

Table 9.1 – continued from previous page

		g_DSO	AD	N	N	
		g_PSO	AD	N	N	
		g_Tot	AD	N	N	
		g_iso	D	N	N	
		Delta_g	AD	N	N	
		Delta_g_iso	D	N	N	
		orientation	AD	N	N	
		Method	S	N	N	
		Level	S	N	N	
		Mult	I	N	N	
		State	I	N	N	
		Irrep	I	N	N	
MDCI_LED	N	numOfFragments	I	N	N	The number of fragments
		sumNonDisStrong	D	N	N	Sum of non dispersive correlation strong pairs
		sumNonDisWeak	D	N	N	Sum of non dispersive correlation weak pairs
		electrostaticRef	AD	N	N	Electrostatics reference
		exchangeRef	AD	N	N	Exchange reference
		dispContr	AD	N	N	Strong pair dispersion contribution
		dispWeak	AD	N	N	Weak pair dispersion contribution
		refInt	AD	N	N	Interaction reference
		corrInt	AD	N	N	Interaction correlation
		totInt	AD	N	N	Interaction total
Natural_Orbitals	N	NEL	I	N	N	
		NSOMO	I	N	N	
		NDOMO	I	N	N	
		NVMO	I	N	N	
		NNatoOrbs	I	N	N	
		OccUno	AD	N	N	Occupation of natural orbitals
		OccUnso	AD	N	N	Occupation of spin natural orbitals
Calculation_Timings	N	GTOINT	D	Y	N	
		STOINT	D	Y	N	
		SCF	D	Y	N	
		SCFGRAD	D	Y	N	
		GSTEP	D	Y	N	
		PLOT	D	Y	N	
		MOM	D	Y	N	
		MP2	D	Y	N	
		CPSCF	D	Y	N	
		LOC	D	Y	N	
		CIS	D	Y	N	
		MRCI	D	Y	N	
		REL	D	Y	N	
		SOC	D	Y	N	
		EPRNMR	D	Y	N	
		CASSCF	D	Y	N	
		MDCI	D	Y	N	
		CHELPG	D	Y	N	
		ROCIS	D	Y	N	
		MRCC	D	Y	N	
		FCI	D	Y	N	
		MD	D	Y	N	
		FREQ	D	Y	N	
		FREQ_NUM	D	Y	N	
		COMPOUND	D	Y	N	
		CIPSI	D	Y	N	
		AUTO CI	D	Y	N	
		ESD	D	Y	N	
		DLPNOCC	D	Y	N	
		MCRPA	D	Y	N	
		NEB	D	Y	N	
		CONFSC	D	Y	N	
		ANMR	D	Y	N	
		MM	D	Y	N	
		MMGRAD	D	Y	N	
		XTB	D	Y	N	

continues on next page

Table 9.1 – continued from previous page

SCF_Timings	N	XTBGRAD	D	Y	N
		LFT	D	Y	N
		EXT	D	Y	N
		ONIOM_S	D	Y	N
		ONIOM_M	D	Y	N
		ONIOM_L	D	Y	N
		PROPINT	D	Y	N
		RESPONSE	D	Y	N
		PROP	D	Y	N
		CASRESP	D	Y	N
		MP2RESP	D	Y	N
		COSMORS	D	Y	N
		AUTOCIRESP	D	Y	N
		GOAT	D	Y	N
		CISRESP	D	Y	N
		EDA	D	Y	N
		SUM	D	Y	N
		TOTAL	D	Y	N
		PREP	D	Y	N
		GUESS	D	Y	N
		FOCK	D	Y	N
		DENS	D	Y	N
		DIAG	D	Y	N
		ETOT	D	Y	N
		POP	D	Y	N
		TRAFO	D	Y	N
		ORTHO	D	Y	N
		DIIS	D	Y	N
		SOSCF	D	Y	N
		NR	D	Y	N
		J	D	Y	N
		X	D	Y	N
		XC	D	Y	N
		GRID	D	Y	N
		FSYMMETR	D	Y	N
		PSYMMETR	D	Y	N
		FOCKSTART	D	Y	N
		SOLV	D	Y	N
		SOLV_INIT	D	Y	N
		SOLV_FINAL	D	Y	N
		INT_PREP	D	Y	N
		INT_PRESCREEN	D	Y	N
		INT_BF	D	Y	N
		INT_DENS	D	Y	N
		INT_DENSIO	D	Y	N
		INT_FUNC	D	Y	N
		INT_POT	D	Y	N
		INT_POTIO	D	Y	N
		INT_SUM	D	Y	N
		SPLITRIJ	D	Y	N
		COSX	D	Y	N
		COSX1C	D	Y	N
		RIJK	D	Y	N
		STABANA	D	Y	N
		DFET	D	Y	N
		ADFT_GUESS	D	Y	N
		ADFT_JDENS	D	Y	N
		ADFT_JMAT	D	Y	N
		ADFT_XDENS	D	Y	N
		ADFT_XMAT	D	Y	N
		ADFT_TMAT	D	Y	N
		ADFT_EQNSOLV	D	Y	N
		ECRISM	D	Y	N
		ECRISM_QESP	D	Y	N
		ECRISM_TOOL	D	Y	N
		ECRISM_FOCK	D	Y	N
		FROZCORE	D	Y	N
Solvation_Details	N	Solvent	S	N	N
		Epsilon	D	N	N
		Refrac	D	N	N
		RSolv	D	N	N

continues on next page

Table 9.1 – continued from previous page

Geometry	N	SurfaceType	S	N	N	
		CPCMDielEnergy	D	N	N	
		NPoints	I	N	N	
		SurfaceArea	D	N	N	
		NAtoms	I	N	N	
		NCorelessECP	I	N	N	
		NGhostAtoms	I	N	N	
		Coordinates	CC	N	N	Bohr
		ATNO	AI	N	N	
		coreless_ECP	AI	Y	N	Non-zero for cECPs
		ghostAtoms	AI	Y	N	Non-zero for ghost atoms
		fragments	AI	Y	N	Fragment ID

¹ The name in brackets (if given) is the base property name, which is used for JSON export.

² Are multiple properties with the same base property name allowed. If so, the JSON value is a list of properties.

³ Type, abbreviated as follows: D: Double, C: Complex, I: Integer, B: Boolean, S: String, AD: ArrayOfDoubles, AC: ArrayOfComplex, AI: ArrayOfIntegers, AB: ArrayOfBooleans, CC: Coordinates,

⁴ Whether the component is optional (only stored if applicable).

⁵ Whether there are multiple entries of the component. In JSON, these will be stored as a list.

ARCHITECTURE OF ORCA

10.1 ORCA Structure

10.1.1 Program Components

The program system ORCA consists of several separate programs that call each other during a run. The following basic modules are included in this release (listed in alphabetical order):

orca	Main input+driver program
orca_ailft	Ab Initio Ligand Field Theory
orca_autoci	CI type program using the automated generation environment (ORCA-AGE)
orca_casscf	Main program for CASSCF driver
orca_casscfresp	CASSCF static linear response
orca_cclib	Precalculation of one particle coupling coefficients for ACCCI
orca_chelpg	Calculation of different kinds of ESP charges
orca_ciprep	Preparation of data for MRCI calculations (frozen core matrices and the like)
orca_cipsi	Iterative Configuration Expansion Configuration Interaction (ICE-CI)
orca_cis	Excited states via CIS and TD-DFT
orca_cisresp	CIS response calculation
orca_compound	General driver of the ORCA compound language
orca_crystalprep	Set up an embedding calculation from a crystal structure file
orca_eca	Auxiliary program for solving magnetic model Hamiltonians
orca_eda	Energy decomposition analysis module
orca_esd	Excited state dynamics program for calculating vibronic spectra, resonance Raman, ...
orca_grad	General gradient module
orca_guess	Generation of an initial guess for SCF and CASSCF
orca_leanscf	Self consistent field program for HF and DFT
orca_lft	Ligand Field Theory
orca_loc	Calculation of localized molecular orbitals
orca_mcrpa	CASSCF linear response for excited states
orca_md	Molecular dynamics program
orca_mdci	Matrix driven correlation program: CI, CEPA, CPF, QCISD, CCSD(T)
orca_mm	Classical forcefield components
orca_mp2	MP2 program (conventional, direct and RI)
orca_mrci	MRCI and MRPT calculations (individually selecting)
orca_ndoint	Calculates semiempirical integrals and gradients
orca_nmrspectrum	Simulates NMR spectra from calculated NMR parameters
orca_numfreq	Numerical Hessian computation
orca_pc	Addition of point charge terms to the one-electron matrix
orca_plot	Generation of orbital and density plots
orca_pop	External program for population analysis on a given density
orca_prop	Calculation of molecular properties
orca_propint	Calculation of property integrals

continues on next page

Table 10.1 – continued from previous page

orca_rel	(Quasi) Relativistic corrections
orca_rocis	Excited states via the ROCIS method
orca_scfgrad	Analytic derivatives of SCF energies (HF and DFT)
orca_scfresp	self consistent field response
orca_startup	Calculation of startup data before each single point
orca_vpot	Calculation of the electrostatic potential on a given molecular surface
orca_vpt2	VPT2 analysis

10.2 The Architecture of ORCA

In this chapter we describe the broad features of ORCA's internal structure. This knowledge is not necessary to carry out calculations with ORCA. It may, however, help to understand which modules are being called in which order and why this is happening in the sequence it does.

10.2.1 The structure of the ORCA source code

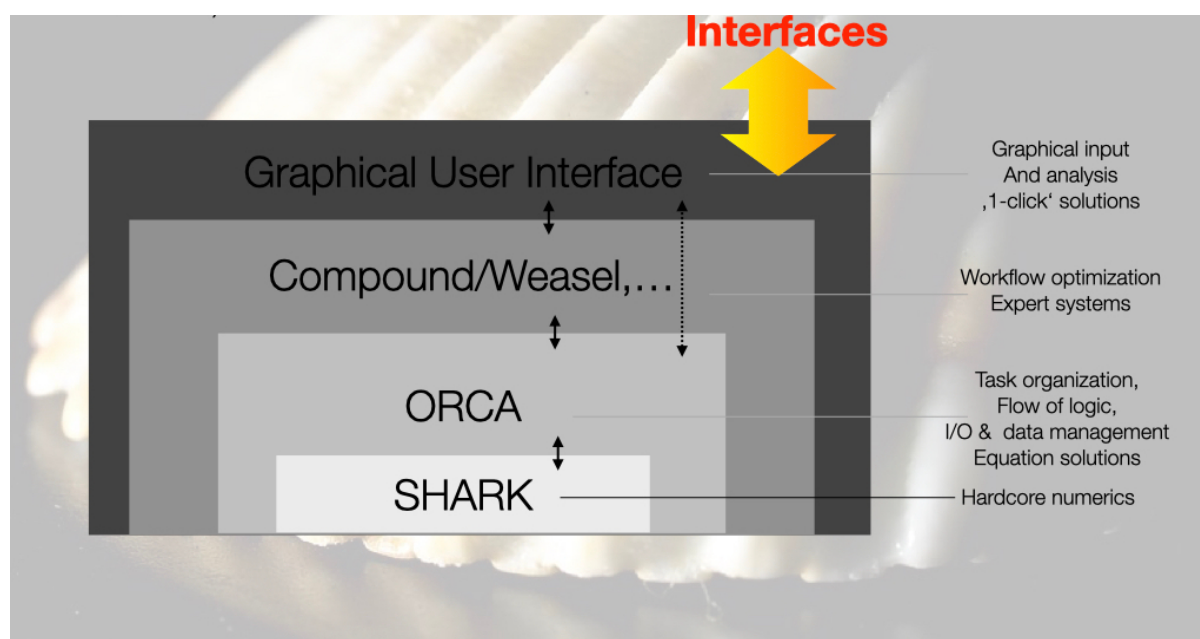
The source code of ORCA is broadly structured in three parts:

- 1) The main program.
- 2) The code for the "ORCA tool library".
- 3) The code for individual modules.

The code in the ORCA tool library is being compiled into one library file that subsequently is linked with all ORCA modules. The code for the individual modules can make use of everything that is in the library, but the modules are not supposed to link to or use any code of the main program or any other module. This way of structuring the code ensures that the modules remain maintainable and that there are no complex and unwanted interdependencies that would make it eventually impossible to exchange modules for new code.

10.2.2 The shell structure of ORCA

The whole organization of the code and the information flow can be thought of as consisting of a shell structure where a shell is a layer of software with well-defined functions and a well-defined interface to the shells above and below it.



The lowest layer of the shell consists of the SHARK integral package. This is the “motor” of the program that takes care of most compute intensive tasks and essentially all parts that involve integrals over basis functions. This amounts to calculating all one- and two-electron integrals, forming Fock and Fock-like matrices and performing integral transformations using all kinds of basis sets and kernels. Most of the SHARK code is independent of the remaining ORCA source code infrastructure.

The layer above SHARK is the actual ORCA source code. ORCA interacts with the user through the interface and it orchestrates the entire workflow of the calculation. At this point in time, it also performs compute intensive tasks like forming diagonalizing and handling Fock-operators, solving large linear equation systems, forming sigma vectors, densities etc in correlated calculations and similar tasks. The ORCA source code interacts with SHARK through a small set of DRIVERS. The drivers feature genuine ORCA data structures as well as SHARK infrastructure. The DRIVERS call the genuine SHARK functions (ORCA calls SHARK directly only in a few places) and, most importantly, the DRIVERS take care of finding their way through the approximation jungle. By this we mean they handle the necessary basis sets, auxiliary basis sets, grids, integral approximations, solvation, relativity, point charges etc.

Concentrating these important and highly error prone tasks in a number of well documented routines provides a highly transparent way of identifying and properly maintaining the functionality that is at the heart of the program.

Above the ORCA software layer there are tools that orchestrate workflows. Workflows typically consisting of a number of interdependent computational steps that are later combined into a single meaningful chemical result. For example, one may optimize the geometry with a DFT functional and calculate zero-point and thermodynamic corrections at the same level and follow it up with a single point calculation at the coupled cluster level that may or may not feature another correction for core correlation at the MP2 level or complete basis set extrapolation. Such tasks and many others, like running series of calculations on a test-sets of molecules or permuting calculation options like functionals or basis sets on a given test system can be addressed with workflow tools. Inside ORCA, there is the very powerful compound scripting language to achieve and organize such tasks. On the commercial branch, FAccTs develops the Weasel tool that is a powerful and highly reliable workflow organizer.

At the final layer, one might envision a graphical user interface (GUI) that helps building molecules and facilitates running calculations and analyzing results. At this point in time, ORCA does not have a dedicated GUI. There are many free and commercial solutions that interface to ORCA. This form of interfacing is facilitated by the `orca_2json` interface and the property file that ORCA produces. We hope that the transparency of this interface motivates GUI developers to provide ever improved GUIs that interface ORCA. We do not exclude the possibility that ORCA will feature its own GUI sometime down the road.

10.2.3 The master/slave concept and the calling sequence

Within ORCA, the code follows a “master/slave” concept in which the main program is allowed to know everything about every module while the modules are not allowed to know anything about the main program. Thus, the main program is the piece of software that orchestrates the entire ORCA run and the interaction with the user. The main program is, however, not supposed to carry out compute intensive tasks or even parallelized tasks by itself.

An ORCA calculation commences with reading and analyzing the ORCA input file. The plausibility of this input is checked by an elaborate “maincheck” routine. Quite obviously, the number of possible combinations of ORCA features is far larger than what could possibly be checked with realistic effort. However, the maincheck routine has evolved over the year in a way that allows to detect the most common combinations of invalid or inconsistent input keywords and take appropriate action (that can be changing some parameters or aborting the job).

After through maincheck the basis sets used in the calculations and the coordinates are completely known. This is then the time to initialize the SHARK integral package and will then carry out the bulk of the computation heavy tasks.

The first module that is being called is `orca_startup`. This module takes care of calculating the one electron integrals such as the HCore and overlap matrices, the metric matrices in RI calculations, the pre-screening matrices for direct SCF and possibly also the two-electron integrals for integral conventional runs.

The next module downstream is the `orca_guess` program. This module has the task to produce an initial set of molecular orbitals and also an initial density matrix from these orbitals. To this end, there are a number of options out of which the most common ones are to a model density guess (PModel) or to read orbitals from a previous

calculation (MOREAD). In the latter case, the calculation may feature a different geometry and/or a different basis set but the number, nature and ordering of atoms need to be consistent with the target system.

In the third step, the program branches out into either the SCF or the CASSCF module (`orca_lean` or `orca_casscf`). These modules produce converged orbitals and a self-consistent field energy as well as one-particle density matrix. The latter is stored in the `DensityContainer`, which is a centralized storage facility for densities that will also be left over after the calculation and can be accessed by the users.

The next step of the calculation consists of calling various post-SCF programs like `orca_mp2`, `orca_mdci` or `orca_autoci`. NEVPT2 and CASPT2 are calculated by code that is integrated with `orca_casscf`. These calculate correlation energies and excited states among many other things.

There are many different pathways that the program can take next, for example geometry optimization or embedding calculations or molecular dynamics to only name a few. Rather than going into each and every one of the possible pathways, we will describe the calculation of molecular properties inside ORCA.

10.2.4 The calculation of molecular properties

For the calculation of molecular properties, ORCA has a fairly unique and strongly streamlined infrastructure that is focusing on the commonalities in the calculated properties and that are independent of the underlying electronic structure method used.

There are three qualitatively different sets of properties that are covered in ORCA property calculations:

- 1) Response properties. These are properties that can be formulated as derivatives of the total energy.
- 2) QDPT properties. These are properties that are calculated by quasi-degenerate perturbation theory (QDPT). This amounts to diagonalizing the Hamiltonian containing external fields and relativistic corrections over a number of roots delivered by the underlying electronic structure method.
- 3) Excited state properties: these are, at least at this point in time, transition moments between different electronic states of the system.

Response properties

A large number of properties can be written as derivatives of the total energy. These are first order properties:

$$X_M = \frac{\partial E}{\partial M} = \sum_{\mu\nu} P_{\mu\nu}^{\pm} \langle \mu | \hat{h}_M | \nu \rangle$$

Here, $P_{\mu\nu}^{\pm}$ is the first-order electron ('+') or spin ('-') density \hat{h}_M is the operator that is describing the property of interest (e.g. the dipole moment operator) and the basis set is $\{\mu\}$.

Second order properties are:

$$X_{MN} = \frac{\partial^2 E}{\partial M \partial N} = \sum_{\mu\nu} P_{\mu\nu}^{\pm} \langle \mu | \hat{h}_{MN} | \nu \rangle + \sum_{\mu\nu} \frac{\partial P_{\mu\nu}^{\pm}}{\partial N} \langle \mu | \hat{h}_M | \nu \rangle$$

Here the important quantity is $\frac{\partial P_{\mu\nu}^{\pm}}{\partial N}$, the first derivative (or "response") of the electron or spin density with respect to perturbation N . It can be shown that the order of perturbations M and N is immaterial and hence one can choose the more convenient perturbation to calculate the response density for.

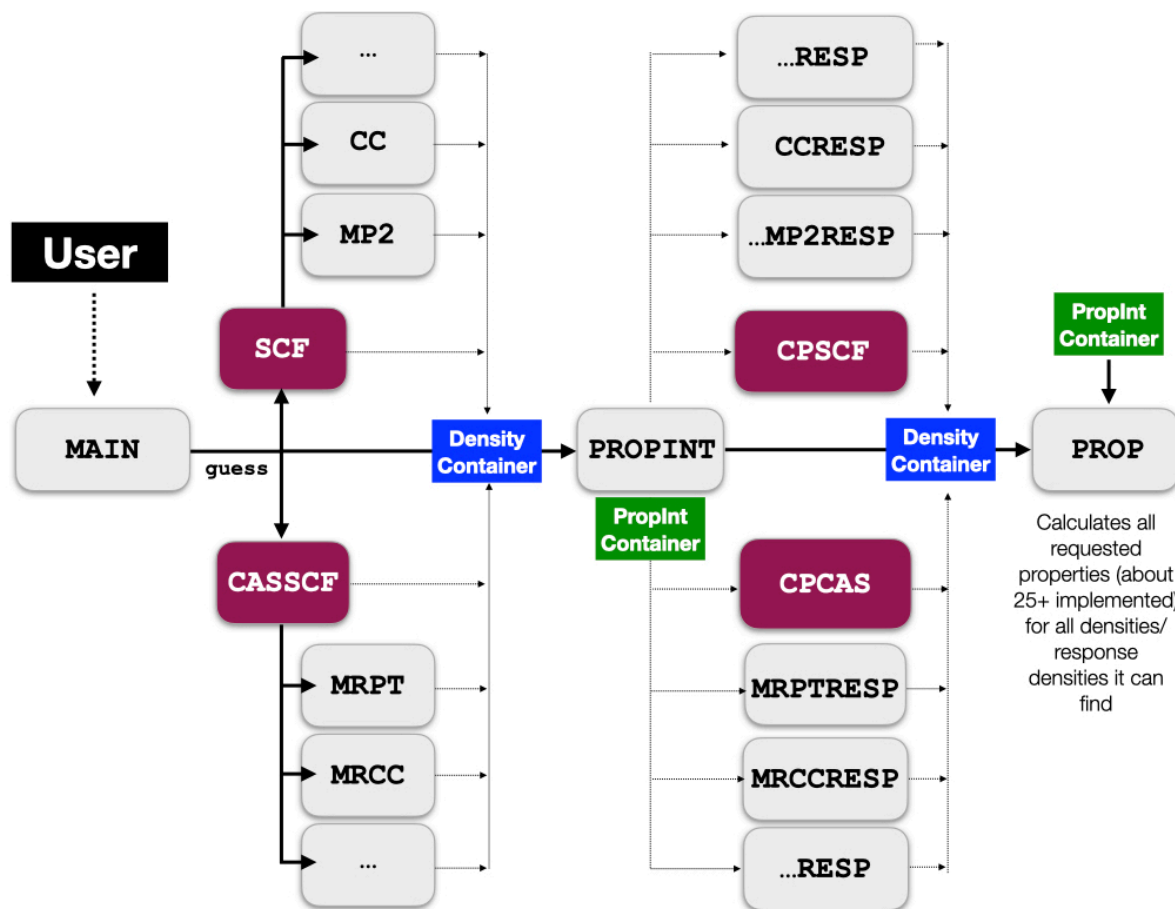
It is important to point out that the equations for response properties can *always* be brought into this form, irrespective of whether the calculation is a Hartree-Fock, DFT, MP2, coupled cluster, CASSCF or full-CI wavefunction.

Given the considerable generality of the property equations, it seems logical to create an infrastructure in which these similarities are exploited to the largest possible extent. In ORCA the central place where all densities and response densities are stored is the `DensityContainer`. This is used intensely throughout the actual calculation and left on disk as `BaseName.densities` where it can be used for visualization.

In terms of the calculation flow the main program contains all the logic to drive this calculation. It first drives the SCF and possibly the post-SCF calculations. After this is done, the SCF program collects the information about

For a number of other electronic structure methods like MP2, CASSCF or AutoCI the respective modules can be run in “response mode” where instead of solving the equations for the energy, they pick up the amplitudes that were determined in the energy run and use them together with the property integrals in order to produce response densities.

The `orca_prop` program works by browsing through the density container and looks for densities that are appropriate for calculating a requested property. As soon as it finds an appropriate density (or combination of unperturbed and response density), it will calculate the property. For example, if you have calculated SCF, MP2 and CCSD in the same calculation and have asked for the calculation of all three densities, `orca_prop` will calculate three dipole moments and print them right next to each other.



1511

The results of all property calculations are stored in the central property file that will be left after the calculation. Users interested in reading properties, it should be read from the property file or its JSON translation.

QDPT properties

Some properties are not calculated as energy derivatives but from quasi degenerate perturbation theory. In this method one diagonalizes the QDPT matrix:

$$Q_{IJ} = \left\langle \Psi_I^{SM} \left| E_I \delta_{IJ} + \hat{H}_{SOC} + \hat{H}_{SS} + \hat{H}_{LB} + \dots \right| \Psi_J^{S'M'} \right\rangle$$

Here $|\Psi_I^{SM}\rangle$ are the roots of a given method that can generate excited states with energy E_I . For example, these can be TD-DFT roots, CASSCF roots, CASSCF roots with energy corrections from NEVPT2 or CASPT2, MRCI-roots, EOM-CCSD roots etc. and \hat{H}_{SOC} is the spin-orbit coupling (SOC) operator, \hat{H}_{SS} the electron-electron spin-spin coupling operator, \hat{H}_{LB} the molecular Zeeman operator etc. In practice, only the principle component $M = S$ (i.e. Ψ_I^{SS}) is calculated and the necessary matrix elements are generated using the Wigner-Eckart theorem.

The diagonalization produces the complex valued relativistic eigenstates as linear combinations of the non-relativistic or scalar relativistic eigenstates. Using the eigenstates relativistic densities or transition densities can be calculated that subsequently can be used to calculate magnetic properties or “relativistic” optical or magneto-optical spectra.

The procedure obviously suffers from a truncation error because only a finite number of roots can be calculated in practice. However, results indicate that the QDPT generated properties compare often very favorable with experimental data provided that the underlying electronic structure method delivers reasonable results.

In ORCA, all QDPT properties are calculated in a consistent infrastructure and then also leads to consistent printing and reporting of the results.

Excited state properties

Closely related to the QDPT infrastructure is the “one-photon spectroscopy” (*OPS*) infrastructure. This is an infrastructure that coordinates the calculation of transition moments using a set of transition densities as input. These might have been stored on disk or might have been generated on the fly. In a similar way to the QDPT infrastructure, these transition moments are generated in a consistent manner throughout all modules of ORCA that can generate excited states.

The exact field/matter coupling Hamiltonian is:

$$A(\mathbf{r}, t) = -A_0 \varepsilon \exp(i\mathbf{k}\mathbf{r} - \omega t)$$

Here \mathbf{k} is the wavevector of the radiation with frequency ω and ε is its polarization. A_0 is the intensity of the radiation, \mathbf{r} the electronic coordinate and t is the time.

Evaluation of the matrix element $\left\langle \Psi_I^{SM} \left| \sum_i A(\mathbf{r}_i, t) \right| \Psi_J^{S'M'} \right\rangle$ amount to generating the “exact” field-matter transition moments. This can be requested by setting the flag `DoFullSemiClassical = true` in the appropriate ORCA input block.

A series expansion of the cosine term yields the familiar electric dipole, electric quadrupole, magnetic dipole etc. contributions. Interestingly, the direct evaluation of the electric dipole term would yield it in the gauge independent “velocity” form. They are related to the more familiar “length” form dipole matrix elements by:

$$\left\langle \Psi_I^{SM} \left| \sum_i \mathbf{p}_i \right| \Psi_J^{S'M'} \right\rangle = \frac{-i}{\hbar} (E_J - E_I) \left\langle \Psi_I^{SM} \left| \sum_i \mathbf{r}_i \right| \Psi_J^{S'M'} \right\rangle$$

The results of the length and velocity transition moment evaluation are expected to match in the basis set limit if the electronic structure method at hand satisfies certain conditions. In practice, they usually do not agree very well, even if large basis sets are used.

In order to generate electric length and velocity as well as higher moment evaluations, you can use the keywords

```

DoDipoleLength   = true
DoDipoleVelocity = true
DecomposeFosc    = true

```

in the appropriate ORCA input blocks that trigger the respective excited state calculation.

10.3 Units and Conversion Factors

Internally the program uses atomic units. This means that the unit of energy is the Hartree (Eh) and the unit of length is the Bohr radius (a_0). The following conversion factors to other units are used:

1 Eh	=	27.2113834 eV	
1 eV	=	8065.54477 cm ⁻¹	= 23.0605 $\frac{\text{kcal}}{\text{mol}}$
1 cm ⁻¹	=	29979.2458 MHz	
1 a_0	=	0.5291772083 Å	
1 a.t.u.	=	2.4188843 10 ⁻¹⁷ s	

DETAILED CHANGE LOG

A.1 Changes ORCA 6.1

A.1.1 New Features

Startup, Guess, LeanSCF

- BUPO algorithm.
- Significant memory improvements in Startup, Guess and LeanSCF.

DFT

- Added B3LYP-3c method.
- Extension of HF-3c, PBEh-3c, B97-3c, and B3LYP-3c for elements Fr-Pu (using original D3 parameters) and r2SCAN-3c up to Lr (Z=103).
- ADFT (Energies, gradient, response, TD-DFT, NOT GIAO, NOT analytic Hessian).
- RPA correlation energy a la Andreas Görling (no gradient, density or response yet; NumGrad, NumFreq).
- Simple input keyword for B2NC-PLYP double-hybrid.
- Added various simple input `LibXC(method)` keywords for DFT functionals, e.g. TASK-LDA, revTM, HSE06, GAS22.

Native xTB Methods

- Native implementation of GFN1-xTB with energy & gradient.
- Native implementation of GFN2-xTB with energy & gradient.
- Native xTB methods can use the ORCA SCF and plotting infrastructure. Plotting of molecular orbitals, densities, and fractional occupation densities (FODs) can be done with `orca_plot`.
- Spin-polarized methods spGFN1-xTB and spGFN2-xTB.
- JSON parameter file for custom parameterizations of GFN1- and GFN2-xTB.
- Native implementation of the ALPB solvation model for xTB methods.

Solvation

- Analytical DRACO gradient.

SCF Linear Response

- Analytical Raman intensities: HF, DFT up to meta-GGAs, RIJ and COSX, implicit solvation, external point charges. Significantly faster than NumFreq.
- Static dip/dip/dip hyperpolarizability: same range of validity as Raman intensities.

Orbital localization

- IAO-based valence virtual orbital localization methods (PMVVO, LIVVO).
- Different options for IAO minimal basis.
- Optionally use a different localization method for virtual MOs via `%loc`.
- LMO plotting enabled in input file.

Optimization

- New redesigned L-OPT - an L-BFGS optimizer for geometry optimization of large systems - more efficient than before.
- !RIGIDBODYOPT for optimization of structures/fragments as rigid bodies.
- Replaced frozen coordinates (`$` syntax) with X/Y/Z Cartesian constraints.

NEB

- More robust initial path generation using S-IDPP.

Excited states

- Core ionization/excitation energies using UHF IP-EOM-CCSD and STEOM-CCSD.
- Multi-core calculations for RHF/UHF Canonical and BT-PNO IP-EOM and STEOM-CCSD (DLPNO for now can only do 1 core at the time).

GS-ROCIS:

- Spin-orbit coupling is now implemented.

DeltaSCF

- Generalized mode following method.
- Freeze-and-release method.

MDCI

- Add `DT_in_Triples` correction for RHF reference with a check to skip for small F_{ia} .

AutoCI

- Third order NEVPT (NEVPT3).
- Fourth order NEVPT restricted to singles and double excitation (NEVPT4(SD)).
- Export single-reference CI/CC 2RDM to JSON.
- UHF CCSDTQ.

AutoCI Response

- RHF CCSD dipole-dipole polarizability, with and without orbital relaxation.
- RHF and UHF CCSD(T) dipole-dipole polarizability, with and without orbital relaxation.
- RHF and UHF CCSD quadrupole-quadrupole polarizability, with and without orbital relaxation.
- RHF and UHF CCSD(T) quadrupole-quadrupole polarizability, with and without orbital relaxation.
- RHF and UHF GIAO-CCSD NMR shielding, with and without orbital relaxation.
- RHF and UHF GIAO-CCSD(T) NMR shielding, with and without orbital relaxation.
- UHF GIAO-CCSD EPR g-Tensor, with and without orbital relaxation.
- UHF GIAO-CCSD(T) EPR g-Tensor, with and without orbital relaxation.

CASSCF Module

- ICE NEVPT2 using the full ranked formulation (ICE-FR-NEVPT2).
- Multiconfiguration pair density functional theory (MCPDFT)
 - with translated functionals tLDA, tPBE, and tPBE0
 - and with complex translated functionals ctLDA, ctPBE, and ctPBE0
- Long-range CASSCF short-range DFT (srDFT)
 - with short-range functionals sr-LDA, sr-PBE, and sr-PBE0
 - short-range pair density functionals: sr-ctLDA, sr-ctPBE, and sr-ctPBE0
 - robust RI approximation for range-separated integrals used for RI-J and RI integral transformation
- Effective Hamiltonian CI-DFT approach for excited states with MCPDFT and srDFT.

CASSCF Linear Response

- New options to toggle the orbital response, solution locking, and Olsen preconditioner in `%casresp` block.
- Nuclear-orbit perturbation implemented (allows for calculation of `Aorb` contribution to EPR A-tensor).
- Block preconditioner to reduce the number of iterations and computation time.

MR methods

- RASCI / ORMAS module.

MCRPA module

- Block preconditioner to reduce the number of iterations and computation time.
- Orbital response can be turned off. This corresponds to a CAS-CI calculation.

Multiscale

- Automated generation of QM and active regions.
- MM-only calculations can now make use of active atoms keyword.
- Added fast Simple_Eq option for topology preparation of QM/QM2 runs.

MD Module

- Langevin thermostats are now available (random force and drag term can be switched on separately).
- Constraints: Can now automatically constraint all bonds/angles involving a certain element (e.g. H).
- Restraints: Can now define custom restraint function by arbitrary mathematical expression.
- Added new “Radius of Gyration” Colvar (can restrain the system to spherical shape, e.g. a drop).
- Metadynamics: Can now define custom Colvars by arbitrary mathematical expression.
- Can now modify atom masses during MD simulation.
- Many more options for velocity initialization (e.g. let certain molecules rotate or translate).
- “Reference trajectory” feature to re-compute snapshots from an existing trajectory. Can also compute work along a path.
- Wavefunction extrapolation (ASPC) to reduce number of required SCF cycles.
- Possibility to apply additional arbitrary forces to atoms (e.g. for experiments with own bias potentials).

Thermochemistry

- Diagonal Born-Oppenheimer correction.
- Spin orbit coupling correction to the total energy via approximate 2 components.

NMR

- Enabled shielding calculations with X2C and the finite nucleus model.

Energy Decomposition and Wavefunction Analysis

- Atomic decomposition of London dispersion (ADLD).
- Atomic decomposition of exchange energy (ADEX).
- LED for covalent bonds (COVALED).
- New implementation of ETS-NOCV.
- Enabled LED with X2C Hamiltonian.
- RIJCOSX can now be used for DLPNO-CCSD(T)-LED with non-hybrid DFT as reference wavefunction.

AILFT

- Extended Spaces AILFT (ESAILFT).

Magnetic Relaxation Rates

- Orbach and Raman II pathways (`orca_magrelax`).

RESP/CHELPG charges

- RESP charges with parabolic and hyperbolic constraints.
- Can now define sets of equivalent atoms which will have identical charges.

ANISO

- Update of the ANISO interface in ORCA to support MRCL.

Basis Sets

- 6-31G* for Ga-Kr.
- CRENBL basis and CRENBL-ECP.
- New Jensen basis sets: pcX-n, aug-pcX-n, pcH-n, aug-pcH-n (n = 1,2,3,4).
- Reimplemented def-SVP, def-SV(P), def-TZVP, def-TZVPP (H-Lr).
- Extended def2-mSVP, def2-mTZVP, and MINIX up to Lr.
- Extended def2/J option for elements Fr-Lr.
- Extended def2/JK option for elements La-Lu, Fr-Lr.

Property File

- Added SCF and job timings.
- Added DBOC, gCP, and SOC energy contributions.
- Added ESD fluorescence rates.
- Easier conversion to JSON via `%output JSONPropFile=true`.

orca_2json

- Compute and export AO-basis Fock matrix as well as separate J, K, XC, and CPCM terms.
- Convert JSON property files back to native ORCA format (with limited support for 6.0 files).
- Added SOC integrals.

Miscellaneous

- Geometry input from GBW files.
- TransInvar for Numerical Hessian.
- Thermochemistry Compound script for the HEAT protocol by Stanton et al.

A.1.2 Improvements

TDDFT

- Improved FollowIRoot stability.
- TD-DFT amplitudes now stored in GBW file.
- TD-DFT calculations are restartable from a GBW file.

GOAT

- GOAT-COARSE option for automated coarse-grained conformational search.
- GOAT-DIVERSITY for a conformational search looking for maximal structure diversity.

DOCKER/SOLVATOR

- Faster by about 100x, with many improvements and updates.
- DOCKER can now do the search only on limited region of the host by defining a center and extent of the search.

AutoCI

- Significantly improved scaling and runtime performance (speed, memory and disk space requirements) for AUTOCI-MP5 energy.
- Improved RHF CCSD(T) 2RDM performance.
- UHF CCSD(T) gradient accuracy (disconnected 3rd and 4th order contributions were missing, now corrected).
- Response: UHF CCSD Polarizability, solver performance improved significantly.
- Response: switched to sequential handling of response equations (less disk requirements for large calculations, compared to solving all perturbations together).
- Printing improvements.

CASSCF

- Aborts at beginning if CASSCF is requested with a dispersion correction.
- General warnings added about number of roots when CASSCF LR is run.
- TRAH submodule: final active orbitals are the same and sorted the same as in other CAS convergers.

CASSCF Linear Response

- More reasonable threshold for LR equation solution is determined by default.
- Uses solution locking by default.
- Checking at start that there are response parameters with which to perform LR calculation.

OpenCOSMO-RS

- Gas-phase orbitals can be used for the calculations in solution.

MD Module

- Added many more warnings (e.g. when two atoms come too close during a simulation).
- Added many different options how to handle non-converged SCF.
- Random velocity initialization now yields a zero angular momentum by default (i.e. system will not rotate).

Thermochemistry

- Extended max number of allowed temperatures and pressures to 100.

DKH

- Only compute the shell-pair data once.

X2C

- Skip grid setup if not needed.

OPS refactor

- Important: keywords to request multipolar spectrum has been simplified. Some Orca 6.0 keywords are no longer available.

QDPT module

- g-tensor g (OZ/SOC) now includes QED correction factor.
- g-tensor g_{gauge} and g (OZ/SOC) can now be printed in molecular frame (with higher print level associated with QDPT).
- Enabled sum-over-states for CAS and MRCI.
- Reduced memory usage for SSC ZFS calculation.

Property File

- All properties and components now documented in the manual.
- Some components are now stored optionally/conditionally.
- Disable property file storage by default for !MM and !GFN-FF (in addition to !GOAT and !MD-L-Opt).
- Common component structure for energy properties: SCF_Energy, AutoCI_Energies, CAS_SCF_Energies, CAS_DCD_Energies, CAS_PT2_Energies, CIS_Energies, MDCI_Energies, MP2_Energies.
- Calculation_info: Removed (unassigned) NumOfFCElectrons and NumOfCorrElectrons; moved “final single point energy” to geometry-dependent Single_Point_Data property.
- Stored MR-AUTOCI energies.
- Method-specific population analyses.
- Method- and state-specific nuclear gradient, removed ATNO component.
- G_Tensor: also store g_{PSO} , g_{DSO} , Δg components.
- A_Tensor, EFG_Tensor: removed activeNucs.

Miscellaneous

- Default MaxCore increased to 4GB.
- UCO/UNO generation performance improvement.
- Allowed simple input of DSD-PBEP86, DSD-BLYP and DSD-PBEB95 with D4.
- DeltaSCF: improved preconditioner with SOSCF.
- NEB: fragment preparation.
- Various improvements in orca_mapspc.
- Various fixes, additions, and improvements in Compound.

A.1.3 Fixed

TDDFT

- DCORR should now respect MaxCore and abort neatly if it is insufficient.
- Wrong JSON output of TDDFT/TDA amplitudes and TDA excitation energy.
- Fix bug of TDDFT calculations with semiempirical methods.
- Fixed GS energy storage when SOC is enabled.