



Scientific Computing & Modelling

ADF Property Programs

ADF Program System

Release 2004.01

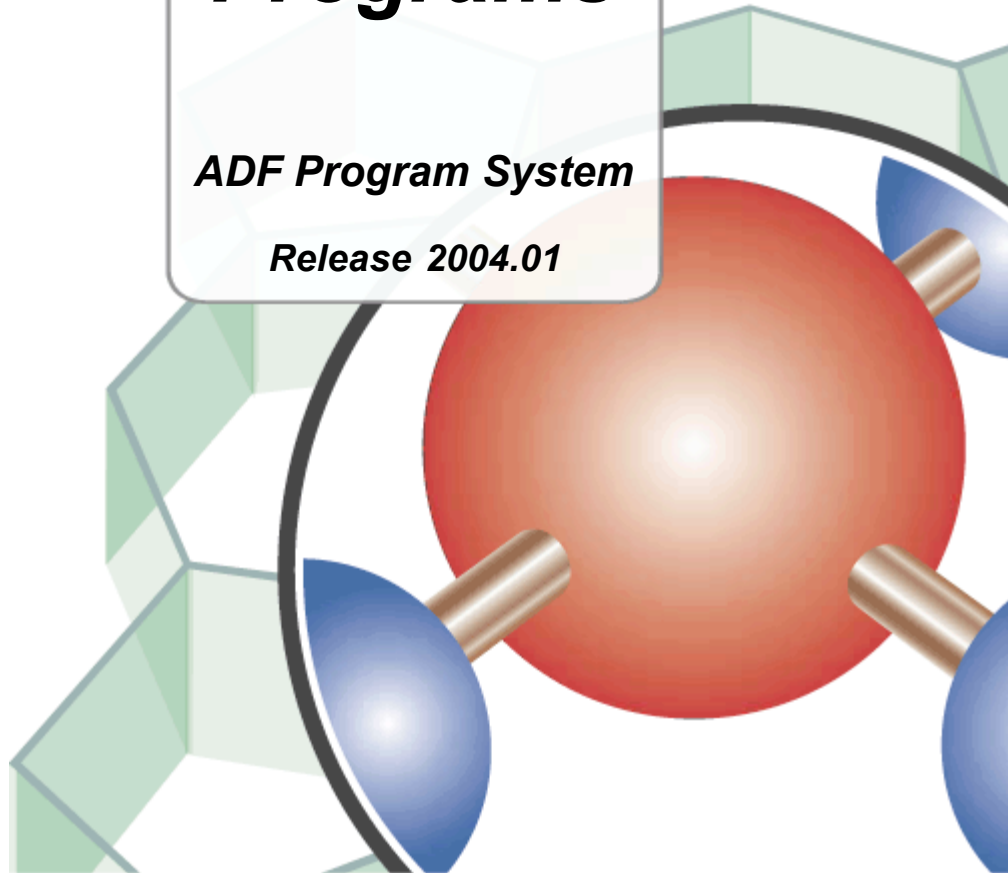


Table of Contents

ADF Property Programs

General introduction

CPL

Introduction

Theoretical and technical aspects

Further technical aspects and current limitations

Running CPL

Input file for CPL: TAPE21

Main input switches

NMRCOUPPING subkeys

Practical Aspects

Minimal input

Restarts

How to avoid the unnecessary computation of many SCF cycles

Computing individual terms in the coupling tensor

Two-bond and more-bond couplings

Principal axis system, the whole coupling tensor

References

EPR/NMR

References

Compatibility and features

Comparison to related functionality in the ADF package

Summary of the input options

Input description

NUCLEI

GHOSTS

CALCVIRTDIA

NOPARA

OUTPUT

MixOccupations / NoMixOccupations

MAGNETIC SUSCEPTIBILITY

EPRGTENSOR

SICOEP

SD

Introduction

Input options for second derivatives program

CALC

PRINT

OPTIONS

MAX_CPKS_ITERATIONS

CHECK_CPKS_FROM_ITERATION

U1_ACCURACY

SAVE_FINAL_U1_TO_FILE

IO_U1_TO_FILE

DEBUG

NMR

NMR: Chemical Shifts

Input options

OUT

CALC
U1K
NUC
ANALYSIS
MAXMEMORYUSAGE

DISPER

Disper: Dispersion Coefficients

References

General introduction

This document describes some programs that can be run after earlier ADF calculation(s) have produced the required result file(s), such as TAPE21. The ADF program itself also enables the calculation of various molecular properties (excitation energies and (hyper)polarizabilities to name a few) that are not mentioned here. Please consult the ADF User's Guide to check for information on the property in which you are interested, if the underlying document does not contain the required information.

The documentation of the ADF program also includes information on ADF utilities. Many of these also use ADF result files, but these executables perform more technical tasks than the programs described here, such as visualization and handling of the output.

There is some overlap in functionality between programs in the ADF suite, regarding the NMR and ESR properties. Each implementation has its own merits and deficiencies. The differences are mentioned in the appropriate places in the documentation, in order to let the user decide which option is most suitable for his problem.

CPL

The original version of this part of the User's Guide was written by Jochen Autschbach, primary author of the CPL code.

Introduction

The CPL code of the Amsterdam Density Functional program system allows the user to calculate *Nuclear Spin-spin Coupling Constants* (NSSCCs) [1,2]. NSSCCs are usually observed in NMR (Nuclear Magnetic Resonance) spectroscopy and give rise to the splitting of the signals of the NMR spectrum in multiplets. They contain a wealth of information about the geometric and electronic structure of the compound being investigated.

The calculation needs a standard TAPE21 ADF output file. CPL reads also an input key and optional settings from stdin (usually from an input file). Technical parameters such as the maximum memory usage can be set here as well.

One of the key features of the program is its ability to treat heavy nuclei with the ZORA relativistic formalism. We refer the reader to the literature for details about our implementation [1,2]. Please use the information printed in the output header of the CPL program in order to provide references of this work in scientific publications.

Please note that the CPL program has been developed only very recently. CPL provides the main functionality in order to evaluate accurate NSSCCs based on DFT, as well as many additional features in order to provide an analysis of the results. Additional features that increase the range of applicability, result in faster program execution, or allow for a more extended analysis of the results will be implemented for future versions. Please report bugs or suggestions to SCM at support@scm.com.

Theoretical and technical aspects

Within the nonrelativistic theory of nuclear spin-spin coupling, there are four terms contributing to the NSSCC between two nuclei A and B: the paramagnetic and diamagnetic orbital terms (OP and OD, respectively), and the electron-spin dependent Fermi-contact (FC) and spin-dipole term (SD). In the literature, the OP and OD terms are often named PSO and DSO (for paramagnetic and diamagnetic spin-orbital). In the more general ZORA formulation, very similar operators are responsible for the NSSCC, therefore we use the same terminology for the individual contributions. In general, the interpretation of the results for a heavy atom system is basically equivalent to a nonrelativistic situation.

In most cases, the FC term yields the most important contribution to the NSSCC. However, many exceptions are known for which one or each of the other terms can be non-negligible or even dominant. We therefore suggest that you always check, at least for a smaller but similar model system, or by using a smaller basis set, which of the four terms are negligible and which are dominant.

By default, the CPL program computes the FC coupling between the first and all other nuclei of the molecule, respectively. Other couplings or the computation of the OP, OD and SD terms can be requested by input switches (see the 'Running CPL' section of this document for details).

All contributions to the NSSCC are evaluated with the help of the numerical integration scheme implemented into ADF. In general, the computation of the OD term is computationally very cheap, since only integrals involving the electron density have to be evaluated. The next expensive term is the OP term. For this contribution, the first-order perturbed MOs have to be computed. With the available density functionals in

ADF, the OP term does not cause a change in the Kohn-Sham potential, and the first-order MOs can be computed directly (i.e. without an iterative procedure). This is equivalent to the approach that has been implemented in the NMR code for ADF.

Both the FC and the SD terms induce electron spin-density to first-order as a perturbation. Equivalent to the iterative solution of the unperturbed Kohn-Sham equations, the first-order MOs depend on that first-order spin-density, which in turn depends on the first-order MOs. Therefore, in order to evaluate the FC and SD NSSCC contributions, the CPL program carries out a SCF cycle. In the scalar or non-relativistic case, the computational cost for the FC term is comparable to an ADF single point calculation with a local density functional. The evaluation of the SD term is more expensive. The current implementation utilizes the CPL spin-orbit code to compute the combined FC+SD contribution and therefore leaves some room for future speed-ups. In most cases, the SD term yields a negligible NSSCC and the much faster code for the scalar- or non-relativistic FC term can be used. However, it is very important to include the SD term in the computation if coupling anisotropies are to be evaluated.

In the case where the NSSCC computation is based on spin-orbit coupled relativistic two component ZORA MOs, the SD term causes only a marginal increase in computational time as compared to the FC term alone. Generally, in this case the computational cost for the FC term is already approximately one order of magnitude higher than in the scalar or non-relativistic case, since the 3 (x, y, z) components of the spin-density with respect to 3 components of the perturbation, respectively, have to be determined self-consistently. The additional presence of the SD term only shows up in a somewhat more costly evaluation of the matrix elements of the perturbation operator. However, CPL spends most of its computational time in the SCF cycle. Therefore, in spin-orbit computations the computation of the FC+SD terms is the default. The OP term has to be evaluated self-consistently, too, in this case and is added as a perturbation in the SCF cycle upon request.

We use the terminology 'perturbing' and 'responding nucleus' within the CPL output. The 'perturbing' nucleus is the one, for which the first-order MOs have to be computed (self-consistently), while the NSSCC is then determined by these first-order MOs and the FC, SD, and OP matrix elements of the second, 'responding' nucleus. For the OD term, this distinction makes no sense but is used in the output for reasons of consistency.

Experimental NSSCCs between two nuclei A and B are usually reported as $J(A,B)$ in Hertz. From a computational point of view, the so-called reduced NSSCCs $K(A,B)$ are more convenient for comparisons. CPL outputs both. The J 's are set to zero in case the nuclear magneto-gyric ratio of one of the nuclei A or B is not available at run time.

Further technical aspects and current limitations

In order to facilitate the future computation of rather large molecules, all matrix elements of the perturbation operators FC, SD, and OP are evaluated in the Slater AO basis that is specified as input in the CREATE runs of ADF. The implementation of cut-off and related features that allow for a nearly linear scaling of the computational cost with system size for very large molecules is under way, but not yet available. The AO matrix elements are further transformed to the basis of MOs and the calculation proceeds within the MO basis. This allows for a convenient analysis of the results in terms of contributions from individual occupied and virtual MOs. Such an analysis can be requested by input.

The matrix elements themselves as well as the first-order contributions to the potential are evaluated by numerical integration. A parallel version of the CPL code that will be able to use multiple processors for these steps of the computation will be available in the future. The accuracy setting for the numerical integration is of high importance to obtain accurate matrix elements. Furthermore, the basis set being employed needs to be flexible enough to describe the perturbation correctly. This means usually that modified basis sets have to be used in particular for heavy element calculations.

The first-order potential is currently approximated by the VWN functional. The Xa potential is available as an

alternative but usually leads to less accurate results.

Currently, only spin-restricted computations for systems with an even number of electrons are supported. Further, the calculation does not make use of symmetry and must be based on an ADF run with input SYMMETRY=NOSYM. Non-Aufbau configurations are not supported. The atom input list must not contain dummy atoms.

With the present version of CPL, the SD term and the FC/SD cross term cannot be evaluated separately. Either, the sum of FC + SD + cross terms, or the FC term individually, are computed.

CPL is restartable after various time-consuming steps of the computation.

Running CPL

Input file for CPL: TAPE21

In order to run the CPL code, you need the general ADF output file TAPE21 being present in the directory where CPL is running. Most of the computation's specific settings will be taken from TAPE21, such as the integration accuracy, the basis set, the density functional being employed, nuclear coordinates, and so on. *That also means that nearly all of the aspects that affect the quality of CPL's results are already determined in the input for the ADF run.* Four aspects are of particular importance here:

1. The numerical integration accuracy: the perturbation operators are large in the vicinity of the nuclei. Therefore, you have to make sure that the integration grid is fine enough in the atomic core regions. We have found that INTEGRATION 6 in the ADF input yields high enough integration accuracy for the CPL code in most cases. In case you can not afford such a high integration accuracy throughout, we suggest the use of the INTEGRATION block key to assure that the integration parameter equals 6 at least in the atomic core regions:

```
INTEGRATION
  ACCINT 4.0 :: or higher
  ACCSPH 6.0
END
```

This should yield a reasonable integration precision for many applications. We do not recommend to use an ACCINT parameter smaller than 4 for obtaining meaningful results and encourage to use higher settings whenever possible.

2. The basis set: NSSCCs are sensitive chemical probes, and therefore flexible basis sets have to be employed in order to yield a valid description of the MOs that determine the NSSCCs. We have found that it is imperative to use at least basis set TZ2P (V) from the ADF basis set database. Additional polarization functions in the valence shell may be necessary. Furthermore, the FC perturbation usually requires additional steep 1s functions (i.e. with exponents much higher than the nuclear charge) for a proper description. In the relativistic heavy element case, the use of additional steep basis functions as compared to the ZORA/TZ2P basis is mandatory. The use of steep functions is only of high importance for those nuclei, for which the NSSCC is to be evaluated.

Currently, no standard basis set suitable for NSSCCs has been added to the ADF basis set directory. We suggest to use basis TZ2P as a starting point and to add some 1s basis functions (and appropriate fit functions) with higher exponents in order to improve the accuracy of the FC term. For some of the heavy NMR nuclei (W, Pt, Hg, Tl, Pb), basis sets suitable for the computation of NSSCCs will be added to the ADF basis set directory in the near future. See Refs. .

For the nuclei for which NSSCCs are to be evaluated, it is necessary to use all-electron basis sets. This is not a restriction due to the implementation, but we have found that, with the available frozen core basis sets, the flexibility of the basis in the vicinity of the nuclei is not sufficient. It is possible to use frozen core basis sets if you add enough basis functions in the core region such that the basis approaches the flexibility

of at least a double-z all-electron basis there [1]. In that sense, the savings in computational time due to usage of a frozen core basis are not as pronounced as in standard ADF computations. Unless reliable frozen-core basis sets for the NSSCC computation are available we strongly discourage the use of frozen core basis sets with the CPL program!

3.The density functional: the results of the CPL code depend mostly on the shape of the MOs that have been determined by ADF, and their orbital energies. Both, in turn, depend on the density functional or Kohn-Sham potential that has been chosen for the ADF run (and the basis set quality). It is difficult to give a general advice here concerning the NSSCCs. So far we have found that the use of GGAs improves the NSSCCs with respect to experiment in most cases in comparison to LDA. Different GGAs often yield very similar results. Further, in particular for those cases for which the OP term is large or even dominant, both standard LDAs and GGAs sometimes do not provide an accurate enough description of the orbitals, and deviation of the CPL results as compared to experiment can be substantial. Future developments of density functionals might be able to cure these problems. For the time being, we recommend that you base the CPL run on different choices of density functionals in the ADF run, and investigate the convergence of the result with respect to basis set and integration accuracy. Note that CPL itself uses the VWN functional by default to determine the first-order perturbed MOs. There are enough indications to believe that this is a reasonable approximation for NMR purposes.

4.Modeling the experimental setup: computing such sensitive numbers as NMR chemical shifts and in particular NSSCCs can result in substantial deviations from experimental data. The simple reason might be that the isolated system that has been computed at zero temperature is not at all a good approximation to the system that has been studied experimentally. We [3,4] and other authors have found that in particular solvent effects can contribute very substantially to the NSSCC. In case you are comparing CPL results to experimental data obtained in strongly coordinating solvents we suggest that you consider solvent effects as a major influence. We have found that even weakly coordinating solvents can cause sizeable effects on the NSSCCs for coordinatively unsaturated metal complexes. Other sources of errors can be the neglect of vibrational corrections to the NSSCCs (usually in the range of a few percent).

If the parameters of the underlying ADF computation are carefully chosen and the density functional is able to provide an accurate description of the molecule under investigation, it is possible to compute NSSCCs by means of DFT with very satisfactory accuracy (please note that for properties as sensitive as NSSCCs, agreement with experimental results within about 10% error can be regarded as quite good). Further, chemical trends will be correctly reproduced for a related series of molecules in most cases. However, due to the inherent approximate character of the density functionals currently available with ADF, and necessary basis set limitations, great care should be taken that the results are reliable. *CPL assumes Aufbau configurations. Please make sure that there are no empty orbitals with energies below the highest occupied MO (HOMO). In addition, the SYMMETRY NOSYM key has to be used in the ADF computation. It is currently not possible to use dummy atoms in the ADF input if the TAPE21 is intended to be used for a subsequent CPL computation.*

Main input switches

With the ADF output TAPE21 present in the current working directory, the CPL code is invoked by:

```
| $ADFBIN/cpl < input_file
```

where input_file contains the input for CPL. We have tried to ensure some backward compatibility with older ADF versions, such as ADF 1999 and ADF 2.3. Normally, you will use the ADF suite that contains the CPL code of the same version. CPL tries to detect if the TAPE21 belongs to an older version of ADF and exits with an error message in case it is not able to process this file. For ADF 2.3, you have to supply also the TAPE10 of ADF 2.3 in addition to TAPE21 (specify SAVEFILE TAPE10 in the ADF input file). We provide this option for testing purposes, however this functionality is not supported and we do not recommend to run CPL on top of the output of an older ADF version.

input_file must contain at least one block-type input key in order to start the CPL run. The input key is

```
| NMRCOUPLING
| END
```

This represents a minimal input file for CPL. The NMRCOUPLING key hosts all optional keys that are relevant for the NSSCCs themselves. In addition to the mandatory NMRCOUPLING key, CPL recognizes the following input switches:

```
| MAXMEMORYUSAGE nmem
```

maximum amount nmem of memory in Mbyte that the program is allowed to use

```
| RESTART restart_file
```

restart the computation from file restart_file. This is the TAPE13 produced during a CPL run. By default, TAPE13 is deleted after a successful completion of CPL. As with ADF restarts, you can not use the name TAPE13 for restart_file but you have to rename it, e.g. to tape13.restart.)

```
| SAVEFILE TAPE13
```

keep the restart file even after a successful completion of CPL. TAPE13 is currently the only file that is meaningful as a parameter to SAVEFILE

NMRCOUPLING subkeys

The available switches within a NMRCOUPLING/END block control the computation of the NSSCCs. By default, the program will evaluate the FC coupling contribution for the first nucleus being the perturbing nucleus and all remaining nuclei responding. *Please note that the ordering of atoms in CPL is generally different from the ADF input. Currently, there exists no workaround for this, but it will be available in future versions. The ordering of atoms is the one being stored in TAPE21 and it is grouped by fragment types. In case you are in doubt about the ordering of atoms, you can run CPL for a few seconds. It will print a list of atoms with their coordinates. The ordering is currently the same as required the NMR program in the ADF program system.* Available subkeys are:

```
| NMRCOUPLING
| NUCLEI {npert nresp1 nresp2}
| GAMMA {nnuc gamma}
| DSO
| PSO
| SD
| FC
| SCF {ITERATIONS=25 | NOCYCLE | CONVERGE=1e-4 }
| XALPHA
| CONTRIBUTIONS {1E19}
| END
```

NUCLEI {npert nresp1 nresp2}

Use nucleus no. npert as the perturbing nucleus, and nuclei nresp1, nresp2, etc as responding nuclei. You can supply more than one NUCLEI keys, in which case CPL evaluates the first-order MOs for each perturbing nucleus that is specified and computes the NSSCCs between all specified responding nuclei. For each NUCLEI line in the input, CPL has to perform an SCF cycle.

GAMMA {nnuc gamma}

Input a non-default magneto-gyric ratio of $g = \text{gamma}$ for nucleus no. nnuc, in units of $10^7 \text{ rad}/(\text{T s})$. CPL normally uses the g value of the most abundant NMR active isotope for a nucleus of a given charge by

default. With the GAMMA keyword you can override this value or supply a value if CPL does not know about it. A list of g's that is used in the computation is printed in the output. You have to provide the GAMMA key for each nucleus you want to specify.

DSO

Compute the diamagnetic orbital term for each NSSCC that is requested (not default)

PSO

Compute the paramagnetic orbital term for each NSSCC (not default)

SD

Compute the SD term for each NSSCC. This is only default for spin-orbit ADF runs. The output will contain the sum of the FC and SD contributions. *Please note that requesting this option results in a greatly increased computational cost in scalar or non-relativistic runs.* The option "NOSD" will turn the SD computation off in spin-orbit runs and has no effect otherwise.

FC

Compute the FC contribution to the NSSCCs. This is the default option. Please note that it is currently not possible to compute the SD term without the FC term. Consult the "practical aspects" section for instructions how to estimate the FC/SD cross term. The option "NOFC" will disable both the FC and SD computation.

SCF options

Settings related to the SCF cycle that is carried out by CPL. Valid options are (with default values if applicable):

ITERATIONS 25

maximum number of iterations

NOCYCLE

perform no cycle, equivalent to ITERATIONS 0

CONVERGE 1e-4

convergence criterion, an input of e corresponds approximately to a convergence of log(-e) digits, i.e. the results will be converged to about four significant digits by default. The measurement for the convergence is based on the sum S of the magnitudes of all occupied-virtual matrix elements of the induced first-order exchange potential. Note that the actual convergence criterion being used in the computation is e times S of the first cycle, i.e. the convergence criterion is set relative to the initial value of S.

XALPHA

Use first-order Xalpha potential instead of VWN potential. This will usually decrease the accuracy for couplings involving hydrogen, and does not have a large effect for couplings between heavier nuclei (not default). The key is mainly intended to ensure compatibility with our previously published results.

CONTRIBUTIONS {1e19}

Print contributions from individual orbitals to the FC and OP term of the NSSCCs that are larger in magnitude than a certain threshold. The threshold refers to the reduced coupling constant K in SI units (not default).

Practical Aspects

Minimal input

The default settings for CPL are invoked by the simple minimal content of the input file:

```
| NMRCOUPLING
| END
```

This is equivalent to

```

NMRCOUPLING
  NUCLEI 1 2 3 4 5 6 7 8 ..(up to number of atoms)
  SCF CONVERGE 1e-4 ITERATIONS 25
  FC
END

```

Restarts

CPL is restartable after the computation of each the complete set of FC or FC/SD and OP matrix elements, and after their transformation to the MO basis. Further, in spin-orbit runs or in scalar- or nonrelativistic computations involving the SD term, CPL is restartable after each SCF cycle. Currently, restart of the SCF in standard scalar or nonrelativistic computations without the SD term is not implemented. As with ADF restarts, you need to supply a proper input file for a restarted computation, and the restart file TAPE13 (which needs to be renamed). Changing the input of a calculation for a restart is not supported. In restarted runs, the program will automatically continue at the latest possible point before the execution stopped, and changing the input between restarts can cause inconsistencies that may lead to a crash.

Unless you are computing a very large molecule, the most likely need for a restart will probably occur during a computation of the FC/SD SCF cycle. We have already mentioned that this is a very time consuming part of the computation, and for this reason CPL can be restarted after each completed SCF cycle. The convergence of the results should not be affected by a restart. You can, e.g., use this in order to complete a lengthy CPL computation in case you have tight time limits in your queuing system, or after a power loss.

How to avoid the unnecessary computation of many SCF cycles

As already mentioned, once the first-order MOs with respect to the perturbation by one of the nuclear spins have been determined, the NSSCC between this and all other nuclei can be computed rather quickly. For each nucleus that participates in at least one of the coupling constants to be determined, the matrix elements of the FC, SD, and OP operators have to be evaluated once (unless the computation of the respective terms is disabled).

You can use this information in order to minimize the number of nuclei for which an SCF cycle has to be performed. This can lead to a great speedup of the computation. The final result, the NSSCC between A and B, does not depend on which nucleus has been chosen as the 'perturbing' one, and which as the 'responding' one (convergence has to be good enough, though). Suppose you want to compute the NSSCCs in the water molecule, with O being nucleus no. 1. In that case,

```

| NUCLEI 1 2 3
| NUCLEI 2 3

```

yields the same O-H and H-H coupling constants as the input

```

| NUCLEI 2 1
| NUCLEI 1 3
| NUCLEI 3 2

```

but with less computational effort due to the fact that only 2 instead of 3 SCF cycles will be performed. The example chosen here is trivial, but in other cases it can be worthwhile to consider different sequences of computations.

Computing individual terms in the coupling tensor

As we have mentioned before, the FC, OP and OD terms can be calculated individually, but not the SD term. In case the SD input option is given, the FC+SD contribution is evaluated instead. This is NOT equal to the sum of the individual FC and SD contributions since there is a cross term between these two. Due to computational simplicity and efficiency, CPL evaluates either the matrix elements for the FC operator, or the combined ones for FC+SD. The final result therefore contains either FC only, or FC, SD plus the cross terms. Only the latter, in addition to the OP and OD contributions, should be compared to experimental results. We will implement the computation of the individual SD term in a future version of CPL in order to assist the analysis of the CPL results.

Likewise, in a spin-orbit based relativistic computation, there exists a cross term between the spin-dependent FC and SD terms, and the OP term. In the scalar- or nonrelativistic limit, this contribution is always zero. With the PSO option present, CPL computes the FC, SD and OP terms including all cross contributions. Even though the output suggests that the individual OP and FC+SD terms are printed, they contain additional cross terms if spin-orbit coupling is large. You can run CPL with the options

```
| NMRCOUPLING  
| NOFC  
| NOSD  
| PSO  
| END
```

in order to evaluate the individual OP contribution(s). In a second run, you can then compute just the FC+SD contributions. The differences between these two CPL runs and a third one with all three terms present yields the relativistic (FC+SD)/OP cross term.

Two-bond and more-bond couplings

CPL does not discriminate between one-bond and two-bond couplings etc. in any technical sense. Even though we [1-4] have validated the code mostly for one-bond NSSCCs, the coupling between any pair of nuclei in the molecule can be computed. See Ref. [4] for an example.

Principal axis system, the whole coupling tensor

CPL evaluates the complete 3x3 coupling tensor with respect to the Cartesian input coordinate system. Depending on the orientation of the molecule, and the local symmetry, the coupling tensor has in fact often only a small number of independent components. CPL evaluates the 'principal components' by the following procedure: the 3x3 matrix is transformed into the basis of the eigenvectors of its symmetric part. This diagonalizes the symmetric part of the coupling tensor. A set of eigenvectors (= 'principal axis system') is also printed.

References

- [1] Autschbach, J.; Ziegler, T. *J Chem Phys* 2000, 113, 936.
- [2] Autschbach, J.A.; Ziegler, T. *J Chem Phys* 2000, 113, 9410.
- [3] Autschbach, J.A.; Ziegler, T. *J Am Chem Soc* 2001, 123, 3341.
- [4] Autschbach, J.A.; Ziegler, T. *J Am Chem Soc* 2001, 123, 5320.

See also:

- N. F. Ramsey, *Phys. Rev.* 91, 303 (1953).
- Dickson, R.M.; Ziegler, T. *J Phys Chem* 1996, 100, 5286.
- Khandogin, J.; Ziegler, T. *Spectr Acta A* 1999, 55, 607.
- D. L. Bryce, R. Wasylshen, *J. Am. Chem. Soc.* 122, 3197 (2000).

ADF User's manual, SCM, Vrije Universiteit, Amsterdam, The Netherlands.
G. Schreckenbach, S. K. Wolff, T. Ziegler, Modeling NMR chemical shifts, ACS Symposium Series, Washington DC (1999).

EPR/NMR

The EPR/NMR part of this User's Guide is adapted from a document originally written by G. Schreckenbach, and later updated by S. Patchkovskii, the two main authors of this program.

References

The original implementation is documented fairly thoroughly in Georg Schreckenbach's Ph.D. thesis (University of Calgary, 1996), which is available at <http://www.scm.com/Doc/publist.html>, as well as in the following papers:

G. Schreckenbach and Tom Ziegler,

[5] J Phys Chem 1995, 99, 606. (NMR all-electron formulation)

[6] Int J Quantum Chem 1996, 60, 753. (frozen core approximation)

[7] Int J Quantum Chem 1997, 61, 899. (scalar relativistic method)

[8] J Phys Chem A 1997, 101, 3388. (ESR g-tensor)

[9] Patchkovskii, S.; Ziegler, T. J Phys Chem 2001, A105, 5490. (High-spin EPR g-tensor)

There are also various application papers published. These papers illustrate some of the concepts and functionality that is described here.

Compatibility and features

The stand-alone EPR/NMR program is compatible with ADF 2000, and the more recent versions. For ADF 2000 (or more recent) only a TAPE21 from a self-consistent ADF calculation is required on input. The program will also accept TAPE10, which will make the calculation somewhat faster. In either case, the SCF calculation, used to create TAPE21, **must** be done with the 'SYMMETRY NOSYM' keyword.

The program supports calculations of NMR shielding tensors of closed-shell molecules, and EPR g-tensors of open shell molecules. Low-spin and high-spin EPR g-tensors can be calculated. Both non-relativistic and scalar Pauli Hamiltonians are supported. Spin-orbit and ZORA are not supported. A detailed breakdown of the orbital contributions can be provided on output.

Comparison to related functionality in the ADF package

The ADF package contains already other possibilities to calculate NMR properties and ESR (EPR) properties, namely the stand-alone NMR program and the ESR option within the ADF program.

The stand-alone NMR program calculates the NMR shielding tensors of closed-shell molecules not only for the non-relativistic and scalar relativistic Pauli Hamiltonian, but also for the spin-orbit coupled Pauli Hamiltonian and scalar relativistic and spin-orbit coupled ZORA Hamiltonian, using a TAPE21 from a self-consistent ADF calculation. Only in the spin-orbit coupled cases the ADF calculation should use 'SYMMETRY NOSYM' (See ADF user's guide). However, the analysis of the different orbital contributions to the shielding tensor can be done much more extensively in the EPR/NMR program described here.

The ESR option within the ADF program can calculate the EPR g-tensor if spin-orbit coupling is included, either at the Pauli or ZORA level. The calculation must be spin-restricted and there must be exactly one unpaired electron, which means it can calculate only low spin g-tensors, with an effective spin of 1/2. In contrast the EPR/NMR program can also calculate the g-tensor at the unrestricted level, and can also

calculate high-spin g-tensors.

With the ADF program it is also possible to calculate more EPR parameters, namely the nuclear magnetic dipole hyperfine interaction (A-tensor) and the nuclear electric quadrupole hyperfine interaction (Q-tensor), see the ADF user guide and the keywords ESR and QTENS, respectively.

Summary of the input options

The program recognizes the following input keys:

```

CLGEPR
  Nuclei   [ALL|n ...]
  Ghosts
    x1 y1 z1
    [...]
  Subend
  CalcVirtDia (CALCV)
  NoParamagneticShielding (NOPARA)
  Output
    Size                [Minimal|Small|Large|Debug|Bigdebug]
    EprSize              [Minimal|Small|Large|Debug|Bigdebug]
    Criteria             psmal
    Format               [F|D]
    PrincipalAxisRep    [Print|Noprint]
    AntisymmetricTensors
    PrintMat             {B1, SF, COEF}
    SumUpOccVir
    RelativisticShielding [Print|Noprint]
  Subend
  MixOccupations / NoMixOccupations (MIXOCC / NOMIXOCC)
  MagnSusceptibility (MAGNSUSC)
  EprGTensor
    EPRXC               [X-Alpha [par]|None]
    XCCutOff            n
    KinCorrection       [On|Off]
  SubEnd
END
SICOEP
  READV filename
END
MAXMEMORYUSAGE nnn

```

Not all combinations of the input parameters and SCF options (used to create TAPE21) are allowed, or required. See below for the detailed description. Additional keys and subkeys may be recognized, but are not intended for general use.

Input description

The calculation of the NMR shielding tensor(s), ESR (EPR) g-tensor is invoked with a block type key:

```
| CLGEPR
```

If this block key is not present, the program will attempt to read the input from the block key:

```
| NMRSHIELDING
```

Thus, you just put the lines

```
| CLGEPR
| END
```

into the input to start the calculation of the shieldings with some default settings that are described below. Note that the word "shielding" refers in the following generally to (any of) NMR shielding, ESR (EPR) g-tensor. Cases where this statement is not true should be recognizable from the context.

The available subkeys are (abbreviations in brackets)

```
| CLGEPR
|   Nuclei (NUCL) NUOPTIONS
|   Ghosts (GHOST)
|   CalcVirtDia (CALCV)
|   NoParamagneticShielding (NOPARA)
|   Output (OUTP)
|   MixOccupations / NoMixOccupations (MIXOCC / NOMIXOCC)
|   MagnSusceptibility (MAGNSUSC)
|   EprGTensor (EPRGT)
| End
```

"Output", "Ghosts", and "EprGTensor" are block type subkeys. "Output" and "EprGTensor" have a number of possible arguments of their own, see below. A disabled subkey is "Tape". An undocumented subkey is "TestIt" (used occasionally for debugging).

NUCLEI

The optional subkey NUCLEI (NUCL) has as arguments either "ALL", "NONE", or some integer numbers. The default is "ALL", unless EPRGT is specified in which case the default is "NONE". This subkey specifies for which nuclei the NMR shielding is calculated. The input line

```
| NUCLEI ALL
```

keeps the NMR default; in this case, the shielding is calculated at all nuclei in the molecule. The keyword NONE specifies that no nuclei are desired. Obviously, (at least) one "ghost" has to be chosen in this case, or EPRGT or MAGNSUSC have to be specified; the program aborts otherwise (cf. subkey Ghosts). Alternatively, the numbers of the desired nuclei can be specified. Example:

```
| NUCLEI 2 1
```

The numbers refer to the internal numbering of the nuclei as it appears somewhere early in the general ADF output. The internal numbering is not necessarily the same as the input ordering. Note that the number of nuclei has not a very significant consequence for the total CPU time since the CPU intensive parts of the NMR calculations are mostly independent of it. However, the length of the output file does depend on the number of NMR nuclei.

GHOSTS

The subkey GHOSTS is a block type subkey. The format is:

```
| Ghosts
|   xx1 yy1 zz1
```

```

|   xx2 yy2 zz2
|   .....
| SubEnd

```

With this key, the user can specify ANY point(s) within the molecule at which the shielding is to be calculated (whatever the physical meaning of this shielding is). One can think of those points as neutrons within the molecule. There is a publication by P. Schleyer et al. using a similar feature

(J. Am. Chem. Soc. **118**, 6317, 1996). They call it NICS, Nucleus-Independent Chemical Shift.

```

| xx1 yy1 zz1

```

real numbers that specify the Cartesian coordinates of "ghost" 1, etc.

The coordinates have to be specified in the same units as any other input (ADF subkey Units). That is, you use Angstrom for the ghosts if you did so for the atomic coordinates, or bohr otherwise. The same set of coordinates has to be specified as "point charges with charge zero" using the key EFIELD. This is necessary in order to allow the appropriate distribution of integration points around the ghosts.

E.g., if you want to have two "ghosts" with the coordinates xx1 yy1 zz1 and xx2 yy2 zz2 then you **must** also have in the input the key EFIELD as follows

```

| EFIELD
|   xx1 yy1 zz1   0.0
|   xx2 yy2 zz2   0.0
| END

```

(the last number is the charge at these coordinates - zero).

Eventually, this step should be programmed internally but for now the procedure outlined above works. No check is done to verify whether those "point charges" are taken care of or not, but their omission leads to unpredictable results.

Only Cartesian coordinates are possible for ghosts, even if the atoms were originally specified using internal coordinates. This shouldn't be a problem, though (e.g., one could start an ADF run of the molecule of interest, and get very soon the Cartesian coordinates of the atoms in the output. This run would then be aborted, and restarted with the ghosts specified as desired.) The ghosts are numbered in the output as NNUC+1, NNUC+2 ... where NNUC is the total number of nuclei in this molecule. Default: no ghosts.

CALCVIRTDIA

The subkey CALCVIRTDIA invokes the calculation of the diamagnetic integrals for a specified number of virtual orbitals. The argument is an integer number, NVDIA. NVDIA is the number of virtual orbitals for which those integrals are desired. Example:

```

| CalcVirtDia 12

```

results in the calculation (and output) of these integrals for the first 12 (lowest energy) virtuals of each spin. Default: NVDIA=0. The program aborts if the specified number is bigger than the total number of virtuals that this molecule has for the given basis set.

NOPARA

The subkey NOPARA suppresses the calculation of the paramagnetic shielding completely. This can be useful if single atoms are considered. Then, the paramagnetic shielding must vanish by symmetry. On the

other hand, the program would crash in the calculation of the paramagnetic shielding of atoms since the occupation numbers are not all equal (see also the key MixOcc). The key doesn't have any argument. The default is obviously that this feature be not used. A warning is given to the user if this switch is set for molecules.

OUTPUT

The subkey OUTPUT is a block type subkey. Possible arguments are (abbreviations in brackets)

```

Output
  Size
  EprSize
  MSuscSize
  Criteria (CRIT)
  Format
  PrincipalAxisRep (PRINCIP)
  AntisymmetricTensors (ANTISY)
  PrintMat (PrintM)
  SumUpOccVir (SUMUP)
  RelativisticShielding (RELATIVIST)
SubEnd

```

Many of those are not necessary for most cases. All of the keywords in detail:

Size

regulates the amount of information about the NMR shielding that is printed (and sometimes calculated). Possible arguments are (abbreviations in brackets):

```

MINIMAL (MIN)
SMALL (SMAL)
LARGE (LARG)
DEBUG
BIGDEBUG (BIGDEB)

```

Default is "small".

If MIN is specified, then only the isotropic shielding constant and its contributions as well as the total shielding tensor are printed for the different nuclei.

For SMALL, the default, also the dia- and paramagnetic shielding tensors are shown in the output.

If LARGE is specified, then the different contributions to the paramagnetic shielding tensor are printed according to our GIAO formulation. Furthermore, the dia- and paramagnetic shieldings are analyzed with respect to the occupied and virtual orbitals of the (unperturbed) molecule. With other words, the different matrix elements of these shielding tensors are printed.

Finally, DEBUG and BIGDEBUG invokes the calculation and output of various test parameters etc. that are of interest for debugging only, and can be ignored otherwise (see the code for details).

EprSize and MSuscSize

completely analogous to Size for EPR g-tensors, respectively.

Default: same as "Size", if EPR is input specified (the keys are obsolete otherwise).

Crit

The argument is a real number PSMAL. Only those matrix elements M of the paramagnetic shielding are printed for which the absolute of M is greater then [psmal times the isotropic shielding] at this nucleus.

E.g.,

```

| CRIT 0.1

```

prints only matrix elements if they are at least 10% of the isotropic shielding in magnitude.

Default: psmal=0.00001.

The keyword is not relevant if the SIZE is specified as "MINIMAL" or "SMALL". It is ignored in this case.

Format

This keyword refers to the format that is used for some of the output - d format or f format. Possible arguments are F (the default) or D. The f format is easier to read and thus the default. The D format gives a few more digits, and might sometimes be useful for this reason.

PrincipalAxisRep

The total shielding tensor is being symmetrized and transformed into its principal axis representation. The given keyword regulates the step. It has the two possible arguments PRINT (default) and NOPRINT. NOPRINT suppresses the diagonalization.

AntisymmetricTensors

This key allows the printing of the antisymmetric part of the total shielding tensor. The antisymmetric part is usually not accessible for experiment. Thus, the default is the argument NOPRINT. The other recognized argument is PRINT.

PrintMat

This keyword invokes the printing of the respective matrices. Arguments: (any of) B1 SF COEF. The default is no printing.

B1 refers to the first order B coefficients of the frozen core approximation. It is irrelevant in all-electron calculations.

COEF refers to the first order coefficients of the magnetic wave functions, i.e., the S-matrix (occupied-occupied) and the U-matrix (occupied-virtual).

SF finally means the constituting matrices of the U coefficients: the first order S-matrix (occupied-virtual) and the first order F-matrix (occupied-virtual). This keyword is probably of interest for debugging only.

SumUpOccVir

invokes the additional output of the occ-vir matrix elements, summed up per occupied MO over all the virtuals. This is only relevant for "Large" Output, and ignored otherwise. Default: turned off. No arguments are required. This keyword is sometimes quite useful since it gives a quick overview as to which occupied orbitals have a significant contribution to the paramagnetic shielding. Also, it doesn't cost any CPU time, and not much space in the output file.

RelativisticShielding

Print key for the output of the direct (scalar) relativistic shielding contributions. The arguments are PRINT, the default for relativistic calculations, and NOPRINT, which is, obviously, the default for non-relativistic calculations.

MixOccupations / NoMixOccupations

The subkeys MixOccupations / NoMixOccupations (MIXOCC / NOMIXOCC). Our GIAO formulation of the shielding does not allow for broken (non-integer) occupation numbers that are used in ADF for partially filled, degenerate orbitals. The reason is that we use the "S-Matrix" for the occupied-occupied contributions to the paramagnetic shielding which in turn avoids degenerate perturbation theory. Hence, the program aborts when these broken occupation numbers occur (key NoMixOcc, the default). This default can be overwritten, key MixOcc, in which case we use the occupied-occupied U-matrix instead of the S-matrix, but only for combinations of occupied MOs that do not have equal occupation numbers. This is mostly an experimental feature.

MAGNETIC SUSCEPTIBILITY

Subkey MagnSusceptibility

The calculation of the magnetic susceptibility (more precisely, magnetizability) is not functional at the moment. Hence, no more documentation at this point!

EPRGTENSOR

Subkey *EprGTensor*

invokes the calculation of the EPR (or ESR) g-tensor and its contributions.

The key is a block type subkey, and (most of) the arguments are not really required. Those arguments are (abbreviations in brackets):

```
| EprGTensor
|   EPRXC {X-ALP NONE}
|   XCCutOff (XCCUT)
|   KinCorrection (KINCORR)
|   Skip
|   SubEnd
```

Disabled arguments include *TestPot* and *TestPar*. The other keywords in detail:

EPRXC

regulates which XC potential to use for the exchange in the effective ESR potential (Georg Schreckenbach's Ph.D. thesis, p. 124).

Note that this does **not** influence the XC potential that is used elsewhere in the calculation.

Currently implemented are the following two arguments:

X-Alpha (X-ALP) and NONE

In addition, it is possible to specify the X-alpha parameter to be used as a non-negative real number after the keyword. The default for the whole keyword is:

```
| EPRXC X-Alpha 0.7
```

XCCutOff

A numerical cutoff to be used with the EPR exchange potential. It avoids numerical instabilities, but has no influence on the results otherwise. The default turns out to be sensitive, and should not be changed (it depends on the number of integration points and the number of electrons). Alternatively, one can specify a real number CUTFF: the EPR exchange potential is set to zero if its value is smaller than 10^{**}-CUTFF in the given integration point.

KinCorrection

This keywords includes or excludes the use of the isotropic kinetic energy to the g-tensor. Arguments are ON (default) and OFF.

Skip

This key allows one to exclude various contributions from the calculation of the g-tensor that might be useful to see their respective importance (see, e.g., the EPR paper, tables 7 and 8).

Possible arguments are (any of)

```
| NucDerivative (NUCDER)
| CoreDerivative (COREDER)
| ValDerivative (VALDER)
| XCDerivative (XCDER)
| KinCorrection (KINCORR)
```

Specifying these arguments excludes the derivative of the nuclear, core electronic, valence electronic, or exchange potentials (arguments NUCDER, COREDER, VALDER, or XCDER, respectively) from the effective potential to be used for the g-tensor. Specifying XCDER here is equivalent to the argument NONE of the keyword EPRXC. Finally, Skip KinCorr excludes the kinetic energy correction from the g-tensor, and is equivalent to the keyword KinCorr OFF.

SICOEP

This option [10] is not yet available. It will become available in the future.

Additionally, the program will accept an (experimental) block key:

SICOEP

The only meaningful subkey at this time is:

```
| SICOEP  
|   READV file-with-OEP-potential  
| END
```

The maximum memory usage may be specified with MAXMEMORYUSAGE key, see ADF documentation for the description. Memory requirements of the NMR/EPR calculation can be reduced by decreasing numerical integration vector length (see VECTORLENGTH) in the preceding SCF calculation.

SD

This documentation is based on an original version by S.K. Wolff, who is the main author of the present implementation [11]. The implementation heavily uses earlier work from the Calgary group [12,13].

Introduction

The Second Derivatives (SD) program analytically calculates the Hessian matrix, from which the infrared IR frequencies and intensities can be obtained.

The SD program requires that a previous ADF Kohn-Sham calculation be done on the molecule of interest and that the corresponding TAPE21 be saved.

Currently, the SD program is about 3 to 4 times faster than the older numerical second derivatives (in ADF) for molecules that lack symmetry. If symmetry can be taken advantage of, then SD is only slightly faster than the older numerical code. Like the older numerical code, SD can be run in parallel.

Linear-scaling techniques have been introduced into the SD code, but as yet these have not been thoroughly tested.

The SD program can calculate gradients, dipole derivatives, vibrational frequencies and IR intensities. The first half of the output from the SD program is unique to SD, but the second half is identical to the output from a frequency run using the older numerical second derivatives code in ADF.

Unfortunately, SD still only works with the Xa xc-potential, although work is in progress to extend this to general gradient approximations (GGAs).

The requirements on the numerical integration seem to be less strict than for the numerical difference implementation in ADF itself. However, for heavier nuclei, it seems advisable to choose a higher accsph value than general accint value (typically 2 higher). Please check the INTEGRATION keyword in the ADF User's Guide for details on how to specify this.

The SD implementation can be combined with the ZORA treatment for relativistic effects. Some examples are discussed in the Examples document.

NOTE: At the moment the integration accuracy for SD is independent of that of ADF and should be input in the run file, prior to the SD block. To do this use the ADF key-word **integration N** where N is the integration accuracy required (default 4.0). See the SD example runs for more detail.

Input options for second derivatives program

So far the input options are (words between square brackets should be optional):

```
SD
  CALC      { HESSIAN GRADIENT DIPOLE ALL }
  PRINT     { EIGS U1 PARTS }
  MAX_CPKS_ITERATIONS  N
  CHECK_CPKS_FROM_ITERATION  N
  U1_ACCURACY  CRIT
  SAVE_FINAL_U1_TO_FILE
  IO_U1_TO_FILE
```

```

|   DEBUG { FIT HESSIAN DENSITIES NUMBERS SYMMETRY ALL }
|   END

```

CALC

The subkey CALC determines what is actually calculated. By default, nothing is calculated. The subkey has the following options

HESSIAN

Calculate the hessian

GRADIENT

Calculate the gradient

DIPOLE

Calculate the dipole and derivative

ALL

Calculate all of the above

PRINT

The subkey PRINT determines what is printed. By default nothing is printed. The subkey has the following options

EIGS

Print eigenvectors

U1

Print u1 matrix

PARTS

Print major contributions to the Hessian (e.g. kinetic, potential, wenergy)

OPTIONS

FIT

Print out fit coefficients and derivatives and other information relating to these quantities.

HESSIAN

Print out the Hessian.

DENSITIES

Print out information about the various densities used throughout the program. In particular differences between fit and exact densities are printed out.

NUMBERS

Print out the values of the numerous variables used throughout the program in a neat tabular form. For example, the number of basis functions, the number of fit functions, etc.

SYMMETRY

Print out information relating to symmetry. For example the symmetry operators of the symmetry group.

ALL

Print out all of the above plus more.

MAX_CPKS_ITERATIONS

The subkey MAX_CPKS_ITERATIONS determines the maximum number of iterations in the coupled-perturbed Kohn-Sham (CPKS) equations to be performed before stopping. The default value is 10. The iterations will

halt before this value if the convergence criteria (see U1_ACCURACY below) is reached.

N

Maximum number of iterations in CPKS cycle

CHECK_CPKS_FROM_ITERATION

The subkey CHECK_CPKS_FROM_ITERATION determines at which iteration the coupled-perturbed Kohn-Sham (CPKS) should be checked to see if it has converged. This key-word is not likely to effect performance. The default value is 1.

N

Iteration to start checking CPKS for convergence.

U1_ACCURACY

The subkey U1_ACCURACY determines the convergence criteria for the coupled-perturbed Kohn-Sham (CPKS). Convergence is achieved when the difference between the U1 matrix elements of the current and previous cycle are less than the value CRIT that is input. The default is 0.0001.

CRIT

Convergence criteria for CPKS.

SAVE_FINAL_U1_TO_FILE

The subkey SAVE_FINAL_U1_TO_FILE determines whether the final U1 matrix that results from solving the coupled-perturbed Kohn-Sham (CPKS) equations should be saved to file. This may be of use for future programs in ADF that need the U1 matrix (rather than re-calculating it). [Note that this option does not effect the fact that during the running of the SD program, the whole U1 matrix is stored in memory. To alter this see the key-word IO_U1_TO_FILE.]

IO_U1_TO_FILE

The subkey IO_U1_TO_FILE determines whether the whole U1 matrix that results from solving the coupled-perturbed Kohn-Sham (CPKS) is saved to file. If this key-word is not present then the whole U1 matrix is stored in memory, which can cause problems for large molecules. If this key-word is present, then only the required portion of the U1 matrix is held in memory, while the rest is stored on file.

DEBUG

The subkey DEBUG is for debugging purposes.

NMR

The NMR program was originally written by G.Schreckenbach and later adapted and extended by S.K. Wolff.

NMR: Chemical Shifts

The utility program NMR computes NMR chemical shifts. It has been developed in the Theoretical Chemistry group of the University of Calgary [5,7,14,15].

NMR requires an ASCII input file and a TAPE21 result file from an ADF calculation on the molecule to be analyzed. The ADF result file must be present with name TAPE21 in the directory where you execute NMR.

NMR calculations on systems computed by ADF with Spin Orbit relativistic effects included must have used NOSYM symmetry in the ADF calculation. NMR can also be combined with ADF ZORA calculations. The NMR program reads from TAPE21 the relativistic option that is used in the ADF calculation, and will use the same relativistic option in the NMR calculations.

A few sample runs are contained in the ADF distribution package. See the Examples document.

Input options

The whole input file for NMR uses one block key NMR, with several (optional) sub keys, each having a series of options.

```
NMR
  Out  OutOptions
  Calc CalcOptions
  U1K  U1KOptions
  Nuc  NucOptions
  Analysis AnalysisOptions
  MAXMEMORYUSAGE MemValue
End
```

OUT

The sub key Out controls printed output. Its options specify the details by their (optional) presence. The following OutOptions are recognized (Default ISO):

All
Implies all the other options except for 'ISO', which may be specified in addition.

ISO
Isotropic shielding constants

Tens
Shielding tensors

Eig
Eigenvectors

U1
The U1 matrix

F1

The first order change in the Fock matrix

S1

The first order change in the Overlap matrix

AOP

The paramagnetic AO matrix (= the matrix in the representation of elementary atomic basis functions)

AOD

The diamagnetic AO matrix

AOF

The Fermi-contact AO matrix

REFS

Literature references

INFO

General information

CALC

The sub key Calc controls what is actually calculated.

The following options are available (Default ALL):

All

Implies all of the other options to this key.

Para

The paramagnetic part

Dia

The diamagnetic part

FC

The Fermi-contact part

U1K

The sub key U1K determines which terms are included in the calculation of the U1 matrix (first order changes in MO coefficients).

Options (Default none):

Best

The best (recommended) options for each relativistic option are included for this sub key.

All

Implies all the other options to this key.

MV

The mass-velocity term

Dar

The Darwin term

ZMAN

The Spin-Zeeman term.

NUC

The (sub) key Nuc determines for which nuclei the chemical shifts are computed. If this (sub) key is omitted from the NMR block, the calculations are carried out for all nuclei. Else you may use this options by simply typing Nuc in the NMR block (without any further data); this means: for no nuclei at all. Alternatively you may type the index of the atom(s) you want to see analyzed.

Default 'all' (i.e. as for omitting this sub key).

ANALYSIS

The sub key Analysis controls the MO analysis. After the word (sub key) Analysis you type an integer, which then specifies that the first so many MOs are to be analyzed.

Default no Analysis.

MAXMEMORYUSAGE

The subkey MAXMEMORYUSAGE regulates the total amount of dynamic memory for the allocation scheme in the ADF package. Note that with the introduction of Fortran90 automatic arrays and allocatable arrays, the ADF allocation scheme will disappear and, moreover, does not completely determine how much memory a calculation may use.

MemValue is a value in megabyte. Default 80.

DISPER

The DISPER program was originally written by V.Osinga [16]. The original documentation was written by S.J.A. van Gisbergen.

Disper: Dispersion Coefficients

The program DISPER computes Van der Waals dispersion coefficients up to C_{10} for two arbitrary closed-shell molecules. ADF itself can already compute some C_6 and C_8 coefficients between two identical closed-shell molecules. These coefficients describe the long-range dispersion interaction between two molecules. It requires previous ADF-TDDFT calculations for the polarizability tensors at imaginary frequencies for the two interacting molecules. Each such ADF calculation produces a file TENSOR (if suitable input for ADF is given). The TENSOR files must be renamed tensorA and tensorB, respectively and must be present as local files for DISPER. The DISPER program takes no other input and prints a list of dispersion coefficients.

A schematic example, taken from the set of sample runs, for the usage of DISPER is the following:

Step1: run ADF for, say, the HF molecule. In the input file you specify the RESPONSE data block:

```
RESPONSE
  MaxWaals 8      ! Compute dispersion coefficients up to C8
  ALLTENSOR      ! This option must be specified in the ADF calc for a subsequent DISPER run
  ALLCOMPONENTS ! Must also be specified for DISPER
End
```

At the end of the run, copy the local file 'TENSOR' to a file 'tensorA'. For simplicity, we will now compute the dispersion coefficients between two HF molecules. Therefore, copy 'tensorA' to 'tensorB'.

Now run DISPER (without any other input). It will look for the local files 'tensorA' and 'tensorB' and compute corresponding dispersion coefficients to print them on standard output.

The output might look something like this:

```
DISPER 2000.02 RunTime: Apr04-2001 14:14:13
***** C-COEFFICIENTS *****
n  LA KA LB KB L coefficient(Y) coefficient(P)
6  0  0  0  0  0  28.29432373  28.29432373
6  2  0  0  0  2  7.487547697  3.348533127
8  0  0  0  0  0  416.1888455  416.1888455
8  0  0  2  0  2  0.4323024202E-05  0.1933315197E-05
8  2  0  0  0  2  402.3556946  179.9389368
8  2  0  2  0  4  0.4238960180E-05
8  4  0  0  0  4  -36.67895539  -12.22631846
8  4  0  2  0  6  -0.2000286301E-05
```

The n-value in the first column refers to the long-range radial interaction. The case n=6 refers to the usual dipole-dipole type interaction related to a $1/R^6$ dependence in the dispersion energy. The n=7 case relates to a dipole-quadrupole polarizability on one system and a dipole-dipole polarizability on the other (this is not symmetric!). The n=8 term may contain contributions from a quadrupole-quadrupole polarizability on one system in combination with a dipole-dipole polarizability on the other as well as contributions from two dipole-quadrupole polarizabilities.

Terms which are zero by symmetry are not printed. In the example above, this is the case for all $n=7$ terms, because the systems (apparently) are too symmetric to have a nonzero dipole-quadrupole polarizability. The best known and most important coefficients are the isotropic ones, determining the purely radial dependence of the dispersion energy. They are characterized by the quantum numbers: 6 0 0 0 0 0 (or 8 0 0 0 0 0 etc.) Other combinations of quantum numbers refer to different types of angular dependence. The complete set determines the dispersion energy for arbitrary orientations between the two subsystems A and B.

The complete expressions are rather involved and lengthy. We refer the interested reader to the paper [16] which contains a complete description of the meaning of the various parts of the output, as well as references to the earlier literature which contain the mathematical derivations. In particular, a useful review, which was at the basis of the ADF implementation, is given in [17]. Of particular significance is Eq.(8) of the JCP paper mentioned above, as it defines the meaning of the calculated coefficients $C_n^{(L_A, K_A, L_B, K_B, L)}$ as printed above.

For highly symmetric systems, a different convention is sometimes employed. It is based on Legendre polynomials (hence the 'P' in the final column) instead of on the spherical harmonics (the 'Y' in the column before the last). The 'P' coefficients are defined only for those coefficients that are nonzero in highly symmetric systems and never contain additional information with respect to the 'Y' coefficients. They are defined [Eq. (14) in the mentioned J. Chem. Phys. paper] in terms of the 'Y' coefficients by:

$$C_n^L = (-1)^L C_n^{L,0,0,L} / \sqrt{(2L+1)}$$

Because the quality of the dispersion coefficients is determined by the quality of the polarizabilities that are the input for DISPER, it is important to get good polarizabilities from ADF. For that it is important, in the case of small systems, to use an asymptotically correct xc potential (several choices are available in ADF, such as SAOP or GRAC) and a basis set containing diffuse functions. We refer to the ADF User's Guide for details.

References

- [1] Autschbach, J.; Ziegler, T. *J Chem Phys* 2000, 113, 936.
- [2] Autschbach, J.A.; Ziegler, T. *J Chem Phys* 2000, 113, 9410.
- [3] Autschbach, J.A.; Ziegler, T. *J Am Chem Soc* 2001, 123, 3341.
- [4] Autschbach, J.A.; Ziegler, T. *J Am Chem Soc* 2001, 123, 5320.
- [5] Schreckenbach, G.; Ziegler, T. *J Phys Chem* 1995, 99, 606.
- [6] Schreckenbach, G.; Ziegler, T. *Int J Quantum Chem* 1996, 60, 753.
- [7] Schreckenbach, G.; Ziegler, T. *Int J Quantum Chem* 1997, 61, 899.
- [8] Schreckenbach, G.; Ziegler, T. *J Phys Chem A* 1997, 101, 3388.
- [9] Patchkovskii, S.; Ziegler, T. *J Phys Chem* 2001, A105, 5490.
- [10] Patchkovskii, S.; Autschbach, J.A.; Ziegler, T. *J Chem Phys* 2001, 115, 26.
- [11] Wolff, S.K. unpublished 2000.
- [12] Bérces, A.; Dickson, R.M.; Fan, L.; Jacobsen, H.; Swerhone, D.; Ziegler, T. *Comput Phys Commun* 1997, 100, 247.
- [13] Jacobsen, H.; Bérces, A.; Swerhone, D.P.; Ziegler, T. *Comput Phys Commun* 1997, 100, 263.
- [14] Wolff, S.K.; Ziegler, T. *J Chem Phys* 1998, 109, 895.
- [15] Wolff, S.K.; Ziegler, T.; van Lenthe, E.; Baerends, E.J. *J Chem Phys* 1999, 110, 7689.
- [16] Osinga, V.P.; van Gisbergen, S.J.A.; Snijders, J.G.; Baerends, E.J. *J Chem Phys* 1997, 106, 5091.
- [17] Avoird, A.v.d.; Wormer, P.E.S.; Mulder, F.; Berns, R.M. *Topics in Current Chemistry* 1980, 93, 1.

ADF Property Programs: Index

A-tensor [\[1\]](#)
C10 coefficient [\[1\]](#)
C6 coefficient [\[1\]](#)
C8 coefficient [\[1\]](#)
CLGEPR [\[1\]](#)
CPL Diamagnetic orbital term [\[1\]](#)
CPL Fermi-Contact term [\[1\]](#)
CPL Paramagnetic orbital term [\[1\]](#)
CPL Spin-Dipole term [\[1\]](#)
EPR [\[1\]](#) [\[2\]](#)
EPRGTENSOR [\[1\]](#)
ESR [\[1\]](#) [\[2\]](#)
GIAO formalism [\[1\]](#)
Hessian [\[1\]](#)
MAGNSUSCEPTIBILITY [\[1\]](#)
NICS [\[1\]](#)
NMR chemical shift [\[1\]](#)
NMR shielding tensor [\[1\]](#)
NMR spin-spin couplings [\[1\]](#)
NMRCOUPLING [\[1\]](#) [\[2\]](#)
NSSCC [\[1\]](#)
Q-tensor [\[1\]](#)
SICOEP [\[1\]](#)
analytical second derivatives [\[1\]](#)
dispersion coefficients [\[1\]](#)
g-tensor [\[1\]](#) [\[2\]](#)
infrared frequencies [\[1\]](#)
infrared intensities [\[1\]](#)
long range dispersion interaction [\[1\]](#)
magnetizability [\[1\]](#)
nuclear spin-spin coupling constant [\[1\]](#)
nuclear-independent chemical shift [\[1\]](#)
reduced spin-spin coupling constant [\[1\]](#)
spin-spin coupling constant [\[1\]](#)
van der Waals [\[1\]](#)