convention with Y or Z instead of X. The total size of these files is for each perturbation approximately given by $1.5*N_{basis}^4/(4h)$ (CRAY version : $2*N_{basis}^4/(4h)$)), and the storage of the derivative MO integrals and amplitudes (files DERGAM and DERINT with the largest portion given by the perturbed $\langle ab||ci\rangle$ integrals (size nN^3/h eight byte words)). To keep the disk space requirements at a minimum, it is *strongly recommended* to use the keyword TREAT_PERTURBATION=SEQUENTIAL, as this forces the program to treat each magnetic field component separately and thus requires only the storage of one type of GIAO integrals at one time. TREAT_PERTURBATION=SIMULTANEOUS (which is currently the default) requires the simultaneous storage of all GIAO-integrals and is therefore much more demanding in terms of disk space, though more efficient with respect to CPU timings.

The current limits for GIAO-MBPT(2) calculations depend on the available hardware resources. However, as a rough guide the limits for a IBM RS6000/350 work station with 80 Mbyte memory and 3.5 Gbyte scratch space are given. Within this environment, calculations with about 250 basis functions in D_{2h} symmetry, about 200 basis functions in C_{2v}, C_{2h}, and D_2 symmetry, about 170 basis functions in C_2, C_i, and C_s-symmetry and about 130 basis functions in cases without symmetry are feasible. From these estimates it is seen that the limits strongly depend on the molecular symmetry. Thus, users are urged to use symmetry whenever possible. It should be also noted that the computational requirements depend somewhat on the ratio n/N and that the costs increases with the number of occupied orbitals.

As the final result the corresponding chemical shielding tensors of all nuclei in the molecule are obtained in a GIAO-SCF and/or GIAO-MBPT(2) calculation. The output produced by the module **xjoda** gives the absolute isotropic shielding (one third of the trace of the shielding tensor) as well as the anisotropic shielding which is usually of less interest. In order to compare with experimental results, the absolute shielding σ must be converted to the relative shifts δ. This is easily accomplished via

$$\delta = \sigma_{ref} - \sigma \tag{1}$$

with σ_{ref} as the absolute shielding of the chosen reference compounds. For ^{13}C, tetramethylsilane (TMS) has been chosen as standard and the corresponding σ values are 198.944 (dzp/dz), 193.103 (tzp/dz), 193.419 (tzp/dzp) at GIAO-SCF level and 205.720 (dzp/dz),

198.890 (tzp/dz), and 197.191 (tzp/dzp) at GIAO-MBPT(2) level. The MBPT(2)/6-31G* optimized geometry has been used in all the GIAO calculations for TMS.

With respect to basis sets, the following recommendations can be made. In case of ^{11}B and ^{13}C – nuclei which have been extensively studied – the dzp/dz basis set (dzp for C and dz for H) is in most cases sufficient for relative shifts, though the use of the tzp/dz basis (tzp for C and dz for H) is recommended. Larger basis sets (i.e., tzp, tzp2, or even qz2p) are in most cases not needed for the accurate prediction of chemical shifts. On the other side, ^{15}N, ^{17}O, and ^{19}F NMR chemical shift calculations require larger basis sets of at least triple-zeta plus polarization quality. Though qualitatively good results are obtained in many cases with the tzp basis, even larger basis sets such as tz2p or qz2p are recommended for more reliable calculations. For other nuclei, not very much can be said at the moment and the user is strongly urged to check carefully the basis set dependence to ensure reliable theoretical results. However, limited experience suggests that for second-row elements quite large basis sets are needed for accurate calculations.

Note : NMR chemical shift calculations can currently only be performed with Cartesian basis functions—SPHERICAL=ON may not be specified.

```
GIAO-MBPT(2) NMR chemical shift calculations for the benzonium cation
C           0.000000    0.000000   -1.393615
C           0.000000    0.000000    1.413386
C           0.000000   -1.250334   -0.632255
C           0.000000    1.250334   -0.632255
C           0.000000   -1.237545    0.742530
C           0.000000    1.237545    0.742530
H           0.853412    0.000000   -2.102274
H          -0.853412    0.000000   -2.102274
H           0.000000   -2.189398   -1.180688
H           0.000000    2.189398   -1.180688
H           0.000000   -2.161755    1.310996
H           0.000000    2.161755    1.310996
H           0.000000    0.000000    2.502007

*ACES2(CALC=MBPT(2),PROP=NMR,CHARGE=1
COORD=CARTESIAN,TREAT_PERTURBATION=SEQUENTIAL,MEMORY=24000000)

C:DZP
C:DZP
C:DZP
C:DZP
C:DZP
C:DZP
H:DZ
H:DZ
H:DZ
H:DZ
H:DZ
H:DZ
H:DZ
```

This example features a calculation of the NMR chemical shifts at the MBPT(2) level

using gauge-including atomic orbitals (GIAOs). NMR shifts are requested by the keyword PROP=NMR. The input of the coordinates in this example is via Cartesian coordinates (in Ångströms) and the basis set is specified via the non-standard input which allows the use of different basis sets for different atoms (in this case, DZP for C and DZ for H). To reduce disk space requirements, the keyword TREAT_PERTUBATION=SEQUENTIAL is specified which forces the program to treat each magnetic field perturbation separately, thus storing only GIAO integrals of one particular type at one time. The memory requirement in this example is set to 24000000 integer-words (91.5MB on 32-bit builds and 183.1MB on 64-bit builds).

9.2 Geometry optimizations

9.2.1 Full optimization of internal coordinates

```
H2O. CCSD optimization.
H
O 1 ROH*
H 2 ROH* 1 HOH*

ROH=0.95
HOH=104.5

*ACES2(CALC=CCSD,BASIS=DZP)
```

This specifies a geometry optimization of the water molecule with the CCSD method and the DZP basis set. The input is exactly the same as the RHF single-point energy example, except that an asterisk (∗) is placed after each variable to be optimized (both bond lengths and the bond angle). Since all of the parameters are being optimized, it is also sufficient to use GEOM_OPT=FULL and leave the asterisks out of the Z matrix.

Defaults for all of the optimization keywords (e.g., METHOD and CONVERGENCE) are used. The presence of the asterisks promotes the GEOM_OPT keyword from NONE to PARTIAL, which turns on all appropriate derivative keywords. Geometry optimizations by default are performed using analytical gradients. If analytical gradients are not available, one can do geometry optimizations from energies using GRAD_CALC=NUMERICAL.

9.2.2 Partial optimization of internal coordinates

```
Beryllium borohydride, D3d structure, geometry optimization.
X
BE 1 R1
B  2 R*   1  A
B  2 R*   1  A 3 T
X  2 RX*  1  A 3 T
X  2 RX*  1  A 4 T
H  5 RHX* 2  A 1 T
H  5 RHX* 2  A 1 T60
```

```
H  5 RHX* 2  A 1 TM6
H  6 RHX* 2  A 1 T12
H  6 RHX* 2  A 1 TM2
H  6 RHX* 2  A 1 T0
X  3 R1   2  A 1 T0
X  4 R1   2  A 1 T0
H  3 RHT* 13 A 2 T
H  4 RHT* 14 A 2 T

R1=1.207
R=1.05183
RX=0.08546
RHX=1.48313
RHT=0.569
A=90.
T=180.
T60=60.
T12=120.
T0=0.
TM2=-120.
TM6=-60.

*ACES2
BASIS=DZP,CALC=1
OPT_METHOD=MANR,EVAL_HESS=3,MAX_STEP=750,CONV=5
```

This ZMAT file specifies a geometry optimization for the D_{3d} isomer of beryllium borohydride, BeB_2H_8. Compared with the water example above, which uses defaults for the optimization keywords, this optimization input adds some features. The MANR optimization algorithm is used. The (SCF) Hessian will be reevaluated every three cycles, and the maximum step length is set to 750 millibohr. The convergence criterion (CONV) is set to 5, which means the optimization will continue until the root-mean-square force is below 10^{-5}, rather than the default of 10^{-4} atomic units. An MBPT(2) calculation is specified by the CALC keyword. This has been specified using the number corresponding to MBPT(2), although our recommendation is to use the name. The default values for CHARGE, MULT, REF, and OCCUPATION are used. The geometry optimization request automatically turns on the necessary gradient options.

9.2.3 Full optimization of Cartesian coordinates

```
RIC H2 geom opt
H 0. 0. 0.
H 0. 0. 0.5

*ACES2(basis=DZP,geom_opt=full)
```

This feature is controlled by the GEOM_OPT keyword. The RIC analysis includes information about atomic radii. If the input geometry is far from equilibrium, then the bond analysis may not be accurate and the calculation will become unstable (but it might not crash).

9.2.4 Transition state search

```
CH2O --> H2 + CO transition state search.  DZP basis.
O
C 1 R*
H 2 R2*  1 A*
H 3 RHH* 2 A* 1 T

R=1.254
R2=1.08
RHH=1.01
A=133.5
T=0.

*ACES2(OPT_METHOD=EVFTS,MAX_STEP=150,BASIS=DZP
       CALCLEVEL=MBPT(2),OCCUPATION=7-1)
```

As implied by the job title, this ZMAT file specifies an initial geometry and basis set for a transition state search. In the Z matrix, the parameters designated R, R2, RHH, and A will be optimized, while parameter T will be fixed at 0 degrees. This obviously constrains the structure to be planar. The *ACES2 namelist parameters tell ACES II that

1. this is a transition state search, using the Cerjan-Miller eigenvector following method;
2. the maximum allowed step length is 150 millibohr (this corresponds to the absolute step length (1-norm) since the default value of SCALE_ON has not been overridden);
3. the DZP basis set is to be used;
4. the calculation type is MBPT(2); and
5. the occupation (appropriate for the C_s point group) is 7-1.

The value "EVFTS" automatically turns on the necessary gradient options. In transition searches, it is necessary to provide an initial estimate of the hessian. This is usually done by saving the FCMINT file from a frequency calculation and copying this to the workspace at the beginning of a transition state search. Another example details this procedure. Like all optimizations, transition states searches use analytical gradients by default. If these are not available, then the user must use GRAD_CALC=NUMERICAL.

9.2.5 Restarting an optimization (or frequency) calculation

```
% savedir = ./SAVEDIR
restartable H2 geom opt
H
H 1 R*

R=1.0

*ACES2(basis=DZP,restart)
```

This example reinforces the default restart behavior. If restarting the calculation is not necessary, then use RESTART=OFF or !RESTART in the *ACES2 namelist. All of the restart capability is built into xjoda and is controlled by two values: the SAVEDIR directive and the RESTART flag. The SAVEDIR directive is case insensitive, and the path may be absolute or relative. Each time xjoda is called, it copies the optimization history files to SAVEDIR. If xjoda is called and the files are contaminated, meaning the previous job was interrupted (or the dirty flag was not cleared), then it will restore the history files from SAVEDIR and continue on as if nothing had happened. This applies to both geometry optimizations and VIB=FINDIF frequency calculations.

WARNING: Restarts are only implemented for coarse-grain checkpointing. If other ACES II files are in the directory, interesting things might happen. For the best results, calculations should be restarted from a clean directory (i.e., only the original user input files should be present).

When the optimization is finished, xjoda will delete the CURRENT and OLD directories within SAVEDIR but will not delete that top-level directory. (This behavior might change in that if xjoda created SAVEDIR, then it will delete it as well.)

9.2.6 Initializing the Hessian with FCMINT in a geometry search

All of the geometry optimization algorithms incorporated into ACES II are based on the Newton-Raphson method, in which step directions and sizes are related to the first and second derivatives of the molecular potential energy. However, in almost all practical calculations the exact second derivative matrix is not evaluated but rather approximated. As the calculation progresses, well-established numerical methods are used to estimate the elements of the Hessian matrix based on all previous optimization steps. After a large number of steps have been taken, one may safely assume that the totally symmetric Hessian used to form the step is a reasonable approximation to the correct Hessian. However, in the early stages of the optimization there is not a sufficient amount of available information to accurately estimate the Hessian and problems may ensue. By default, ACES II geometry optimizations begin with a very crude estimate of the Hessian in which all force constants for bonded interactions (as specified by the Z matrix connectivity) are set to 1 hartree/bohr^2, all bending force constants (corresponding to bond angles in the Z matrix) are set to 0.25 hartree/bohr^2, and all torsional force constants are set to 0.10 hartree/bohr^2. While this initialization and the subsequent numerical updates work satisfactorily for most small molecules, there can be occasional problems. In these cases, one might wish to use an alternative initial force constant matrix, particularly one obtained by ACES II at the same or another level of theory. There are a number of ways in which one might do this. First, the EVAL_HESS keyword can be used. If nonzero, the value associated with this keyword directs ACES II to calculate the

SCF Hessian matrix prior to the first optimization step and then every N steps thereafter, where N is the value of EVAL_HESS. By setting N to a sufficiently large value (larger than OPT_MAXCYC), then the Hessian will never be recalculated and the optimization will begin with the SCF Hessian.

However, the strategy based on EVAL_HESS is not sufficient for all purposes. For example, one might wish to use a Hessian which is evaluated at the correlated level. This is not possible with EVAL_HESS, since it will direct ACES II to calculate only the SCF Hessian, regardless of the calculation type. Alternatively, one might adopt the economical strategy of using a Hessian which is evaluated at a low level of theory, such as SCF with the STO-3G basis set. This is the recommended approach for transition state searches and all optimizations for which the default Hessian is inadequate. In any event, it is quite straightforward to use another set of force constants to begin an optimization. First, one simply runs a harmonic frequency calculation at the desired level of theory and saves the file named `FCMINT`. This formatted file contains the full internal coordinate force constant matrix. When `xjoda` performs a geometry optimization, it checks the active working area for the presence of `FCMINT`—no special keywords or commands are needed to do this. If the file exists, `xjoda` uses those force constants to initialize the Hessian matrix.

While the geometry (and even the point group symmetry) specified by `ZMAT` in the harmonic frequency calculation need not be the same as that used in the first step of the geometry optimization, *the Z matrix connectivity must be identical.* If one attempts to use an entirely different Z matrix, then the definitions of internal coordinates are no longer the same and chaos may ensue. While this point may seem to be unimportant, this situation occurs relatively frequently. Suppose the equilibrium geometry for a transition state is assumed to be planar, and the user attempts to locate a stationary point. However, when the harmonic frequencies are calculated, two modes are found to have imaginary frequencies: an a' mode (in-plane) and an a'' mode, with the a' mode corresponding (approximately) to the reaction coordinate of interest. As a result, the true transition state geometry does not contain a plane of symmetry and another search must be performed in a reduced symmetry. To this end, it would certainly be useful to use `FCMINT` obtained in the frequency calculation to start the search, but one must be careful that the molecular configuration used in the reduced symmetry has *exactly* the same connectivity scheme as that used in the search for the planar structure.

9.3 Frequency calculations

9.3.1 Numerical frequencies from analytical gradients

```
N4 finite difference frequency calculation
N
```

```
X 1 R
N 2 R 1 TDA
N 2 R 1 TDA 3 T
N 2 R 1 TDA 4 T

R=0.945
TDA=110.
T=120.

*ACES2(VIB=FINDIF,BASIS=TZ2P,CALC=MBPT(2))
```

This ZMAT file directs ACES II to perform a harmonic frequency calculation for N_4 and to compute the force constants by numerical differentiation of analytic gradients. The TZ2P basis set is selected. Note that the TDA parameter is used in the Z matrix. Although the specified value in the parameter input section is not the exact tetrahedral angle, the program will convert TDA to the correct value (109.4712206... degrees) internally.

Since finite differences are used, no symmetry-specific keywords can be specified, so the orbital symmetry specification is omitted. This is because the determination of the force constants requires several gradient calculations at geometries with different symmetries.

The FINDIF option automatically turns on the requisite gradient and property options. Note that no asterisks may appear in the Z matrix in a frequency calculation. This is a common error since frequency calculations are usually preceded by geometry optimizations, which require the asterisks.

9.3.2 Numerical frequencies from energies

```
CCSD(TQf) NUMERICAL VIBRATION CALCULATION FOR N2
N
N 1 R

R=1.116

*ACES2(CALC=CCSD(TQf),BASIS=DZP,VIB=FINDIF,GRAD_CALC=NUMERICAL)
```

This example specifies a finite difference frequency calculation for N_2 using energy points from the CCSD(TQf) method. This method of calculating frequencies is applicable to all types of energy calculations, not just CCSD(TQf).

9.3.3 Isotopic shift

ACES II provides a straightforward way to calculate changes in harmonic vibrational frequencies and infrared intensities due to isotopic substitutions. This is accomplished with a free-format file called ISOMASS, which contains the desired atomic masses in their ZMAT order (excluding dummy atoms). For example, if a user wants to calculate the ^{16}O–^{18}O isotopic shift for the vibrational frequencies of water, then the following ZMAT and ISOMASS files may be used:

ZMAT:

```
Water frequency calculation
X
O 1 R
H 2 R1 1 A
H 2 R1 1 A 3 T

R=1.0
R1=0.95
A=130.
T=180.

*ACES2(CALC=SCF,BASIS=DZP,VIB=EXACT)
```

ISOMASS:

```
18.0
1.00797
1.00797
```

10 Parallelization

10.1 Overview

ACES II can perform the following calculations in parallel:

1. HF-SCF and MBPT(2) single-point energies,
2. HF-SCF analytical geometry optimizations, and
3. all finite-displacement methods (numerical geometry optimizations and vibrational frequency calculations).

The SCF and MBPT(2) energy programs, `xp_vscf` and `xp_dirmp2`, use direct integrals and communicate over the Message Passing Interface (MPI). The SCF gradients in `xp_scfgrd` use direct integral derivatives in the same fashion. The parallel finite displacement algorithm is handled by the main driver program `xp_aces2` and distributes displacements to each MPI task. All of these programs require each MPI task to have its own set of files, which can be managed before and after the parallel run by `xgemini`.

ACES III, the fully parallel successor to ACES II, will be able to compute HF-SCF, MBPT(2), and CCSD energies and gradients using a message passing protocol (either MPI or shmem). Despite the new internal architecture, it will be compatible with the current input files `ZMAT` and `GENBAS`, but it will not require separate file sets for each task.

The following scripts illustrate each parallel capability. The sections that follow describe in further detail how some of these capabilities work.

```
#!/bin/sh
# Script 1: parallel SCF and MBPT(2) energies
mpirun -np N xgemini -i -s -t shared.@GRANK@ "xjoda && xvmol && xvmol2ja"
mpirun -np N xp_vscf
mpirun -np N xp_dirmp2 # OMIT THIS LINE FOR SCF ENERGIES ONLY
mpirun -np N xgemini -x -s

#!/bin/sh
# Script 2: parallel finite differences
mpirun -np N xgemini -i -s -t shared.@GRANK@
mpirun -np N xp_aces2
mpirun -np N xgemini -x -s


#!/bin/sh
# Script 3: parallel SCF geometry optimizations
mpirun -np N xgemini -i -s -t shared.@GRANK@
PRUN="mpirun -np N" parscfopt.sh
mpirun -np N xgemini -x -s
```

10.2 Running `xgemini`

`xgemini` is a tool for managing private scratch directories from a central location. Each MPI task (of `xgemini` or the parallel AMEs) must start in this directory, which means it must be globally visible on distributed computers. Henceforth, this directory will be called WORKDIR. Depending on the machine architecture, the two choices the user must make are: (1) where will the private scratch directories be created and (2) how will the AMEs get to them. These decisions will affect performance if local disks are available on each node and will determine how a parallel job should be restarted.

When the user runs a parallel member executable (usually `xp_aces2`), each parallel task must start in WORKDIR, which usually contains `ZMAT` and `GENBAS`. Before reading any files, each task attempts to cd into a directory called *nodename.rank*. The tasks on a machine named crunch would look for "crunch.0", "crunch.1", etc. If they cannot find these directories, then they try to cd into directories called `shared`.*rank* (shared.0, shared.1, etc.). If those are not found either then the tasks start running in the current directory.

10.2.1 Local scratch directories

For the following example, assume that the parallel computer has compute nodes called compnode0, compnode1, etc. and that the user (named smith) is allowed to create directories in `/local/tmp`. The following command will create a directory called `smithdir` on every node:

```
xgemini -i -t /local/tmp/smithdir
```

Assuming all compute nodes took part in the parallel run, then it is likely that the user would see the following WORKDIR directory listing:

```
ZMAT    GENBAS    compnode0.0    compnode1.1    compnode2.2    ...
```

Each compnodeX.X is a symbolic link that points to `/local/tmp/smithdir`, and inside each of those local directories are symbolic links back to `ZMAT` and `GENBAS`. The `-i` flag is what instructs `xgemini` to create the directories and links.

It is usually the case that if local directory partitions are available then performing I/O on them will yield better performance than reading and writing to a partition over the network. The downside is that if a parallel job is stopped and needs to be restarted, then the user *must* tell the parallel job scheduler (like LoadLeveler or PBS) to run the new job on the same nodes that the previous job ran on (simply because those are the nodes with the data).

10.2.2 Shared scratch directories

If a global file system (like GPFS or even NFS) is used for the scratch directories, then a task on any node can get to any other scratch directory. Assume user smith can create directories in `/global/tmp`. Reusing the previous `xgemini` command line with "local" changed to "global" will not work because every parallel task will attempt to create `/global/tmp/smithdir`. One task will succeed but all of the others will fail (and crash). GEMINI has a rich set of pattern macros for this kind of customization, but the simplest command to use for the job would be:

```
xgemini -i -s -t /global/tmp/smithdir.@GRANK@
```

Each task will replace the string "@GRANK@" will its global rank (an integer from 0 to N-1, where N is the number of parallel tasks).

Creating scratch directories in `/global` allows all nodes in the computer to see all scratch directories, but the symbolic links in WORKDIR will still be named compnodeX.Y unless the `-s` flag is used. If a parallel job is restarted, process 0 might be running on compnode2, and it would expect to see compnode2.0 in WORKDIR. However, if the previous run had process 0 running on compnode0, then it would have created compnode0.0, and the restarted job will crash. With `-s`, the symbolic links in WORKDIR are named `shared`.*rank*, so it does not matter which actual node created the scratch directory or the link to it. If shared links are used, then *every call to* ***xgemini*** *must use the* ***-s*** *flag.* The downside to remote scratch directories is that performance might decrease with high network traffic, although this is highly dependent on the hardware architecture and runtime load.

Some users might feel more comfortable limiting all activity to WORKDIR. In this case, the scratch directory pattern can be the same as the symbolic link. For example,

```
xgemini -i -s -t shared.@GRANK@
```

will create directories instead of symbolic links in WORKDIR. The parallel AMEs will not know the difference when they attempt to cd into the scratch directory.

10.2.3 Command-line flags and pattern macros

As previously demonstrated, `xgemini` can create the runtime environment (with `-i`), can use "shared" node names (with `-s`), and can create private scratch directories with almost any name the user requires (with `-t` and macro substitution). In addition, it can use arbitrary input and basis files (with `-z` and `-b`), can run serial commands in each directory (all arguments that are not flags), and can clean up after a parallel run (with `-x`). The command-line structure is as follows:

```
xgemini [-h] [-s] [-i [-z file] [-b file] [-t pattern]] [-x] [--] [exec]
```

Flag	Description
`-h`	print usage text
`-s`	use shared scratch directories (useful for restarts)
`-i`	initialize (make scratch directories and symlinks)
`-z` *file*	link *file* to `ZMAT` (default is "ZMAT")
`-b` *file*	link *file* to `GENBAS` (default is "GENBAS")
`-t` *pattern*	*pattern* defines scratch directory paths
`-x`	clean (remove scratch directories and symlinks)
`--`	terminate `xgemini` flag parsing
exec	pass *exec* arguments to a system command in each scratch directory

The pattern that defines the scratch directories can be an absolute or relative path with respect to WORKDIR. Another example of directory patterns is

```
/usr/var/tmp/@LOGNAME@/scr.@GRANK@
```

which would become

```
/usr/var/tmp/smith/scr.0
```

for the root task of user smith. The following table shows the complete list of macros and their substituted values.

Macro	Substitution
`@NODENAME@`	the string at `uname.nodename` as described by `sys/utsname.h` (most likely "`uname -n`")
`@LOGNAME@`	the string returned by `getlogin_r()` or `cuserid()` (most likely "`whoami`")
`@SID@`	the session ID of the `xgemini` process
`@PID@`	the ID of the `xgemini` process
`@PPID@`	the ID of the `xgemini` parent process
`@GRANK@`	the MPI process rank in `MPI_COMM_WORLD`
`@GPROCS@`	the number of MPI processes in `MPI_COMM_WORLD`
`@HRANK@`	the MPI process rank in `MPI_COMM_HOST`
`@HPROCS@`	the number of MPI processes in `MPI_COMM_HOST`

10.3 Examples

10.3.1 Parallel finite differences with MPI (automatic)

The two programs that are of primary concern in this exercise are `xgemini` and `xp_aces2`. Assume `xgemini` creates shared scratch directories:

```
WORKDIR> ls
ZMAT    GENBAS
WORKDIR> xgemini -i -s -t /local/tmp/smith.@GRANK@
...
WORKDIR> ls -F
ZMAT    GENBAS    shared.0@    shared.1@    ...
```

Before running `xp_aces2`, the user should determine if output tagging is available for parallel processes. Without this, the standard output stream of every process will be merged and it will be almost impossible to discern which task did what. Furthermore, since the operating system buffers I/O streams, the final lines of output might not contain the "final answer." IBM's parallel environment can tag each line with the global rank if "`-labelio yes`" is used (or if MP_LABELIO=yes is set in the environment). (HP/Compaq computers also have a tagging feature.)

```
WORKDIR> xp_aces2 -labelio yes > out
...
WORKDIR> grep '^ *0:'  out
0:  [output from task 0]
...
```

And finally,

```
WORKDIR> xgemini -x -s
...
WORKDIR> ls
ZMAT    GENBAS    out
```

The user should not request more process elements than there are finite displacements. If this happens, then some instances of `xjoda` will see there are no displacements and bomb. The surest way of checking this before submitting the job is by running `xjoda` on the input file. The FD logic will print out a table like the following: (data from CH_4 fully numerical vibrational frequencies in C_1)

```
@VIBINF-I, Symmetries species for nuclear motions:
Irrep     Label    Total    Vibrations     Translations     Rotations
  1           A    15.00       9.00            3.00            3.00
 Total number of calculations required      :    90
 Number of single-point energy calculations :    90
 Number of energy gradient     calculations :     0
```

For frequency calculations using numerical gradients, the reference geometry is added to the set of displacements, so the example above can be run with at most 91 process elements. CH_4 frequencies in C_1 with analytical gradients (or geometry optimization with numerical gradients) can use at most 18 process elements.

10.3.2 Parallel finite differences with scripts (manual)

Running a set of calculations in parallel by hand (i.e., without MPI) is certainly doable, and because the synchronization is handled through ASCII files, the calculations can even be performed on different architectures; however, it is not pretty to coordinate, and if the user is running an optimization with numerical gradients, then he or she should expect a *lot* of intervention before the coordinates converge. In addition to complicated shell scripting, the user must also know the exact order of ACES member executables that are required for each single point calculation.

The serial `xjoda` binary can accept two command-line flags, `-procs` and `-rank`, that instruct it to set up single-point calculations on a subset of the displacements. Once the last displacement is complete (but before the final `xjoda`), the user must update each local file set with the data from all of the other file sets. For vibrational frequencies, this can be done on a single file set, but for geometry optimizations, each file set must be updated before taking the next step.

The two programs that are of primary concern are `xjoda` and `xa2proc`. Every time `xjoda` is called, the number of processes and the rank *must* be supplied on the command line. `xa2proc` contains a module that will update, print, and load the data needed from each file set.

The following Korn shell script will run through every AME for every virtual process and collate the results of a vibrational frequency calculation. *This example does the same thing as the serial* ***xaces2*** *program on one CPU and is shown for demonstration purposes only.*

```
#!/bin/ksh
echo Do not run this script as is && exit 1

# pick a function
alias loop=lastraman # for analytical gradients w/ raman intensities
alias loop=lastgrad  # for analytical gradients (1st order grid)
alias loop=lastener  # for numerical  gradients (2nd order grid)

test -n "$1" && procs=$1 || procs=1 test $procs -lt 1 && exit 1
```

```
alias xj='xjoda -rank $rank -procs $procs'
if true # Hartree-Fock
   alias scf='xvscf'
   alias dint='xvdint'
else # DFT
   alias scf='xvscf_ks && xintgrt'
   alias dint='xvdint && xvksdint'
fi

function lastener {
   lastgeom=0
   while test $lastgeom -eq 0
   do xa2proc rmfiles
      (xj && xvmol && xvmol2ja && scf) || return 1
      (xvtran && xintprc && xvcc) || return 1
      lastgeom=$(xa2proc jareq i LASTGEOM 1 | tail -1)
   done
}

function lastgrad {
   lastgeom=0
   while test $lastgeom -eq 0
   do xa2proc rmfiles
      (xj && xvmol && xvmol2ja && scf) || return 1
      (xvtran && xintprc && xvcc && xlambda) || return 1
      (xdens && xanti && xbcktrn) || return 1
      (dint) || return 1
      lastgeom=$(xa2proc jareq i LASTGEOM 1 | tail -1)
   done
}

function lastraman {
   lastgeom=0
   while test $lastgeom -eq 0
   do xa2proc rmfiles
      (xj && xvmol && xvmol2ja && xvprops && scf) || return 1
      (xvtran && xintprc && xvcc && xlambda) || return 1
      (xdens && xanti && xbcktrn) || return 1
      (dint && xcphf) || return 1
      lastgeom=$(xa2proc jareq i LASTGEOM 1 | tail -1)
   done
}

# clear out old FD data
rm -rf shared.* fd.out

rank=$((procs-1))
while test $rank -ge 0; do
 # this is the "xgemini" portion
   mkdir shared.$rank
   cd shared.$rank
   ln -s ../ZMAT
   ln -s ../GENBAS
 # this is the meat of the routine
   loop > ../$rank.out || exit 1
   xa2proc parfd updump >> ../fd.out # update and print the FD data
 # reset and cycle to the next process
```

```
   cd ..
   let rank-=1
done
cd shared.0
xa2proc parfd load ../fd.out # load the FD data from the other procs
xjoda -procs $procs # run the final xjoda
cd ..
```

For geometry optimizations, every process will have to load the FD data, and the entire procedure will have to loop over coordinates until convergence—when the integer record JODADONE is 1. Needless to say, this is not an exercise for even intermediate users of ACES II, but the capability exists should the need arise.

10.3.3 SCF geometry optimizations

In Script 3 from the Overview section (page 92), the core of the commands are contained in another shell script `parscfopt.sh`. That script contains the loop logic to iterate until `xjoda` has found the minimum energy geometry.

```
#!/bin/sh
# parscfopt.sh
test -d shared.0 && s="-s" || s=" "
rootdir=‘ls -d *.0‘
$PRUN xgemini $s xjoda
cd $rootdir; jodadone=‘xa2proc jareq i JODADONE 1 | tail -1‘; cd -
while test $jodadone -eq 0
do
   $PRUN xgemini $s "xvmol && xvmol2ja"
   $PRUN xp_vscf
   $PRUN xp_scfgrd
   $PRUN xgemini $s xjoda
   cd $rootdir; jodadone=‘xa2proc jareq i JODADONE 1 | tail -1‘; cd -
done
```

11 Troubleshooting

11.1 Common mistakes

The following tips should help users to detect and avoid errors. For large calculations, it is *strongly* recommended that users run with trial input files (small basis set, high convergence tolerances, etc.) before running the actual system.

- The `*ACES2` namelist is not terminated properly. If it begins with an open parenthesis, then it must end with a close parenthesis; otherwise, it must end with a blank line. Even if the namelist is terminated with a close parenthesis, there must be a blank line after the namelist and the end-of-file mark.

- The BASIS=SPECIAL option is used but the order of the basis sets does not correspond to the order of atoms in the Z matrix. *The code does not check this and will not crash!* For example, one is allowed to put a C basis set on a CL atom. In fact, ghost atoms would not be possible without this feature.

- All atomic symbols should be in upper case. For example, "Cl" has to be entered as "CL". Actually, this is not true anymore, but older versions of the code would simply give incorrect results. If any user finds case-sensitivity in the coordinate matrix parsing, `aces2@qtp.ufl.edu` should be notified with the failing `ZMAT`.

- Lower case characters have been used instead of upper case. This situation is difficult to pinpoint. Most of the `*ACES2` namelist parsing is case insensitive, along with the atomic symbols. File directives in the header are still case sensitive as are basis set names.

- The OCCUPATION keyword takes precedent over CHARGE and MULTIPLICITY. This can lead to confusion in open-shell SCF calculations. Here are some usage tips:

    1. If OCCUPATION has been specified, then the CHARGE and MULTIPLICITY keywords are ignored. It is, however, good practice to make these consistent with OCCUPATION
    2. If the REFERENCE keyword is absent from an open-shell calculation or is erroneously set to RHF, then unpredictable things might happen. ACES II does not have a default type of open-shell SCF.
    3. The occupation specified in the GUESS file takes precedent over all keywords. Again, it is sensible to make them the same to avoid confusion.

4. In older binaries, specifying only the α occupation with OCCUPATION dropped all β electrons. This is trapped in the current version.

- An input file contains text beyond the 80th column.
- There is no title line, and the first line of the Z matrix has been entered as the title.
- There were files from a previous ACES II calculation in the workspace. In general, one should clear the workspace of all previous ACES II files prior to copying in the new files.

11.2 Basic program restrictions

- `ZMAT` should not change during a running calculation. There might be some hacks that involve changing convergence tolerances mid-stream, but these are not documented, supported, or advised.

11.3 Suggestions for reducing resources

- ABCDTYPE=AOBASIS
  This saves disk space for CC methods.
- SINGLE_STORE=ON
  This saves disk space for correlated methods.
- DIRECT,INTEGRALS=GAMESS,FOCK=AO,SYMMETRY=OFF
  This save disk space for all calculations, but only applies to SCF and MBPT(2) theories (without DROPMO).

12 References

Of the many methods currently implemented in ACES II, some are well established, while others are new and descriptions have not yet been published. It is the purpose of this section to list pertinent literature references which provide more information about the techniques and their implementations and should be cited when results from ACES II calculations are published. Some references to basis sets included in the program are also included.

Guide to Correlated Methods

- R.J. Bartlett and J.F. Stanton, "Applications of Post-Hartree-Fock Methods: A Tutorial", in *Reviews of Computational Chemistry* **5**, 65-169, edited by K.B. Lipkowitz and D.B. Boyd (VCH Publishers, New York, 1994).

- R.J. Bartlett, "Coupled-Cluster Theory: An Overview of Recent Developments", in *Modern Electronic Structure Theory, Part I*, edited by D.R. Yarkony (World Scientific Publishing Co., Singapore, 1995).

12.1 Many-body perturbation theory (MBPT)

Review article

- R.J. Bartlett, Ann. Rev. Phys. Chem. **32**, 359 (1981).

a) Closed-shell RHF-MBPT for molecules

- R.J. Bartlett and D.M. Silver, Phys. Rev. **A10**, 1927 (1974); J. Chem. Phys. **62**, 3258 (1975); ibid. **64**, 4578 (1976).

- R.J. Bartlett and I. Shavitt, Chem. Phys. Lett. **50**, 190 (1977).

b) Open-shell UHF-MBPT for molecules

- R.J. Bartlett and G.D. Purvis III, Int. J. Quantum Chem. **14**, 561 (1978).

c) Open-shell ROHF-MBPT for molecules

- W.J. Lauderdale, J.F. Stanton, J. Gauss, J.D. Watts, and R.J. Bartlett, Chem. Phys. Lett. **187**, 21 (1991); J. Chem. Phys. **97**, 6606 (1992).

12.2 Coupled-cluster (CC) theory

Review article

- R.J. Bartlett, J. Phys. Chem. **93**, 1697 (1989).

a) CCD method

- R.J. Bartlett and G.D. Purvis III, Int. J. Quantum Chem. **14**, 561 (1978).

b) CCSD method

- G.D. Purvis III and R.J. Bartlett, J. Chem. Phys. **76**, 1910 (1982).

c) CCSDT-1 method

- Y.S. Lee, S.A. Kucharski, and R.J. Bartlett, J. Chem. Phys. **81**, 5906 (1984).

d) CCSD+T(CCSD) method

- M. Urban, J. Noga, S.J. Cole, and R.J. Bartlett, J. Chem. Phys. **83**, 4041 (1985).

e) CCSD(T) method (adds one (HF case) or two (non-HF case) small terms to CCSD+T(CCSD))

- K. Raghavachari, G.W. Trucks, J.A. Pople, and M. Head-Gordon, *Chem. Phys. Lett.* **157**, 479 (1989).
- R.J.Bartlett, J.D.Watts, S.A.Kucharski, and J.Noga, Chem. Phys. Lett. **165**, 513 (1990).
- J. Gauss, W.J. Lauderdale, J.F. Stanton, J.D. Watts, and R.J. Bartlett, *Chem. Phys. Lett.* **182**, 207 (1991).
- J.D. Watts, J. Gauss, and R.J. Bartlett, *J. Chem. Phys.* **98**, 8718 (1993).

f) CCSDT-2 and CCSDT-3 methods

- J.Noga, R.J.Bartlett, and M.Urban, Chem. Phys. Lett. **134**, 146 (1987).

g) CCSDT methods

- J. Noga and R.J. Bartlett, J. Chem. Phys. **86**, 7041 (1987).
- G.E. Scuseria and H.F. Schaefer, Chem. Phys. Lett. **152**, 382 (1988).

- J.D. Watts and R.J. Bartlett, J. Chem. Phys. **93**, 6104 (1990).
- J.D. Watts and R.J. Bartlett, Int. J. Quant. Chem. Symp. **27**, 51 (1993).

h) ROHF and QRHF CC methods

- M.Rittby and R.J.Bartlett, J. Phys. Chem. **92**, 3033 (1988).

i) Brueckner CC methods

- R.A. Chiles and C.E. Dykstra, J. Chem. Phys. **74**, 4544 (1981).
- J.F.Stanton, J.Gauss, and R.J.Bartlett, J. Chem. Phys. **97**, 5554 (1992).

j) QCISD and QCISD(T) methods

- J.A. Pople, M. Head-Gordon, and K. Raghavachari J. Chem. Phys. **87**, 5968 (1987).

k) UCC methods

- J.D. Watts, G.W. Trucks, and R.J. Bartlett, Chem. Phys. Lett. **157**, 359 (1989).

l) CCSD+TQ*(CCSD), CCSD(TQ), and QCISD(TQ) methods

- R.J. Bartlett, J.D. Watts, S.A. Kucharski, and J. Noga, Chem. Phys. Lett. **165**, 513 (1990).
- K. Raghavachari, J.A. Pople, E.S. Replogle, and M. Head-Gordon, J. Phys. Chem. **94**, 5579 (1990).

m) Two-determinant CCSD method

- A. Balkova and R.J. Bartlett, Chem. Phys. Lett. **193**, 364 (1992).

12.3 Analytical gradients for MBPT/CC methods

Review article

- R.J.Bartlett, J.F.Stanton, and J.D.Watts, in *Advances in Molecular Vibrations and Collision Dynamics*, Vol. 1, ed. J.Bowman, JAI Press, p. 139 (1991).
- J. Gauss and D. Cremer, Adv. Quant. Chem. **23**, 205 (1992).

a) General MBPT/CC gradient theory

- E.A.Salter, G.W.Trucks, and R.J.Bartlett, J. Chem. Phys. **90**, 1752 (1989).

b) Implementation for closed- and open-shell CCSD

- J.Gauss, J.F.Stanton, and R.J.Bartlett, J. Chem. Phys. **95**, 2623 (1991).

c) QRHF-CCSD gradients

- J.Gauss, J.F.Stanton, and R.J.Bartlett, J. Chem. Phys. **95**, 2639 (1991).

d) ROHF-CCSD gradients

- J.Gauss, W.J.Lauderdale, J.F.Stanton, J.D.Watts, and R.J.Bartlett, Chem. Phys. Lett. **182**, 207 (1991).

e) QCISD gradients

- J. Gauss and D. Cremer, Chem. Phys. Lett. **150**, 280 (1988).
- J.Gauss, J.F.Stanton, and R.J.Bartlett, J. Chem. Phys. **95**, 2623 (1991).

f) MBPT(4) gradients for closed- and open-shells

- J. Gauss and D. Cremer, Chem. Phys. Lett. **138**, 131 (1987); **153**, 303 (1988).
- G.W.Trucks, J.D.Watts, E.A.Salter, and R.J.Bartlett, Chem. Phys. Lett. **153**, 490 (1988).
- J.D.Watts, G.W.Trucks, and R.J.Bartlett, Chem. Phys. Lett. **164**, 502 (1989).
- J.D.Watts, J. Gauss, and R.J.Bartlett, *Chem. Phys. Lett.* **200**, 1 (1992).

g) CCSD+T(CCSD), CCSD(T) gradients for closed- and open-shells

- J.D.Watts, J. Gauss, and R.J.Bartlett, *Chem. Phys. Lett.* **200**, 1 (1992).
- J.D. Watts, J. Gauss, and R.J. Bartlett, *J. Chem. Phys.* **98**, 8718 (1993).

h) QCISD(T) gradients

- J. Gauss and D. Cremer, Chem. Phys. Lett. **163**, 549 (1990).
- J.D.Watts, J. Gauss, and R.J.Bartlett, *Chem. Phys. Lett.* **200**, 1 (1992).

i) UCC(4) gradients for closed- and open-shells

- J.D.Watts, G.W.Trucks, and R.J.Bartlett, Chem. Phys. Lett. **157**, 359 (1989).
- J.D.Watts, G.W.Trucks, and R.J.Bartlett, Chem. Phys. Lett. **164**, 502 (1989).

j) Two-determinant CCSD (TD-CCSD) analytical gradients for open-shell singlet states

- P.G. Szalay and R.J. Bartlett, *J. Chem. Phys.* **101**, 4936 (1994).

12.4 Analytical second derivatives for MBPT/CC methods

a) MBPT(2) second derivatives for closed shells

- N.C. Handy, R.D. Amos, J.F. Gaw, J.E. Rice, E.D. Simandiras, T.J. Lee, R.J. Harrison, W.D. Laidig, G.B. Fitzgerald and R.J. Bartlett, in *Geometrical Derivatives of Energy Surfaces and Molecular Properties*, edited by P. Jørgensen and J. Simons (Reidel, Dordrecht, 1986).
- N.C. Handy, R.D. Amos, J.F. Gaw, J.E. Rice, E.D. Simandiras, Chem. Phys. Letters **120**, 151 (1985).
- R.J. Harrison, G.B. Fitzgerald, W.D. Laidig, R.J. Bartlett, Chem. Phys. Lett. **124**, 291 (1986).

b) MBPT(2) second derivatives for open shells (UHF and ROHF)

- J.F. Stanton, J. Gauss, and R.J. Bartlett, Chem. Phys. Lett. **195**, 194 (1992).
- J. Gauss, J.F. Stanton, and R.J. Bartlett, J. Chem. Phys. **97**, 7825 (1992).

12.5 NMR chemical shift calculations

a) GIAO-SCF NMR chemical shift calculations

- R. Ditchfield, Mol. Phys. **27**, 789 (1974).
- K. Wolinski, J.F. Hinton, and P. Pulay, J. Am. Chem. Soc. **112**, 8251 (1990).
- M. Häser, R. Ahlrichs, H.P. Baron, P. Weis, and H. Horn, Theoret. Chim. Acta, **83**, 455 (1992).

b) GIAO-MBPT(2) NMR chemical shift calculations

- J. Gauss, Chem. Phys. Lett. **191**, 614 (1992); J. Chem. Phys. **99**, 3629 (1993).

12.6 Methods for calculating excitation energies

a) Tamm-Dancoff (CI singles) approximation
b) Random-Phase approximation (RPA)
c) Equation-of-Motion Coupled Cluster (EOM-CC) methods

- J.F. Stanton and R.J. Bartlett, J. Chem. Phys. **98**, 7029 (1993)

12.7 Methods for calculating electron attachment energies

The electron affinity equation-of-motion coupled-cluster method :

- M. Nooijen and R.J. Bartlett, *J. Chem. Phys.* **102**, 3629 (1995).

12.8 Time-dependent Hartree-Fock methods

- H. Sekino and R.J. Bartlett, J. Chem. Phys. **85**, 976 (1986).
- H. Sekino and R.J. Bartlett, Int. J. Quantum Chem. **43**, 119 (1992).

12.9 HF-DFT method

- P. M. W. Gill, B. G. Johnson, and J. A. Pople, Int. J. Quantum Chem. Symp. **26**, 319 (1992).
- G. E. Scuseria, J. Chem. Phys. **97**, 7528 (1992).
- N. Oliphant and R. J. Bartlett, J. Chem. Phys. **100**, 6550 (1994).

12.10 Basis sets

STO-3G

- W.J.Hehre, R.F.Stewart, and J.A.Pople, J. Chem. Phys. **51**, 2657 (1969); first-row elements.
- W.J.Hehre, R.Ditchfield, R.F.Stewart, and J.A.Pople, J. Chem. Phys. **52**, 2769 (1970); second-row elements and improved scale factors for Li and Be.
- W.J.Pietro, B.A.Levi, W.J.Hehre, and R.F.Stewart, Inorg. Chem. **19**, 2225 (1980); third-row elements.
- W.J.Pietro, E.S.Blurock, R.F.Hout,Jr, W.J.Hehre, D.J.DeFrees, and R.F.Stewart, Inorg. Chem. **20**, 3650 (1981); fourth-row elements.
- W.J.Pietro and W.J.Hehre, J. Comput. Chem. **4**, 241 (1983); first- and second-row transition metal elements.

3-21G

- J.S.Binkley, J.A.Pople, and W.J.Hehre, J. Am. Chem. Soc. **102**, 939 (1980); first-row elements.

- M.S.Gordon, J.S.Binkley, J.A.Pople, W.J.Pietro, and W.J.Hehre, J. Am. Chem. Soc. **104**, 2797 (1982); second-row elements.

4-31G

- R.Ditchfield, W.J.Hehre, and J.A.Pople, J. Chem. Phys. **54**, 724 (1971); H,C,O,N, and F.
- W.J.Hehre and J.A.Pople, J. Chem. Phys. **56**, 4233 (1972); B.
- W.J.Hehre and W.A.Lathan, J. Chem. Phys. **56**, 5255 (1972); P,S, and Cl.

6-31G

- W.J.Hehre, R.Ditchfield, and J.A.Pople, J. Chem. Phys. **56**, 2257 (1972); C,N,O, and F.
- J.D.Dill and J.A.Pople, J. Chem. Phys. **62**, 2921 (1975); Li,Be,B.
- M.M.Francl, W.J.Pietro, W.J.Hehre, J.S.Binkley, M.S.Gordon, D.J.DeFrees, and J.A.Pople, J. Chem. Phys. **77**, 3654 (1982); second-row elements.

6-31G* (and 6-31G**)

- P.C.Hariharan and J.A.Pople, Theor. Chim. Acta **28**, 213 (1973); first-row elements.
- M.M.Francl, W.J.Pietro, W.J.Hehre, J.S.Binkley, M.S.Gordon, D.J.DeFrees, and J.A.Pople, J. Chem. Phys. **77**, 3654 (1982); second-row elements.

6-311G**

- R.Krishnan, J.S.Binkley, R.Seeger, and J.A.Pople, J. Chem. Phys. **72**, 650 (1980); first-row elements.

Dunning basis sets

- T.H.Dunning, Jr, J. Chem. Phys. **53**, 2823 (1970); double-zeta and triple-zeta valence contractions of $9s5p$ primitive set for H,B-F, including $4s2p$ and $4s3p$.
- T.H.Dunning, Jr, J. Chem. Phys. **55**, 716 (1971); triple-zeta valence and more flexible contractions of $10s6p$ primitive set for H,B-F, including $5s3p$ and $5s4p$. Also $4s$ contraction for Li, $5s$ contraction for Be.

- T.H.Dunning, Jr and P.J.Hay, in *Methods of Electronic Structure Theory*, edited by H.F.Schaefer III, Plenum, New York, 1977; $6s4p$ contractions of $11s7p$ for Al-Cl (Cl basis contains typographical errors, see Craven et al, Chem. Phys. Lett. **116**, 119 (1985)); VDZP basis set for first-row elements; diffuse and Rydberg functions; recommended polarization exponents from SCF calculations.

- T.H.Dunning, Jr, J. Chem. Phys. **90**, 1007 (1989) (PVDZ-PVQZ); R.A. Kendall, T.H. Dunning, Jr, and R.J. Harrison, J. Chem. Phys. **96**, 6796 (1992) (diffuse functions for PVDZ to PVQZ); D.E. Woon and T.H. Dunning, Jr, J. Chem. Phys. **98**, 1358 (1993) (PV5Z and diffuse functions for PV5Z); methodology of the "correlation consistent" sets, but does not include the actual sets : these must be obtained directly from Dunning via FTP.

Atomic natural orbital basis sets of Roos and coworkers (WMR)

- P.O. Widmark, P.A. Malmqvist, and B.O. Roos, *Theor. Chim. Acta* **77**, 291 (1990).

- P.O. Widmark, B.J. Persson, and B.O. Roos, *Theor. Chim. Acta* **79**, 419 (1991).

Polarized basis sets of Sadlej and coworkers

- A.J. Sadlej, *Collec. Czech. Chem. Commun.* **53**, 1995 (1988).

- A.J. Sadlej and M. Urban, *J. Mol. Struct. (THEOCHEM)* **234**, 147 (1991).

- A.J. Sadlej, *Theor. Chim. Acta* **79**, 123 (1991).

- A.J. Sadlej, *Theor. Chim. Acta* **81**, 45 (1992).

- A.J. Sadlej, *Theor. Chim. Acta* **81**, 339 (1992).

Basis sets optimized by Ahlrichs and coworkers

- A. Schäfer, H. Horn, and R. Ahlrichs J. Chem. Phys. **97**, 2571 (1992). These basis sets have been optimized for atoms at the SCF level using analytic gradient techniques and have been supplemented with a suitable choice of polarization functions. Input in the ACES II program is in this case always via BASIS=SPECIAL. The actual basis sets can be obtained via FTP from the authors.

Some other extended basis sets for second-row atoms (sp parts)

- A.D.McLean and G.S.Chandler, J. Chem. Phys. **72**, 5639 (1980); up to $6s5p$ sets for second-row atoms contracted from up to $12s9p$ primitive sets.

Polarization exponents from correlated calculations

- L.T.Redmon, G.D.Purvis III, and R.J.Bartlett, J. Am. Chem. Soc. **101**, 2856 (1979); recommended polarization exponents for use with DZ basis sets for H,B,C,N, and O.

12.11 Integral packages

The direct integrals are obtained by Rys quadrature (1) using an implementation extended to spdfg and L shells taken from GAMESS (2).

(1) J. Rys, M. Dupuis, H.F. King. *J. Comput. Chem.* **4** 154-157 (1983).

(2) M.W. Schmidt, K.K. Baldridge, 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).

A Other Keywords

A.1 Experimental, obsolete, and unused

The following keywords in the `*ACES2` namelist are listed for completeness, but they correspond to untested and/or experimental code.

SOLVENT = (integer) 0

> Sets the dielectric constant used to determine the orbitals. A cavity size may be specified as well by creating a file named "radius", which is read by the SCF code. This contains the cavity radius in Å. If the file is not present, the program uses a value calculated from 0.5 Å plus half of the longest internuclear distance.

TURBOMOLE = (switch) OFF

POLYRATE = (switch) OFF

CCR12 = (switch) OFF

KUCHARSKI = (switch) OFF

KS_POT *Obsoleted by the* ****VSCF*** *namelist.*

FUNCTIONAL *Obsoleted by the* ****INTGRT*** *namelist.*

EOMXFIELD

EOMYFIELD

EOMZFIELD

RDO

FILE_STRIPE

RESET_FLAGS

PSI

INSERTF

GLOBAL_MEM

A.2 Kohn-Sham DFT namelists

A.2.1 *VSCF

KS = (switch) ON

Controls the type of SCF. KS=ON performs a Kohn-Sham DFT calculation with numerical integration, and KS=OFF performs the standard analytical Hartree-Fock SCF with the KS SCF program `xvscf_ks`. The `*ACES2` keyword SCF_TYPE must be set to KS *in addition* to this switch, which defaults to ON.

A.2.2 *INTGRT

POTRADPTS = (integer) 50

RADTYP = (handle) Handy

Handy

Gauss-Legendre

PARTPOLY = (handle) bsrad

equal

bsrad

dynamic

RADSCAL = (handle) Slater

none

Slater

PARTTYP = (handle) fuzzy

rigid

fuzzy

FUZZYITER = (integer) 4

RADLIMIT = (real) 3.0

NUMACC = (switch) ON

EXACT_EX = (switch) OFF

TDKS = (switch) OFF

ENEGRID = (integer) 4

POTGRID = (integer) 4

ENETYPE = (handle) lebedev

POTTYPE = (handle) lebedev

FUNC = (string) none

This keyword can accept multiple types of strings. The regular expression is

exch(,coeff)?(,corr(,coeff)?)?

This means one (exchange) or two (exchange and correlation) functionals can be used to evaluate the total SCF energy, and each can have an optional coefficient (default is 1.0). To use B3LYP, the format must be "func=b3lyp" with no coefficients or additional correlation functionals.

The following exchange (and hybrid) functionals are available:

lda	LDA (Slater, Xalpha) (exchange)
becke	Becke (exchange)
hf	Exact Nonlocal Exchange (exchange)
pbe_ex	Perdew-Burke-Ernzerhof (exchange)
pw91_ex	Perdew-Wang 91 (exchange)
b3lyp	Becke III LYP (hybrid)

The following correlation functionals are available:

vwn	Vosko-Wilk-Nusair
lyp	Lee-Yang-Parr
pbe_cor	Perdew-Burke-Ernzerhof
pw91_cor	Perdew-Wang 91
wl	Wilson-Levy
wi	Wilson-Ivanov

KSPOT = (string) hf

The KSPOT value string takes the same format as FUNC. The available exchange potentials are: lda, becke, and hf. The available correlation potentials are: vwn and lyp.

CUTOFF = (tol) 12

A.3 mrcc namelists

A.3.1 *mrcc_gen, *true_mrcc, *cse

A.3.2 *EE_EOM, *EE_TDA, *EE_STEOM

A.3.3 *IP_EOM, *IP_CI, *DIP_EOM, *DIP_TDA, *DIP_STEOM

A.3.4 *EA_EOM, *EA_CI, *DEA_EOM, *DEA_TDA, *DEA_STEOM

A.3.5 *ACT_EA_EOM

B Standard Basis Sets and ECPs

B.1 Basis sets in GENBAS

The following is a listing of the basis sets currently included in the standard GENBAS file.

STO-3G This is the well known minimal basis set developed by Pople and coworkers in the early 1970s. entries are in the GENBAS file for all atoms from H to Cl. STO-3G basis sets are available in the literature for the third and fourth row main group elements as well as some transition metal elements. The STO-3G basis is now largely obsolete, except for some calculations on large molecules. Its use is not recommended for other than testing and rough preliminary investigations.

3-21G This is a small "split valence" basis set developed by Pople and coworkers. It uses a minimal basis (or single zeta) description for the core orbitals and a double-zeta description for the valence orbitals. Entries are in the GENBAS file for H-Cl. 3-21G sets are available in the literature for heavier elements. This and other sets of this type (such as 4-31G and 6-31G) are often termed "double zeta valence". This is not strictly accurate since the s and p exponents are constrained to be equal, which they are not in a true double zeta set, although the two sets give results of similar quality.

4-31G This is similar to the 3-21G set, but with more primitive gaussian functions. Entries are available for H-Cl.

6-31G This is yet another split valence set, employing still more primitive gaussians. Entries are available for H-Cl.

6-311G This is a split valence set with a triple-zeta description of the valence orbitals and a minimal basis set description of the core orbitals. Entries are available for H-Ne. It has been claimed in the literature that this basis set is not really of triple-zeta valence quality.

6-31G** 6-31G supplemented with d polarization functions (p functions for H and He). It should be used with the SPHERICAL=OFF option ($6d$ functions) since this is how the set was defined and developed. Entries are available for H-Cl. The 6-31G* basis set excludes p functions from H and He, while retaining d functions on other atoms. Polarization exponents are as follows: H, He (1.1); Li (0.2); Be (0.4); B (0.6); C-Ne (0.8); Na, Mg (0.175); Al (0.325); Si (0.45); P (0.55); S (0.65); Cl (0.75).

6-311G** 6-311G supplemented with polarization functions. This should be used with the SPHERICAL=ON option ($5d$ functions). Entries are available for H-Ne. Polarization

exponents are: H, He (0.75); Li (0.2); Be (0.401); C (0.626); N (0.913); O (1.292); F (1.75); Ne (2.304). This basis set was developed for correlated calculations.

DZ This is the well known Dunning double-zeta contraction of Huzinaga's $9s5p$ primitive gaussian basis set for first row atoms. Entries are available for H, B-F.

DZP This is the DZ set augmented with the polarization functions recommended by Redmon, Purvis, and Bartlett, which were determined from correlated calculations. Entries are available for H, B-F. The polarization exponents are: H (0.7); B (0.386); C (0.654); N (0.902); O (1.211); F (1.580).

D95 This is the same as DZ for H, B-F but also includes $6s4p$ contractions by Dunning and Hay of Huzinaga's $11s7p$ primitive set for Al, Si, P, S, and Cl. The published data for Cl are erroneous, having at least two typographical errors (see W.Craven et al, Chem. Phys. Lett. **116**, 119 (1985)). It is not clear if these are the only errors.

D95* This the D95 set augmented with a set of polarization exponents of uncertain origin. Entries are available for H, B-F, and Al-Cl. The polarization exponents for the first row elements appear to be those recommended by Dunning and Hay, while the source of the second row d exponents is uncertain. The exponents are: H (1.0); B (0.7); C (0.75); N (0.8); O (0.85); F (0.9); Al (0.25); Si (0.3247); P (0.37); S (0.532); Cl (0.6).

TZ2P For first row elements this set comprises Dunning's $5s3p$ contraction of Huzinaga's $10s6p$ primitive set augmented with two optimized d (p for H) functions in a (2,1) contraction of three primitives. Entries are available for H, B-F. The F basis is not optimized. There is also an entry for Cl under this name. This is based on a McLean and Chandler sp set.

PVDZ Dunning's polarized valence double-zeta correlation consistent basis set. In terms of contracted functions this set is $3s2p1d$ for second row atoms, with one fewer shell of each angular momentum for H and He, and one extra shell plus an f function for Na-Cl. These and the PVTZ and PVQZ sets are hybrids between segmented and generally contracted sets. All of these sets appear to be best used with the SPHERICAL=ON option. Indeed, whether SPHERICAL is ON or OFF has a significant effect on the results. Entries are available for H, He, B-Ne, and Al-Ar. We do not recommend the PVDZ set. For little extra cost one may use the DZP basis set and obtain significantly improved results.

PVTZ Dunning's polarized valence triple-zeta correlation consistent basis ($4s3p2d1f$ for second row atoms). Entries are available for H, He, B-Ne, Al-Ar.

PVQZ Dunning's polarized quadruple-zeta valence correlation consistent basis set ($5s4p3d2f1g$ for second row atoms). Entries are available for H,He, B-Ne, and Al-Ar.

PV5Z Dunning's polarized pentuple-zeta valence correlation consistent basis set ($6s5p4d3f2g1h$ for second-row atoms). Entries are available for H, B-F, Al-Ar.

WMR Generally contracted basis functions developed by Widmark, Malmqvist and Roos for the study of molecular and atomic properties. These are rather large basis sets [$6s4p3d$ for H, $7s4p3d$ for He, $7s6p4d3f$ for second row atoms], and can be reduced to normal size through use of the GENBAS_X keywords. A limited amount of experience with these basis sets suggests that valence double-zeta and valence triple-zeta contractions [$3s2p1d$ and $4s3p2d1f$ for first row atoms, $2s1p$ for H and He] work reasonably well. For a given level of contraction, these basis sets appear to provide superior molecular structures and properties to the correlation consistent sets of Dunning. Entries are available for H-Ne.

PBS These are double-zeta plus diffuse basis sets developed by Sadlej for the calculation of electrical properties. They seem to do a good job of predicting dipole moments and polarizabilities and are also useful in excited state calculations, where the first member of a Rydberg series is usually recovered.

TZP This is a triple-zeta valence plus polarization basis set. Entries are currently available for H,B-Ne,Na, and Mg. For H it comprises Dunning's $3s$ contraction of Huzinaga's $5s$ primitive set augmented with the p exponent of Redmon, Purvis, and Bartlett (0.7). For B-F it comprises Dunning's $5s3p$ contraction of Huzinaga's $10s6p$ primitive set augmented with the polarization exponents of Redmon, Purvis, and Bartlett (see entry DZP above). For Ne the basis set is the Dunning $5s3p$ set augmented with a polarization exponent of 1.9 (an estimate based on the values of Redmon et al). For Na and Mg the basis set is $6s5p1d$, with the sp part coming from McLean and Chandler and d exponents of 0.1 and 0.2 (reasonable estimates).

$5s4p1d$ As TZP but the sp part Dunning's $5s4p$ contraction of Huzinaga's $10s6p$ primitive. Entries are available for B-Ne.

VDZP This is a valence double-zeta plus polarization basis set for all first row atoms except He, taken from Dunning and Hay. It has the advantage that sets for Li and Be exist, in contrast to the DZP set. However, it has not been well tested. The polarization exponents are those of Redmon et al for H,B-F while estimates of 0.2, 0.3, and 1.9 are used for Li, Be, and F.

svp,dzp,tzp,tzplarge,qz2p,... These are the new basis sets from Schäfer, Horn and Ahlrichs which have been optimized for atoms and supplemented by suitable polarization functions. Note that these basis sets are denoted in the `GENBAS` file by lower case letters They are recommended, in particular, for chemical shift calculations and should in the long run replace the old and not completely optimized Dunning-Huzinaga basis sets (denoted by DZP and TZ2P) in the `GENBAS` file. Basis sets are, in principle, available for all atoms. If a needed basis set is not included in the `GENBAS` file, it can be obtained via FTP.

The tables on the following pages list the numbers of generally contracted AO basis functions for each element in each set. The negative subscript shows the number of redundant Cartesian AOs. For example, oxygen in the CC-PVQZ basis set (listed as 70_{-15}) has 70 Cartesian AOs, but only 55 if spherical harmonics are used (with SPHERICAL=ON in the `*ACES2` namelist).

Table 1: The number of contracted AO basis functions for each basis set in `GENBAS` (for 1s through 4s elements).

Basis Set	1s		2s		2p						3s		3p				
	H	HE	LI	BE	B	C	N	O	F	NE	NA	MG	AL	SI	P	S	Cl
STO-2G	1		5		5						9		9				
STO-3G	1		5		5						9		9				
STO-6G	1		5		5						9		9				
STO-3G*	1		5		5						15_{-1}		15_{-1}				
MINI	1		2		5						6		9				
MINI-SCALED	1		2		5						6		9				
MIDI	2		3		9						7	•	13				
MIDI!	2	•	•		•	9		15_{-1}		•	•		•		19_{-1}		
3-21G	2		9		9						13		13				
3-21G*	2		9		9						19_{-1}		19_{-1}				
3-21++G	3	2	13		13						17		17				
3-21++G*	3	2	13		13						23_{-1}		23_{-1}				
3-21GSP	2		9		9						13		13				
4-22GSP	2		9		9						13		13				
4-31G	2		9		9						•		•		13		
6-31G	2		9		9						13		13				
6-31G*	2		15_{-1}		15_{-1}						19_{-1}		19_{-1}				
6-31G**	5		15_{-1}		15_{-1}						19_{-1}		19_{-1}				
6-31+G*	2		19_{-1}		19_{-1}						23_{-1}		23_{-1}				
6-31++G	3	2	13		13						17		17				
6-31++G*	3	2	19_{-1}		19_{-1}						23_{-1}		23_{-1}				

Table 1: The number of contracted AO basis functions for each basis set in `GENBAS` (for 1s through 4s elements).

Basis Set	1s		2s		2p						3s		3p				
	H	HE	LI	BE	B	C	N	O	F	NE	NA	MG	AL	SI	P	S	CL
6-31++G**	6	2	19_{-1}		19_{-1}						23_{-1}		23_{-1}				
6-31G(3DF,3PD)	17_{-1}		37_{-6}		37_{-6}						41_{-6}		41_{-6}				
6-311G	3		13		13						21		21				
6-311G*	3		19_{-1}		19_{-1}						27_{-1}		27_{-1}				
6-311G**	6		19_{-1}		19_{-1}						27_{-1}		27_{-1}				
6-311+G*	3		23_{-1}		23_{-1}					19_{-1}	•		•				
6-311++G**	7	6	23_{-1}		23_{-1}					19_{-1}	•		•				
6-311++G(2D,2P)	10	9	29_{-2}		29_{-2}						37_{-2}		37_{-2}				
6-311G(2DF,2PD)	15_{-1}		35_{-5}		35_{-5}						•		•				
6-311++G(3DF,3PD)	19_{-1}	18_{-1}	45_{-6}		45_{-6}						53_{-6}		53_{-6}				

Table 1: The number of contracted AO basis functions for each basis set in `GENBAS` (for 1s through 4s elements).

Basis Set	1s		2s		2p						3s		3p				
	H	HE	LI	BE	B	C	N	O	F	NE	NA	MG	AL	SI	P	S	CL
SV	2	•	9		9						•		•				
DZ	2	•	10	•	10						•		18				
TZ	3	•	4	5	14						•		•				
SVP	5	•	15_{-1}		15_{-1}						•		•				
DZP	5	•	16_{-1}	•	16_{-1}						•		24_{-1}				
CC-PVDZ	5		15_{-1}		15_{-1}						19_{-1}		19_{-1}				
CC-PVTZ	15_{-1}		35_{-5}		35_{-5}						39_{-5}		39_{-5}				
CC-PVQZ	35_{-5}		70_{-15}		70_{-15}						74_{-15}		74_{-15}				
CC-PV5Z	70_{-15}		126_{-35}		126_{-35}						130_{-35}		130_{-35}				
CC-PV6Z	126_{-35}		•		210_{-70}						•		214_{-70}				
CC-PCVDZ	•		19_{-1}	•	19_{-1}						29_{-2}		29_{-2}				
CC-PCVTZ	•		49_{-6}	•	49_{-6}						69_{-10}		69_{-10}				
CC-PCVQZ	•		104_{-20}	•	104_{-20}						139_{-30}		139_{-30}				
CC-PCV5Z	•		•		195_{-50}						•		•				
CC-PCV6Z	•		•		•			335_{-105}	•	335_{-105}	•		•				
CC-PWCVDZ	•		•		19_{-1}						•		29_{-2}				
CC-PWCVTZ	•		•		49_{-6}						•		69_{-10}				
CC-PWCVQZ	•		•		104_{-20}						•		139_{-30}				
CC-PWCV5Z	•		•		195_{-50}						•		251_{-70}				
AUG-CC-PVDZ	9		•		25_{-2}						•		29_{-2}				
AUG-CC-PVTZ	25_{-2}		•		55_{-9}						•		59_{-9}				

Table 1: The number of contracted AO basis functions for each basis set in `GENBAS` (for 1s through 4s elements).

Basis Set	1s		2s		2p						3s		3p				
	H	HE	LI	BE	B	C	N	O	F	NE	NA	MG	AL	SI	P	S	CL
AUG-CC-PVQZ	55_{-9}		•		105_{-25}						•		109_{-25}				
AUG-CC-PV5Z	105_{-25}		•		182_{-55}						•		186_{-55}				
AUG-CC-PV6Z	182_{-55}		•		266_{-90}						•		270_{-90}				
AUG-CC-PCVDZ	•		•		29_{-2}					•	•		39_{-3}				
AUG-CC-PCVTZ	•		•		69_{-10}						•		89_{-14}				
AUG-CC-PCVQZ	•		•		139_{-30}						•		174_{-40}				
AUG-CC-PCV5Z	•		•		251_{-70}					•	•		•				

Table 1: The number of contracted AO basis functions for each basis set in `GENBAS` (for 1s through 4s elements).

Basis Set	1s		2s		2p						3s		3p				
	H	HE	LI	BE	B	C	N	O	F	NE	NA	MG	AL	SI	P	S	CL
CC-PVDZ_DK	5		•		15_{-1}						•		19_{-1}				
CC-PVTZ_DK	15_{-1}		•		35_{-5}						•		39_{-5}				
CC-PVQZ_DK	35_{-5}		•		70_{-15}						•		74_{-15}				
CC-PV5Z_DK	70_{-15}		126_{-35}		126_{-35}						•		130_{-35}				
GAMESS-VTZ	3	•	•	14	14						21		21				
GAMESS-PVTZ	6	•	•	20_{-1}	20_{-1}						•		•				
CHIPMAN	7	•	•		27_{-2}					•	•		•				
AHLRICHS-VDZ	2		3		9						7		13				
AHLRICHS-PVDZ	5		6	9	15_{-1}						10		19_{-1}				
AHLRICHS-VTZ	3		6		15						13		22				
AHLRICHS-TZV	•		5		14						11		17				
AHLRICHS-POL	3		3	6	6_{-1}						3		6_{-1}				
PARTRIDGE-1	•		14		41						48		53				
PARTRIDGE-2	•		16		49						52		57				
PARTRIDGE-3	•		18		57						56		65				
PARTRIDGE-4	•		•		•						•		•				
WTBS	•	1	2		5						6		9				
SADLEJ-PVTZ	9	•	26_{-2}		•	26_{-2}				•	34_{-2}		•	34_{-2}			
WACHTERS+F	•		•		•						•		•				
ROOS-ADZP	9		25_{-2}		25_{-2}						29_{-2}		29_{-2}				

Table 1: The number of contracted AO basis functions for each basis set in `GENBAS` (for 1s through 4s elements).

Basis Set	1s		2s		2p						3s		3p				
	H	HE	LI	BE	B	C	N	O	F	NE	NA	MG	AL	SI	P	S	CL
ROOS-ATZP	25_{-2}		55_{-9}		55_{-9}						59_{-9}		59_{-9}				
NASA-AMES	35_{-5}	•	•		70_{-15}						•		74_{-15}	•	74_{-15}		
BAUSCHLICHER-ANO ???																	
DGAUSS-DZVP	2		12_{-1}		15_{-1}						19_{-1}		19_{-1}				
DGAUSS-DZVP2	5		12_{-1}		15_{-1}					•	•		19_{-1}				
DGAUSS-TZVP	6	•	•		•	19_{-1}				•	•		23_{-1}				
DGAUSS-A1-COULOMB	4		19_{-1}		34_{-3}						39_{-3}		45_{-4}				
DGAUSS-A1-EXCHANGE	4		19_{-1}		34_{-3}						39_{-3}		45_{-4}				
DGAUSS-A2-COULOMB	13_{-1}		19_{-1}		44_{-4}						39_{-3}		45_{-4}				
DGAUSS-A2-EXCHANGE	13_{-1}		19_{-1}		44_{-4}						39_{-3}		45_{-4}				
DEMON-COULOMB	7		34_{-3}	19_{-1}	34_{-3}						36_{-3}	39_{-3}	45_{-4}				
AHLRICHS-COULOMB	15_{-1}	8	35_{-5}		44_{-6}				50_{-7}	43_{-6}	33_{-5}		36_{-5}	51_{-11}			
IGLO-II	6	•	•		23_{-1}					•	•		37_{-2}				
IGLO-III	10	•	•		37_{-2}					•	•		47_{-3}				
ECP basis sets																	
HAY-WADT-MB	•		•		•						•		•				
HAY-WADT-VDZ	•		•		•						•		•				
LANL2DZ	2	•	9		9						8		8				
LANL2DZDP	6	•	•		•	18_{-1}				•	•		•	17_{-1}			
SBKJC	2		8		8						8		8				
CRENBL	4	•	16		16						18		16				

Table 1: The number of contracted AO basis functions for each basis set in `GENBAS` (for 1s through 4s elements).

Basis Set	1s		2s		2p						3s		3p				
	H	HE	LI	BE	B	C	N	O	F	NE	NA	MG	AL	SI	P	S	Cl
CRENBS ???	•		•		•						•		•				
ST-RLC	•		8		8			11		44_{-6}	8		8			11	
ST-RSC97	•		•		•						•		•				

Table 2: The number of contracted AO basis functions for each basis set in `GENBAS` (for 3d through 5s elements).

Basis Set	3d										4p				
	SC	TI	V	CR	MN	FE	CO	NI	CU	ZN	GA	GE	AS	SE	Br
STO-2G	•										•				
STO-3G	19_{-1}										19_{-1}				
STO-6G	19_{-1}										19_{-1}				
MIDI!	•										•				29_{-}
3-21G	29_{-2}										23_{-1}				
6-31G	29_{-2}										•				
6-31G*	39_{-5}										•				
6-31G**	39_{-5}										•				
6-311G	•										41_{-2}				
6-311G*	•										47_{-3}				
6-311G**	•										47_{-3}				
CC-PVDZ	•										29_{-2}				

Table 2: The number of contracted AO basis functions for each basis set in `GENBAS` (for 3d through 5s elements).

Basis Set	$3d$										$4p$				
	SC	TI	V	CR	MN	FE	CO	NI	CU	ZN	GA	GE	AS	SE	BR
CC-PVTZ	•										49_{-6}				
CC-PVQZ	•										84_{-16}				
CC-PV5Z	•										140_{-36}				
AUG-CC-PVDZ	•										39_{-3}				
AUG-CC-PVTZ	•										69_{-10}				
AUG-CC-PVQZ	•										119_{-26}				
AUG-CC-PV5Z	•										196_{-56}				
CC-PVDZ_DK	•										29_{-2}				
CC-PVTZ_DK	•										49_{-6}				
CC-PVQZ_DK	•										84_{-16}				
CC-PV5Z_DK	•										140_{-36}				
AHLRICHS-VDZ	23_{-2}										29_{-2}				
AHLRICHS-PVDZ	26_{-2}										35_{-3}				
AHLRICHS-VTZ	41_{-3}										38_{-2}				
AHLRICHS-TZV	33_{-3}										33_{-2}				
AHLRICHS-POL	3										6_{-1}				
PARTRIDGE-1	110_{-9}		104_{-8}	110_{-9}			107_{-8}		110_{-9}		119_{-9}				
PARTRIDGE-2	118_{-8}	118_{-9}	127_{-10}	124_{-10}	133_{-11}		115_{-8}	133_{-11}	124_{-10}	115_{-8}	129_{-10}				
PARTRIDGE-3	119_{-8}			134_{-11}	119_{-8}			131_{-10}	116_{-8}	119_{-8}	•				
PARTRIDGE-4	134_{-11}										•				

Table 2: The number of contracted AO basis functions for each basis set in `GENBAS` (for 3d through 5s elements).

Basis Set	3*d*										4*p*				
	SC	TI	V	CR	MN	FE	CO	NI	CU	ZN	GA	GE	AS	SE	BR
WTBS	16_{-1}										19_{-1}				
SADLEJ-PVTZ	•										•				74_{-1}
WACHTERS+F	60_{-7}									•	•				
ROOS-ADZP	65_{-10}										•				
ROOS-ATZP	119_{-26}										•				
NASA-AMES	•	69_{-10}	•			109_{-25}	•	69_{-10}	•		•				
BAUSCHLICHER-ANO	109_{-25}									•	•				
DGAUSS-DZVP	26_{-2}										29_{-2}				
DGAUSS-DZVP2	29_{-2}										•				
DGAUSS-A1-COULOMB	55_{-5}										55_{-5}				
DGAUSS-A1-EXCHANGE	55_{-5}										55_{-5}				
DGAUSS-A2-COULOMB	55_{-5}										•				
DGAUSS-A2-EXCHANGE	55_{-5}										•				
DEMON-COULOMB	55_{-5}										55_{-5}				
AHLRICHS-COULOMB	96_{-24}	90_{-23}					96_{-24}	91_{-23}			40_{-5}	55_{-11}			
ECP basis sets															
HAY-WADT-MB	11_{-1}					24_{-2}				•	•				
HAY-WADT-VDZ	24_{-2}									•	•				
LANL2DZ	24_{-2}									20_{-2}	8				
LANL2DZDP	•										•	17_{-1}			
SBKJC	34_{-3}										34_{-3}	8			

Table 2: The number of contracted AO basis functions for each basis set in `GENBAS` (for 3d through 5s elements).

Basis Set	$3d$										$4p$				
	SC	TI	V	CR	MN	FE	CO	NI	CU	ZN	GA	GE	AS	SE	BR
CRENBL	61_{-6}										36_{-4}				
CRENBS	7_{-1}										•				
ST-RLC	•									9	8			11	
ST-RSC97	49_{-6}								39_{-3}		•				

Table 3: The number of contracted AO basis functions for each basis set in `GENBAS` (for 4d through 6s elements).

Basis Set	4*d*										5*p*				
	Y	ZR	NB	MO	TC	RU	RH	PD	AG	CD	IN	SN	SB	TE	I
STO-3G	29_{-2}										29_{-2}				
MIDI	•										•				
MIDI!	•										•				39_{-3}
3-21G	39_{-3}										33_{-2}				
6-311G	•										•				61_{-4}
6-311G*	•										•				67_{-5}
6-311G**	•										•				67_{-5}
WTBS	26_{-2}							25_{-2}	26_{-2}		29_{-2}				
SADLEJ-PVTZ	•										•				94_{-12}
DGAUSS-DZVP	39_{-3}										39_{-3}				
DGAUSS-A1-COULOMB	55_{-5}										55_{-5}				
DGAUSS-A1-EXCHANGE	55_{-5}										55_{-5}				
DEMON-COULOMB	55_{-5}										55_{-5}				
AHLRICHS-COULOMB	90_{-23}		96_{-24}	97_{-24}	96_{-24}	97_{-24}	103_{-25}	91_{-23}	97_{-24}	91_{-23}	49_{-11}	56_{-12}	57_{-12}	56_{-12}	49_{-11}
ECP basis sets															
HAY-WADT-MB	24_{-2}									•	•				
HAY-WADT-VDZ	24_{-2}									•	•				
LANL2DZ	24_{-2}									20_{-2}	8				
LANL2DZDP	•										•	17_{-1}			
SBKJC	34_{-3}										34_{-3}	8			
CRENBL	44_{-4}										36_{-4}				

Table 3: The number of contracted AO basis functions for each basis set in `GENBAS` (for 4d through 6s elements).

Basis Set	$4d$										$5p$				
	Y	ZR	NB	MO	TC	RU	RH	PD	AG	CD	IN	SN	SB	TE	I
CRENBS	10_{-1}										•				
ST-RLC	•										8			11	
ST-RSC97	39_{-3}										•				

B.2 ECP sets in ECPDATA

C Queue Scripts

While nothing prevents ACES II programs from running interactively or in the background, many computing facilities require large batch jobs to be executed by automated queueing systems. Documenting every scheduler that users might encounter is far beyond the scope of this manual, but the following sections list the most common options that the developers have used.

C.1 ACES II script body

```
#!/bin/sh
zmt=ZMAT_FILE
out=OUT_FILE
genbas=GENBAS_FILE
workdir=WORK_DIRECTORY
test -d $workdir && rmwd=0 || rmwd=1
test $rmwd -eq 1 && mkdir -p $workdir
cd $workdir
cp $zmt ZMAT
cp $genbas GENBAS
xaces2 > $out
cd -
test $rmwd -ne 0 && rm -rf $workdir
```

C.2 LoadLeveler

```
#@ output = STDOUT_FILE
#@ error  = STDERR_FILE
#@ class = CLASS
#@ job_type = JOB_TYPE
#@ node_usage = USAGE_TYPE
#@ node = NNODES
#@ total_tasks = NTASKS
#@ network.MPI = css0,shared,us
#@ Requirements = (OPT1 == VALUE1) && (OPT2 == VALUE2)
#@ wall_clock_limit = HH:MM:SS
```

C.3 LSF

```
#BSUB -P ACCOUNT
#BSUB -J JOBNAME
#BSUB -o STDOUT_FILE
#BSUB -e STDERR_FILE
#BSUB -q QUEUE -W TIME
#BSUB -n NPES
```

C.4 GridEngine

```
#$ -S SHELL
#$ -N JOBNAME
#$ -o STDOUT_FILE
```

```
#$ -e STDERR_FILE
#$ -l COMPLEX
#$ -pe PE_ENV NPES
```

Index