Replacing C_X by $\frac{3}{2}\alpha C_X$ leads to the X-Alpha method. Here, α is an empirical parameter with default value 2/3 in order to recover the HFS method. Empirically, it has been found, however, that values between 2/3 and 1 are more suitable with the value 0.7 being a widely accepted default.

We now have to deal with the odd power law, using techniques developed by Cook and also by Dunlap and co-workers. The idea here is to expand $\rho_\sigma(r)^{\frac{1}{3}}$ itself in another auxiliary basis set:

$$\rho_\sigma\left(\mathbf{r}\right)^{1/3} = \sum_K a_K^\sigma A_K(\mathbf{r})$$

It turns out that we also need (or perhaps not need but want):

$$\rho_\sigma\left(\mathbf{r}\right)^{2/3} = \sum_K b_K^\sigma B_K(\mathbf{r})$$

The functions {A} and {B} are uncontracted exchange auxiliary basis functions. The easiest choice for these auxiliary basis sets is to derive them from the Coulomb fitting basis set, possibly with restricting the maximum angular momentum. Alternatively, the program can provide one of presently four fitting basis of increasing size called XFIT/1 through XFIT/4.

Based on theoretical arguments, the exponents of these fitting bases are scaled differently. The exponents of the A-basis are scaled by the factor 1/3 whereas those of the B-basis are scaled by 2/3. Dunlap remarked that it is sufficient to scale the exponents of the s-functions and we have followed that recommendation but have not observed large differences to scaling all exponents.

The determination of the fitting coefficients {a} and {b} leads to a set of nonlinear equations that can, for example, be solved by a Newton-Raphson procedure. They can be more efficiently also calculated by numerical integration, but that defeats the purpose of the methodology. The equations to be satisfied are:

$$\frac{\partial E}{\partial a_K^\sigma} = C_X \left\{ \sum_{\mu\nu} P_{\mu\nu}^\sigma (\mu\nu A_K) - \sum_{LM} a_L^\sigma b_M^\sigma (A_K A_L B_M) \right\}$$

$$\frac{\partial E}{\partial b_M^\sigma} = \frac{1}{2} C_X \left\{ -\sum_{KL} a_K^\sigma a_L^\sigma \left(A_K A_L B_M\right) + \sum_N b_N^\sigma (B_M B_N) \right\}$$

Where $(A_K A_L B_M)$ are 3-center overlap integrals, μ, ν are basis functions and P^σ is the density matrix for spin $\sigma = \alpha, \beta$. One can bring that into the form:

$$\left(\mathbf{H}(\mathbf{a})^{AB} \overline{\mathbf{S}}^{-1} \mathbf{H}(\mathbf{a})^{AB+}\right) \mathbf{a} = \mathbf{P}^X$$

With

$$P_K^{X;\sigma} = \sum_{\mu\nu} P_{\mu\nu}^\sigma (\mu\nu A_L)$$

$$\overline{S}_{MN} = (B_M B_N)$$

$$H_{KM}^{AB;\sigma} = \sum_L a_L^\sigma \left(A_K A_L B_M\right)$$

Which needs to be solved for a. We then obtain b from

$$\mathbf{b} = \overline{\mathbf{S}}^{-1} \mathbf{H}(\mathbf{a})^{AB+} \mathbf{a}$$

This is essentially the procedure followed in the ORCA implementation. Optimized code has been written for the three center overlap integrals.

The implementation in ORCA covers closed and spin-unrestricted energies, analytic geometry gradients, response properties and excitation energies as well as numerical Hessians. The efficiency of the implementation is such that energies are slower than with numeric integration while all other steps are executed faster.

One unique opportunity that ADFT offers is to scale the exchange for each atom in the molecule differently. This seems warranted, because X-Alpha studies on atoms revealed that optimal α values change between atoms. Some contemplation shows that this can be achieved by scaling the density matrix for the exchange calculation as follows:

$$P^{\sigma}_{\mu_A\nu_B} \rightarrow P^{\sigma}_{\mu_A\nu_B} \left(\frac{3}{2}\alpha_A\right)^{3/8} \left(\frac{3}{2}\alpha_B\right)^{3/8}$$

Alpha-values can be chosen in a variety of ways other than assigning a global value of 0.7. For example, the alpha values can be fitted to reproduce atomic Hartree-Fock energies or atomic coupled-cluster energies. They could also be optimized to reproduce molecular energetics.

The usage of ADFT in ORCA is simple:

```
! ADFT def2-SVP def2/J XFit/3
# or ADFT(XAlpha), ADFT(HF), ADFT(CC)
# default parameters for HF are H-Kr
# default parameters for CC are H-Zn
# XFIT/1 -- XFIT/4 exists but only /3 and /4 are recommended

%method ADFTNRTol 1e-9           # tol. for NR equation solution (def=1e-9)
        ADFTNRMaxIter 6          # max no of NR iterations (def=15)
        ADFTLmaxAuxJ  5          # max L allowed in the AUXJ basis (def=3)
        ADFTLmaxAuxX  5          # max L allowed in the AUXX basis (def=2)
        ADFTXAlpha  [36] 0.71337 # array with alpha values for each element
end
```

Note

- ADFT is really limited to the HFS/X-Alpha method. It makes no sense to input the name of any other functional. It will be ignored. This is due to the way the exchange is fitted which is very specific to this particular functional and the whole theory would be completely different for other functionals.
- The implementation is experimental. It is motivated by the intellectual beauty of the construction and the fact that in early computational chemistry, users got a lot of mileage out of the method. Hence, this is a "back to the roots" kind of movement with the idea to perhaps take grid free DFT to new places in the future.

Relevant Papers:

1. Werpetinski, Katrina S.; Cook, Michael. Grid-free density-functional technique with analytical energy gradients. *Phys. Rev. A*, **1995**, 52, R3397–R3400. DOI: 10.1103/PhysRevA.52.R3397.
2. Werpetinski, Katrina S.; Cook, Michael. A new grid-free density-functional technique: Application to the torsional energy surfaces of ethane, hydrazine, and hydrogen peroxide. *The Journal of Chemical Physics*, **1997**, 106 (17), 7124–7138. arXiv:https://pubs.aip.org/aip/jcp/article-pdf/106/17/7124/19164467/7124_1_online.pdf, DOI: 10.1063/1.473734.
3. Zope, Rajendra R.; Dunlap, Brett I. Slater's Exchange Parameters α for Analytic and Variational Xα Calculations. *Journal of Chemical Theory and Computation*, **2005**, 1 (6), 1193–1200. PMID: 26631663. arXiv:https://doi.org/10.1021/ct050166w, DOI: 10.1021/ct050166w.

- Neese, F. 2025, in preparation

3.8 Random Phase Approximation (RPA)

The random phase approximation is a rather different DFT method that is based on the frequency dependent response function. It is more expensive than standard DFT (same order of magnitude as RI-MP2), has many attractive features, in particular it's good to excellent accuracy while being essentially non-empirical. The seminal work that established RPA in a quantum chemical context is due to Furche and is based on the original work of Langreth and Perdew. The implementation in ORCA closely follows the work of Görling and co-workers who we also gratefully acknowledge for help with the implementation.

The RPA energy is calculated using some set of input orbitals and orbitals energies (in practice, PBE orbitals and energies are recommended). The calculation proceeds by a numerical integration over frequencies:

$$E_{RPAC} \approx -\frac{\pi}{2} \sum_{g} w_g \sum_{K} h_K\left(\omega_g\right)$$

$$h_K(\omega) = -log\left(1 + x_K(\omega)\right) + x_K(\omega)$$

Where x_K are the negative eigenvalues of the Kohn-Sham response matrix expressed in an auxiliary fitting basis (an auxiliarly basis of the '/C' type).

$$X_{KL}(\omega) = \sum_{i} \left(\mathbf{X}^i\right)^T \mathbf{Y}^i(\omega)$$

With

$$X^i_{aK} = (ia|K)$$

$$Y^i_{aK}(\omega) = (ia|K)\, \lambda_{ia}(\omega)$$

$$\lambda_{ia} = f_\sigma \frac{\epsilon_i - \epsilon_a}{\left(\epsilon_i - \epsilon_a\right)^2 + (\omega)^2}$$

Here ϵ_i, ϵ_a are occupied and virtual orbital energies from the SCF calculation $(ia|K)$ is a three-index electron repulsion integral in the MO basis $f_\sigma = 4$ for closed-shell states and $f_\sigma = 2$ for spin-unrestricted calculations. The numerical integration is carried out with a Gauss-Legendre quadrature using the weights and frequencies given by:

$$w_g = w_{g0} \frac{5}{\left(1 - \omega_{g0}\right)^2}$$

$$\omega_g = \frac{5}{2}\left(\frac{1 + \omega_{g0}}{1 - \omega_{g0}}\right)$$

Where w_{g0} and ω_{g0} are the Gauss-Legendre weights and roots mapped on the interval -1 to 1.

The usage of RPA in ORCA is straightforward

```
! PBE RPAC def2-SVP def2/J def2-SVP/C
# we need a /C basis for RPAC and the /J basis for PBE

%method RPACNPoints 50  # Number of numerical integration points (def=50)
end
```

Note

- This is a pilot implementation that only features closed-shell and spin-unrestricted energies. Gradients are done numerically and density matrices are the ones from the underlying DFT calculation, NOT genuine RPA densities.
- Excited states and response properties are NOT available with this method.
- The method requires an integral transformation with scaling O(N^4) and on the order of 50 diagonalizations of the response matrix which is of dimension NAUXC. This can become expensive for larger molecules.

1. Langreth, D.C.; Perdew, J.P. The exchange-correlation energy of a metallic surface. *Solid State Communications*, **1975**, 17 (11), 1425–1429. DOI: 10.1016/0038-1098(75)90618-3.

2. Furche, Filipp. Molecular tests of the random phase approximation to the exchange-correlation energy functional. *Phys. Rev. B*, **2001**, 64, 195120. DOI: 10.1103/PhysRevB.64.195120.

3. Furche, Filipp. Developing the random phase approximation into a practical post-Kohn–Sham correlation model. *The Journal of Chemical Physics*, **2008**, 129 (11), 114105. arXiv:https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/1.2977789/13579843/114105_1_online.pdf, DOI: 10.1063/1.2977789.

4. Trushin, Egor; Thierbach, Adrian; Görling, Andreas. Toward chemical accuracy at low computational cost: Density-functional theory with σ-functionals for the correlation energy. *The Journal of Chemical Physics*, **2021**, 154 (1), 014104. arXiv:https://pubs.aip.org/aip/jcp/article-pdf/doi/10.1063/5.0026849/15583499/014104_1_online.pdf, DOI: 10.1063/5.0026849.

PAPERS:

- Langreth, C.; Perdew, J.P. Solid. State. Comm., **1975**, 17(11), 1425-1429 The Exchange-Correlation Energy of a Metallic Surface [328]
- Furche, F. Phys. Rev. B, **2001**, 64, 195120, Molecular tests oft he Random Phase Approximation tot he Exchange-Correlation Energy Functional [329]
- Furche, F. J. Chem. Phys. **2008**, 129, 114105 Develpping the Random Phase Approximation into a Practical post-Kohn-Sham Correlation Model [330]
- Trushin, E.; Thierbach, A., Görling, A. **2021**, J. Chem. Phys., 154, 014104, Toward Chemical Accuracy at Low Computational Cost: Density-Fnctional Theory with sigma-Functionals for the Correlation Energy [331]

3.9 Perturbation Theory - MP2

In this section, we discuss the second order Møller-Plesset Perturbation Theory (MP2) and its implementation in the `orca_mp2` module. Higher order correlated methods, such as MP3 and MP4 theories, are available through computer generated code in the *AUTOCI module*.

3.9.1 Standard (non-RI) MP2

The standard (or full accuracy) MP2 module has two different branches. One branch is used for energy calculations, the other for gradient calculations.

For standard MP2 energies, the program performs two half-transformations and the half-transformed integrals are stored on disk in compressed form. This appears to be the most efficient approach that can also be used for medium sized molecules.The module should parallelize acceptably well as long as I/O is not limiting.

For standard MP2 gradients, the program performs four quarter transformations that are ordered by occupied orbitals. Here, the program massively benefits from large core memory (`%maxcore`) since this minimizes the number of batches that are to be done. I/O demands are minimal in this approach.

In "memory mode" (Q1Opt>0) basically the program treats batches of occupied orbitals at the same time. Thus, there must be at least enough memory to treat a single occupied MO at each pass. Otherwise the MP2 module will fail. Thus, potentially, MP2 calculations on large molecules take significant memory and may be most efficiently done through the RI approximation.

Alternatively, in the "disk based mode" (Q1Opt=-1) the program performs a half transformation of the exchange integrals and stores the transformed integrals on disk. A bin-sort then leads to the AO operator $K^{ij}(\mu,\nu) = (i\mu|j\nu)$ in (11|22) integral notation. These integrals are then used to make the final K^{ij}(a,b) (a,b=virtual MOs) and the EMP2 pair energy contributions. In many cases, and in particular for larger molecules, this algorithm is much more efficient than the memory based algorithm. It depends, however, much more heavily on the I/O system of the computer that you use. It is important, that the program uses the flags `CFLOAT, UCFLOAT, CDOUBLE` or `UCDOUBLE` in order to store the unsorted and sorted AO exchange integrals. Which flag is used will influence the performance of the program and to some extent the accuracy of the result (float based single precision results are usually very slightly less

accurate; μEh-range deviations from the double precision result[1]). Finally, gradients are presently only available for the memory based algorithm since in this case a much larger set of integrals is required.

The `! MP2` command does the following: (a) it changes the `Method` to `HFGTO` and (b) it sets the flag `DoMP2` to `true`. The program will then first carry out a Hartree-Fock SCF calculation and then estimate the correlation energy by MP2 theory. RHF, UHF and high-spin ROHF reference wavefunctions are permissible and the type of MP2 calculation to be carried out (for high-spin ROHF the gradients are not available) is automatically chosen based on the value of `HFTyp`. If the SCF is carried out conventionally, the MP2 calculation will also be done in a conventional scheme unless the user forces the calculation to be direct. For `SCFMode =Direct` the MP2 energy evaluation will be fully in the integral direct mode.

The following variables can be adjusted in the block for conventional MP2 calculations:

```
%mp2
  EMin       -1.5       # orbital energy cutoff that defines the
                        # frozen core in Eh
  EMax        1.0e3     # orbital energy cutoff that defines the
                        # neglected virtual orbitals in Eh
  EWin        EMin,EMax # the same, but accessed as array
                        # (respects settings in %method block!)
  MaxCore     256       # maximum amount of memory (in MB) to be
                        # used for integral buffering
  ForceDirect false     # Force the calculation to be integral
                        # direct
  RI          false     # use the RI approximation
  F12         false     # apply F12 correction
  Q1Opt                 # For non-RI calculations a flag how to perform
                        # the first quarter transformation
                        #  1 - use double precision buffers
                        #      (default for gradient runs)
                        #  2 - use single precision buffers. This reduces
                        #      the memory usage in the bottleneck step by
                        #      a factor of two. If several passes are re-
                        #      quired, the number of passes is reduced by
                        #      a factor of two.
                        # -1 - Use a disk based algorithm. This respects
                        #      the flags UCFLOAT,CFLOAT,UCDOUBLE and
                        #      CDOUBLE. (but BE CAREFUL with FLOAT)
                        #      (default for energy runs)
  PrintLevel  2         # How much output to produce. PrintLevel 3 produces
                        # also pair correlation energies and other info.
  DoSCS       false     # use spin-component scaling
  Ps          1.2       # scaling factor for ab pairs
  Pt          0.333     # scaling factor for aa and bb pairs
  Density     none      # no density construction
              unrelaxed # only "unrelaxed densities"
              relaxed   # full relaxed densities
  NatOrbs     false     # calculate natural orbitals

  # Generate guess natural orbitals for CASSCF:
  # "TNat" is an alias for natural orbitals with unrelaxed density
  # omitting amplitudes < TNat. (Default = no truncation).
  TNat        0.0       # good results are obtained with 1e-6
```

Note

Throughout this section, indices $i, j, k, \ldots$ refer to occupied orbitals in the reference determinant, $a, b, c, \ldots$ to virtual orbitals and $p, q, r, \ldots$ to general orbitals from either set while $\mu, \nu, \kappa, \tau, \ldots$ refer to basis functions.

[1] However, sometimes, and in particular when transition metals and core orbitals are involved we have met unpleasantly large errors. So – be careful and double check when using floats!

3.9.2 RI-MP2

The RI-MP2 module is of a straightforward nature. The program first transforms the three-index integrals $(ia|\tilde{P})$, where "i" is a occupied, "a" is a virtual MO and "$\tilde{P}$" is an auxiliary basis function that is orthogonalized against the Coulomb metric. These integrals are stored on disk, which is not critical, even if the basis has several thousand functions. The integral transformation is parallelized and has no specifically large core memory requirements.

In the next step, the integrals are read ordered with respect to the occupied labels and the exchange operators $K^{ij}(a,b) = (ia|jb) = \sum_{\tilde{P}}^{\text{NAux}} (ia|\tilde{P})(\tilde{P}|jb)$ are formed in the rate limiting O(N^5) step. This step is done with high efficiency by a large matrix multiplication and parallelizes well. From the exchange operators, the MP2 amplitudes and the MP2 energy is formed. The program mildly benefits from large core memory (%maxcore) as this minimizes the number of batches and hence reads through the integral list.

The RI-MP2 gradient is also available. Here, all necessary intermediates are made on the fly.

In the RI approximation, one introduces an auxiliary fitting basis $\eta_P(\mathbf{r})$ and then approximates the two-electron integrals in the Coulomb metric as:

$$(pq|rs) \approx \sum_{PQ} (pq|P)\, V_{PQ}^{-1}\, (Q|rs) \tag{3.24}$$

where $V_{PQ} = (P|Q)$ is a two-index electron-electron repulsion integral. As first discussed by Weigend and Häser, the closed-shell case RI-MP2 gradient takes the form:

$$E^x_{\text{RI-MP2}} = 2\sum_{\mu\nu P} (\mu\nu|P)^{(x)} \sum_i c_{\mu i}\Gamma^P_{i\nu} + \sum_{RS} V^x_{RS} \left(\mathbf{V}^{-1/2}\gamma\mathbf{V}^{-1/2}\right)_{RS} + \langle \mathbf{DF}^x \rangle \tag{3.25}$$

The **F**-matrix derivative terms are precisely handled as in the non-RI case and need not be discussed any further. Γ^P_{ia} is a three-index two-particle "density":

$$\Gamma^P_{ia} = \sum_{jbQ} (1+\delta_{ij})\, \tilde{t}^{ij}_{ab} V_{PQ}^{-1/2}\, (Q|jb) \tag{3.26}$$

Which is partially transformed to the AO basis by:

$$\Gamma^P_{i\nu} = \sum_a c_{\nu a}\Gamma^P_{ia} \tag{3.27}$$

The two-index analogue is given by:

$$\gamma_{PQ} = \sum_{iaR} \Gamma^Q_{ia}\, (ia|R)\, V_{RP}^{-1/2} \tag{3.28}$$

The RI contribution to the Lagrangian is particularly convenient to calculate:

$$L^{RI}_{ai} = \sum_\mu c_{\mu a} \left[2\sum_{PQ\nu} \Gamma^P_{i\nu}\, (\mu\nu|Q)\, V_{PQ}^{-1/2} \right] \tag{3.29}$$

In a similar way, the remaining contributions to the energy weighted density matrix can be obtained efficiently. Note, however, that the response operator and solution of the CP-SCF equations still proceed via traditional four- index integrals since the SCF operator was built in this way. Thus, while the derivatives of the three-index integrals are readily and efficiently calculated, one still has the separable contribution to the gradient, which requires the derivatives of the four-index integrals.

The RI-MP2 energy and gradient calculations can be drastically accelerated by employing the RIJCOSX or the RIJDX approximation.

3.9.3 Calculating MP2 and RI-MP2 Energies

You can do conventional or integral direct MP2 calculations for RHF, UHF or high-spin ROHF reference wavefunctions. MP3 functionality is not implemented as part of the MP2 module, but can be accessed through the MDCI module. Analytic gradients are available for RHF and UHF. The analytic MP2-Hessians have been deprecated with ORCA-6.0. The frozen core approximation is used by default. For RI-MP2 the $\langle \hat{S}^2 \rangle$ expectation value is computed in the unrestricted case according to [332]. An extensive coverage of MP2 exists in the literature.[180, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345]

```
! MP2 def2-TZVP TightSCF
%paras  rCO = 1.20
        ACOH = 120
        rCH  = 1.08
        end
* int 0 1
C 0 0 0 0.00 0.0 0.00
O 1 0 0 {rCO} 0.0 0.00
H 1 2 0 {rCH} {ACOH} 0.00
H 1 2 3 {rCH} {ACOH} 180.00
*
```

Note

There are two algorithms for MP2 calculations without the RI approximation. The first one uses main memory as much as possible. The second one uses more disk space and is usually faster (in particular, if you run the calculations in single precision using `! FLOAT, UCFLOAT or CFLOAT`). The memory algorithm is used by specifying `Q1Opt >0` in the `%mp2` block whereas the disk based algorithm is the default or specified by `Q1Opt = -1`. Gradients are presently only available for the memory based algorithm.

The RI approximation to MP2[342, 343, 344, 345] is fairly easy to use, too. It results in a tremendous speedup of the calculation, while errors in energy differences are very small. For example, consider the same calculation as before:

```
# only the auxiliary basis set def2-TZVP/C is added to
# the keyword line
#
! RI-MP2 def2-TZVP def2-TZVP/C TightSCF
%mp2     MaxCore 100
         end
%paras   rCO  =  1.20
         ACOH =  120
         rCH  =  1.08
         end
* int 0 1
 C  0 0 0 0.00   0.0     0.00
 O  1 0 0 {rCO}  0.0     0.00
 H  1 2 0 {rCH} {ACOH}   0.00
 H  1 2 3 {rCH} {ACOH}   180.00
*
```

Generally, the RI approximation can be switched on by setting `RI true` in the `%mp2` block. Specification of an appropriate auxiliary basis set ("/C") for correlated calculations is required. Note that if the RIJCOSX method (section *RIJCOSX*) or the RI-JK method (section *RI-JK*) is used to accelerate the SCF calculation, then two basis sets should be specified: firstly the appropriate Coulomb ("/J") or exchange fitting set ("/JK"), and secondly the correlation fitting set ("/C"), as shown in the example below.

```
# Simple input line for RIJCOSX:
! RHF RI-MP2 RIJCOSX def2-TZVP def2/J def2-TZVP/C TightSCF

# Simple input line for RI-JK:
```

(continues on next page)

(continued from previous page)

```
! RHF RI-MP2 RI-JK def2-TZVP def2/JK def2-TZVP/C TightSCF
```

The MP2 module can also do Grimme's spin-component scaled MP2 [346]. It is a semi-empirical modification of MP2 which applies different scaling factors to same-spin and opposite-spin components of the MP2 energy. Typically it gives fairly bit better results than MP2 itself.

```
#
# Spin-component scaled MP2 example
#
! SCS-MP2 def2-TZVPP TightSCF
%paras    rCO  =  1.20
          ACOH =  120
          rCH  =  1.08
          end
* int 0 1
 C  0 0 0 0.00   0.0     0.00
 O  1 0 0 {rCO}  0.0     0.00
 H  1 2 0 {rCH}  {ACOH}  0.00
 H  1 2 3 {rCH}  {ACOH}  180.00
*
```

Energy differences with SCS-MP2 appear to be much better than with MP2 itself according to Grimme's detailed evaluation study. For the sake of efficiency, it is beneficial to make use of the RI approximation using the `RI-SCS-MP2` keyword. The opposite-spin and same-spin scaling factors can be modified using `PS` and `PT` in the `%mp2` block, respectively. By default, `PS` $= 6/5$ and `PT` $= 1/3$.

Note

In very large RI-MP2 runs you can cut down the amount of main memory used by a factor of two if you use the keyword `! FLOAT`. This is more important in gradient runs than in single point runs. Deviations from double precision values for energies and gradients should be in the μEh and sub-μEh range. However, we have met cases where this option introduced a large and unacceptable error, in particular in transition metal calculations. You are therefore advised to be careful and check things out beforehand.

A word of caution is due regarding MP2 calculations with a linearly dependent basis. This can happen, for example, with very diffuse basis sets (see *Linear Dependence* for more information). If some vectors were removed from the basis in the SCF procedure, those redundant vectors are still present as "virtual" functions with a zero orbital energy in the MP2 calculation. When the number of redundant vectors is small, this is often not critical (and when their number is large, one should probably use a different basis). However, it is better to avoid linearly dependent basis sets in MP2 calculations whenever possible. Moreover, in such a situation the orbitals should not be read with the `MORead` and `NoIter` keywords, as that is going to produce wrong results!

3.9.4 Frozen Core Options

In MP2 energy and gradient runs the Frozen Core (FC) approximation is applied by default. This implies that the core electrons are not included in the perturbation treatment, since the inclusion of dynamic correlation in the core electrons usually effects relative energies or geometry parameters insignificantly.

The frozen core option can be switched on or off with `FrozenCore` or `NoFrozenCore` in the simple input line. Furthermore, frozen orbitals can be selected by means of an energy window:

```
%method FrozenCore FC_EWIN end
%mp2 ewin -1.5, 1.0e3 end
```

More information and the different options can be found in section *Frozen Core Options*

3.9.5 MP2 and RI-MP2 Gradients

Geometry optimization with MP2, RI-MP2, SCS-MP2 and RI-SCS-MP2 proceeds just as with any SCF method. With frozen core orbitals, second derivatives of any kind are currently only available numerically. The RIJCOSX approximation (section *RIJCOSX*) is supported in RI-MP2 and hence also in double-hybrid DFT gradient runs (it is in fact the default for double-hybrid DFT since ORCA 5.0). This leads to large speedups in larger calculations, particularly if the basis sets are accurate.

```
#
# MP2 optimization example
#
! SCS-MP2 def2-TZVP OPT NoFrozenCore
* int 0 1
 C  0 0 0 0.00   0.0     0.00
 O  1 0 0 1.20   0.0     0.00
 H  1 2 0 1.09 120.0     0.00
 H  1 2 3 1.09 120.0   180.00
*
```

This job results in:

```
---------------------------------------------------------------------------
Redundant Internal Coordinates

--- Optimized Parameters ---
(Angstroem and degrees)

Definition                    OldVal   dE/dq     Step     FinalVal
---------------------------------------------------------------------------
1. B(O   1,C   0)               1.2081  0.000488 -0.0003    1.2078
2. B(H   2,C   0)               1.1027  0.000009 -0.0000    1.1027
3. B(H   3,C   0)               1.1027  0.000009 -0.0000    1.1027
4. A(O   1,C   0,H   3)          121.85  0.000026   -0.00    121.85
5. A(H   2,C   0,H   3)          116.29 -0.000053    0.01    116.30
6. A(O   1,C   0,H   2)          121.85  0.000026   -0.00    121.85
7. I(O   1,H   3,H   2,C   0)     -0.00 -0.000000    0.00      0.00
---------------------------------------------------------------------------
```

Just to demonstrate the accuracy of RI-MP2, here is the result with RI-SCS-MP2 instead of SCS-MP2, with the addition of def2-TZVP/C:

```
---------------------------------------------------------------------------
Redundant Internal Coordinates

--- Optimized Parameters ---
(Angstroem and degrees)

Definition                    OldVal   dE/dq     Step     FinalVal
---------------------------------------------------------------------------
1. B(O   1,C   0)               1.2081  0.000487 -0.0003    1.2078
2. B(H   2,C   0)               1.1027  0.000009 -0.0000    1.1027
3. B(H   3,C   0)               1.1027  0.000009 -0.0000    1.1027
4. A(O   1,C   0,H   3)          121.85  0.000026   -0.00    121.85
5. A(H   2,C   0,H   3)          116.29 -0.000053    0.01    116.30
6. A(O   1,C   0,H   2)          121.85  0.000026   -0.00    121.85
7. I(O   1,H   3,H   2,C   0)     -0.00  0.000000   -0.00     -0.00
---------------------------------------------------------------------------
```

You see that *nothing* is lost in the optimized geometry through the RI approximation thanks to the efficient and accurate RI-auxiliary basis sets of the Karlsruhe group (in general the deviations in the geometries between standard MP2 and RI-MP2 are very small). Thus, RI-MP2 really is a substantial improvement in efficiency over standard MP2.

Geometric gradients can be calculated with RI-MP2 in conjunction with the RIJCOSX method. They are called the same way as with a conventional SCF wave function, for example to perform a geometry optimization with tight convergence parameters: (Please note that you have to switch on NumFreq for the MP2-Hessian, as the analytical (RI-)MP2-Hessians are no longer available).

```
! RI-MP2 def2-TZVPP def2/J def2-TZVPP/C TightSCF RIJCOSX
! TightOpt
...
```

3.9.6 RIJCOSX-RI-MP2 Gradients

Additional grids are introduced for the RIJCOSX-MP2 gradient. They have sensible default settings and therefore do not usually require any intervention from the user. However, a number of expert options are available, as described below.

The COSX terms in the Z-vector equations are calculated on a grid, controlled by the keywords `Z_GridX` and `Z_IntAccX`, as discussed in sections *Changing TD-DFT, CP-SCF and Hessian Grids* and *CP-SCF Options*. For example, the `DefGrid3` CP-SCF COSX grid can be requested as:

```
%method
  Z_GridX   2     # Lebedev 110-point grid
  Z_IntAccX 3.067 # radial integration accuracy
end
```

The grid used for evaluation of the response operator on the right-hand side of the Z-vector equations (see for example eqs (3.52) and (3.53)) can be independently selected using the keyword `Z_GridX_RHS`. Note that starting with ORCA 5, the usage is different to `Z_GridX` - the choice is between one of the three grids used during the RIJCOSX SCF procedure: a small grid for the initial iterations, a medium grid for the final iterations (default in ORCA 5), and a large grid to evaluate the energy more accurately after the iterations have converged.

```
%method
  Z_GridX_RHS 1   # small SCF grid
              2   # medium SCF grid (default)
              3   # large SCF grid
end
```

Yet another grid is used to evaluate basis functions derivatives. Appropriate parameters are chosen through `! DefGridn` (in addition to the three SCF grids), but one can override this by setting the angular (`GridX`) and radial (`IntAccX`) grids explicitly through:

```
%mp2 GridX    4       # default 4: angular Lebedev grid 302
     IntAccX  4.871   # radial grid
end
```

3.9.7 MP2 and RI-MP2 Second Derivatives

Analytical second-order properties with the MP2, RI-MP2 and double-hybrid DFT methods are available in ORCA for calculations without frozen core orbitals. The most expensive term in the second derivative calculations is the four-external contribution which can be evaluated either via an AO direct (default) or a semi-numerical Chain-of-Spheres approach. In case that the latter approach is chosen, appropriate grid parameters are defined through the `! DefGridn` settings. However, a more fine-grained specification is available to expert users as follows:

```
%mp2 KCOpt _AOBLAS          # (default) AO direct with BLAS routines
           _COSX            # semi-numerical evaluation using the COSX method
     KC_GridX    2          # default 2: angular Lebedev grid 110
     KC_IntAccX  4.020      # radial grid
end
```

Alternatively, all the grid settings can be defined in the `%method` block, as discussed in section *SCF Grid Keyword List*. The first three entries define the three SCF grids, the fourth entry the MP2 grid for basis function derivatives (refer to section *RIJCOSX-RI-MP2 Gradients*) and the fifth entry the grid for the four-external contribution.

```
%method
   IntAccX     Acc1,  Acc2,  Acc3,  Acc4,   Acc5
   GridX       Ang1,  Ang2,  Ang3,  Ang4,   Ang5
end
```

3.9.8 MP2 Properties, Densities and Natural Orbitals

The MP2 method can be used to calculate electric and magnetic properties such as dipole moments, polarizabilities, hyperfine couplings, g-tensors or NMR chemical shielding tensors. For this purpose, the appropriate MP2 density needs to be requested - otherwise the properties are calculated using the SCF density!

Two types of densities can be constructed - an "unrelaxed" density (which basically corresponds to the MP2 expectation value density) and a "relaxed" density which incorporates orbital relaxation. For both sets of densities a population analysis is printed if the SCF calculation also requested this population analysis. These two densities are stored as `JobName.pmp2ur.tmp` and `JobName.pmp2re.tmp`, respectively. For the open shell case case the corresponding spin densities are also constructed and stored as `JobName.rmp2ur.tmp` and `JobName.rmp2re.tmp`.

In addition to the density options, the user has the ability to construct MP2 natural orbitals. If relaxed densities are available, the program uses the relaxed densities and otherwise the unrelaxed ones. The natural orbitals are stored as `JobName.mp2nat` which is a GBW type file that can be read as input for other jobs (for example, it is sensible to start CASSCF calculations from MP2 natural orbitals). The density construction can be controlled separately in the input file (even without running a gradient or optimization) by:

```
#
# MP2 densities and natural orbitals
#
%mp2  Density none         # no density
              unrelaxed    # unrelaxed density
              relaxed      # relaxed density
      NatOrbs true         # Natural orbital construction on or off
      end
```

Below is a calculation of the dipole and quadrupole moments of a water molecule:

```
! RI-MP2 def2-SVP def2-SVP/C
%mp2 density relaxed end
%elprop dipole     true
        quadrupole true
        end
* int 0 1
O 0 0 0 0      0      0
H 1 0 0 0.9584 0      0
H 1 2 0 0.9584 104.45 0
*
```

Another example is a simple g-tensor calculation with MP2:

```
! RI-MP2 def2-SVP def2-SVP/C TightSCF SOMF(1X) NoFrozenCore
%eprnmr gtensor  1
        ori     CenterOfElCharge
end
%mp2 density relaxed end
* int 1 2
O 0 0 0 0      0      0
H 1 0 0 1.1056 0      0
```

(continues on next page)

(continued from previous page)

```
H 1 2 0 1.1056 109.62 0
*
```

NMR chemical shielding as well as g-tensor calculations with GIAOs are only available for RI-MP2. The input for NMR chemical shielding looks as follows:

```
! RIJK RI-MP2 def2-SVP def2/JK def2-SVP/C TightSCF NMR NoFrozenCore
%mp2
  density relaxed # required
end
* int 0 1
O 0 0 0 0      0      0
H 1 0 0 1.1056 0      0
H 1 2 0 1.1056 109.62 0
*
```

Note that by default core electrons are not correlated unless the `NoFrozenCore` keyword is present.

For details, see sections *Perturbation Theory - MP2* and *MP2 Level Magnetic Properties*.

3.9.9 RI-MP2 and Double-Hybrid DFT Response Properties

Starting from ORCA 5, both the electric (for the dipole polarizability) and the magnetic (for NMR shielding and the EPR g-tensor) field response as well as the nucleus-orbit response (hyperfine couplings A_{orb} term) for RI-MP2 (and double-hybrid functionals) is handled by a different implementation of the RI-MP2 second derivatives than that used for geometric Hessian calculations (*MP2 and RI-MP2 Second Derivatives*). This code is more efficient, uses the RI approximation throughout (including the four-external contribution) and supports frozen core orbitals. The implementation is described in detail in refs [347, 348]. Consider the following input for a GIAO-RI-MP2 NMR shielding calculation:

```
! RIJK RI-MP2 def2-SVP def2/JK def2-SVP/C TightSCF NMR NoFrozenCore
%mp2
  Density         relaxed # required
  UsePertCanOrbs  true    # Whether to use perturbed canonical orbitals for
                          # the internal block of the perturbed Fock matrix
  PertCan_EThresh 1e-6    # Energy threshold for special treatment of
                          # degenerate orbital pairs
  PertCan_UThresh 10      # Coefficient threshold for special treatment of
                          # strongly interacting orbital pairs
  FCut            1e-5    # Threshold for internal perturbed Fock elements
  RespStoreT      true    # Whether to precalculate and store all necessary
                          # unperturbed amplitudes on disk
  RespDijConv     false   # Whether to store intermediates required for the
                          # internal block of the response density on disk
end
* int 0 1
O 0 0 0 0      0      0
H 1 0 0 1.1056 0      0
H 1 2 0 1.1056 109.62 0
*
```

By default perturbed canonical orbitals are used for the occupied block, i.e., the internal orbital rotation coefficients are chosen as

$$U_{ij}^{\mathbf{B}} = \frac{F_{ij}^{(\mathbf{B})} - S_{ij}^{(\mathbf{B})}\varepsilon_j}{\varepsilon_j - \varepsilon_i}$$

which results in $F_{ij}^{\mathbf{B}} = 0$, thereby eliminating its contribution to the perturbed amplitudes:

$$T_{ab}^{ij,\mathbf{B}} \leftarrow -\sum_k \left[T_{ab}^{ik} F_{kj}^{\mathbf{B}} + T_{ab}^{kj} F_{ki}^{\mathbf{B}} \right] \tag{3.30}$$

If $|\epsilon_j - \epsilon_i| <$ `PertCan_EThresh` or $|U_{ij}^B| >$ `PertCan_UThresh`, then $U_{ij}^{\mathbf{B}}$ is chosen using the standard formula

$$U_{ij}^{\mathbf{B}} = -\frac{1}{2} S_{ij}^{(\mathbf{B})}$$

And the relevant contributions to eq (3.30) are added, unless $\left|F_{ij}^{\mathbf{B}}\right| <$ `FCut`. The required amplitudes $\mathbf{T}^{ik}$ and $\mathbf{T}^{kj}$ (all amplitudes, in case `UsePertCanOrbs = false`) are stored on disk if `RespStoreT = true` or recalculated as needed otherwise. The latter option is significantly slower and not recommended unless disk space is an issue. Similarly, in the case of insufficient RAM, the option `RespDijConv = true` tells ORCA to store all amplitudes in the batch (required to calculate $D_{ij}^{\mathbf{B}}$) on disk, rather than keep them in memory. The 3-index 2-particle densities, needed for the right-hand side of the Z-vector equations, are always stored on disk.

Note also that in this implementation the RIJCOSX Fock-response terms are calculated with one of the SCF grids, chosen with `Z_GridX_RHS` (see section *CP-SCF Options*).

3.9.10 Local MP2 calculations

Purely domain-based local MP2 methodology dates back to Pulay and has been developed further by Werner, Schütz and co-workers. ORCA features a local MP2 method (DLPNO-MP2) that combines the ideas of domains and local pair natural orbitals, so that RI-MP2 energies are reproduced efficiently to within chemical accuracy. Its default thresholds are chosen to reproduce about 99.9% of the total RI-MP2 correlation energy, resulting in an accuracy of a fraction of 1 kcal/mol for energy differences. Due to the intricate connections with other DLPNO methods, reading of the section *Local correlation (DLPNO)* is recommended. A full description of the method for RHF reference wave functions has been published.[349]

The local MP2 method becomes truly beneficial for very large molecules and can be used to compute energies of systems containing several hundred atoms. Fig. 3.3 shows the scaling behavior for linear alkane chains. Note that this represents an idealized situation. For three-dimensional molecules the crossover with canonical RI-MP2 is going to occur at a later point.

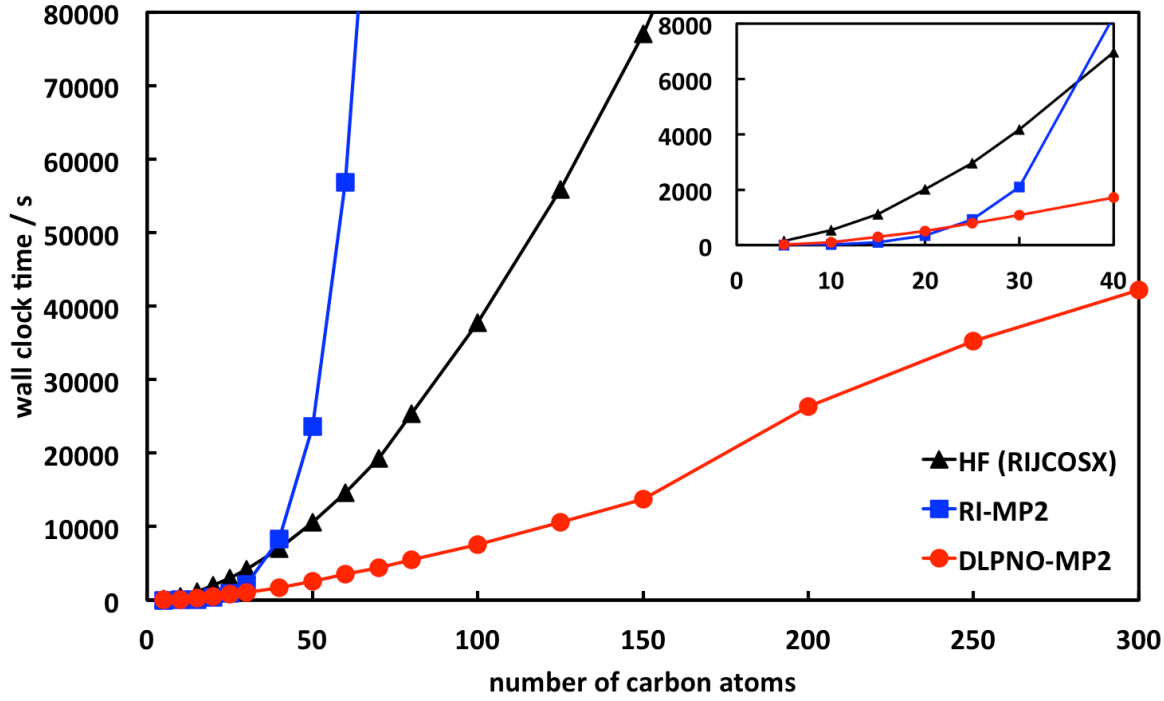

Fig. 3.3: Scaling of the DLPNO-MP2 method with default thresholds for linear alkane chains in def2-TZVP basis. Shown are also the times for the corresponding Hartree-Fock calculations with RIJCOSX and for RI-MP2.

Since DLPNO-MP2 employs an auxiliary basis set to evaluate integrals, its energies converge systematically to RI-MP2 as thresholds are tightened. The computational effort of DLPNO-MP2 with default settings is usually compa-

rable with or less than that of a Hartree-Fock calculation. However, for small and medium-sized molecules, RI-MP2 is even faster than DLPNO-MP2.

Calculations on open-shell systems are supported through a UHF treatment. While most approximations are consistent between the RHF and UHF versions, this is not true for the PNO spaces. **DLPNO-MP2 gives different energies for closed-shell molecules in the RHF and UHF formalisms. When calculating reaction energies or other energy differences involving open-shell species, energies of closed-shell species must also be calculated with UHF-DLPNO-MP2, and not with RHF-DLPNO-MP2.** As for canonical MP2, ROHF reference wave functions are subject to an ROMP2 treatment through the UHF machinery. It is not consistent with the RHF version of DLPNO-MP2, unlike in the case of RHF-/ROHF-DLPNO-CCSD.

In the following, the most important design principles of the RHF-DLPNO-MP2 are pointed out:

Both DLPNO-CCSD(T) and DLPNO-MP2 are linear-scaling methods (albeit the former has a larger prefactor). This means that if a DLPNO-MP2 calculation can be performed, DLPNO-CCSD(T) is often going to be within reach, too. However, CCSD(T) is generally much more accurate than MP2 and thus should be given preference.

A correlation fitting set must be provided, as the method makes use of the RI approximation.

Canonical RI-MP2 energy differences are typically reproduced to within a fraction of 1 kcal/mol. The default thresholds have been chosen so as to reproduce about 99.9 % of the total RI-MP2 correlation energy.

The preferred way to control the accuracy of the method is by means of specifying "LoosePNO", "NormalPNO" and "TightPNO" keywords. "NormalPNO" corresponds to default settings and does not need to be given explicitly. More details and an exhaustive list of input parameters are provided in section *Local MP2 calculations*. Note that the thresholds differ from DLPNO coupled cluster.

Results obtained from RI-MP2 and DLPNO-MP2, or from DLPNO-MP2 with different accuracy settings, must never be mixed, such as when computing energy differences. In calculations involving open-shell species, even the closed-shell molecules need to be subject to a UHF treatment.

Spin-component scaled DLPNO-MP2 calculations are invoked by using the `! DLPNO-SCS-MP2` keyword instead of `! DLPNO-MP2` in the simple input line. Weights for same-spin and opposite-spin contributions can be adjusted as described for the canonical SCS-MP2 method. Likewise, there is a `DLPNO-SOS-MP2` keyword to set the parameters defined by the SOS-MP2 method (but there is no Laplace transformation involved).

The frozen core approximation is used by default. If core orbitals are involved in the calculation, they are subject to the treatment described in section *Local MP2 calculations*.

Calculations can be performed in parallel.

It may be beneficial to accelerate the Hartree-Fock calculation by means of the RIJCOSX method (requiring specification of a second auxiliary set).

Unlike in the 2013 version of the DLPNO methodology, domains are selected by means of the differential overlap $\sqrt{\int i^2(\mathbf{r})\tilde{\mu}'^2(\mathbf{r})d\mathbf{r}}$ between localized MOs i and projected atomic orbitals (PAOs) $\tilde{\mu}'$ which are normalized to unity. The default value for the corresponding cutoff is $T_{\text{CutDO}} = 10^{-2}$.

MP2 amplitudes for each pair of localized orbitals ij are expressed in a basis of pair natural orbitals (PNOs) $\tilde{a}_{ij}$. PNOs are obtained from diagonalization of an approximate, "semicanonical" MP2 pair density $\mathbf{D}^{ij}$. Only PNOs with an occupation number $> T_{\text{CutPNO}}$ are retained, with a default value of $T_{\text{CutPNO}} = 10^{-8}$ for DLPNO-MP2. The pair density is given by:

$$\mathbf{D}^{ij} = \mathbf{T}^{ij\dagger}\tilde{\mathbf{T}}^{ij} + \mathbf{T}^{ij}\tilde{\mathbf{T}}^{ij\dagger} \quad \text{where} \quad \begin{array}{c} T^{ij}_{\tilde{\mu}\tilde{\nu}} = -\dfrac{(i\tilde{\mu}|j\tilde{\nu})}{\varepsilon_{\tilde{\mu}} + \varepsilon_{\tilde{\nu}} - F_{ii} - F_{jj}} \\ \tilde{\mathbf{T}}^{ij} = (1+\delta_{ij})^{-1}\left(4\mathbf{T}^{ij} - 2\mathbf{T}^{ij\dagger}\right) \end{array}$$

Since the occupied block of the Fock matrix is not diagonal in the basis of localized orbitals, the MP2 amplitudes $\mathbf{T}^{ij}$ are obtained by solving the following set of residual equations iteratively (where subscripts of PNOs have been dropped):

$$R^{ij}_{\tilde{a}\tilde{b}} = \left(i\tilde{a}\middle|j\tilde{b}\right) + \left(\varepsilon_{\tilde{a}} + \varepsilon_{\tilde{b}} - F_{ii} - F_{jj}\right) T^{ij}_{\tilde{a}\tilde{b}} - \sum_{k\neq i}\sum_{\tilde{c}\tilde{d}} F_{ik} S^{ij,kj}_{\tilde{a}\tilde{c}} T^{kj}_{\tilde{c}\tilde{d}} S^{kj,ij}_{\tilde{d}\tilde{b}} - \sum_{k\neq j}\sum_{\tilde{c}\tilde{d}} F_{kj} S^{ij,ik}_{\tilde{a}\tilde{c}} T^{ik}_{\tilde{c}\tilde{d}} S^{ik,ij}_{\tilde{d}\tilde{b}} = 0$$

The largest part of the error relative to canonical RI-MP2 is controlled by the domain (`TCutDO`) and PNO (`TCutPNO`) thresholds, which should be adequate for most applications. If increased accuracy is needed (e.g. for

studying weak interactions), tighter truncation criteria can be invoked by means of the `! TightPNO` keyword. Conversely, a less accurate but faster calculation can be performed with the `! LoosePNO` keyword. For more details refer to table Table 3.33.

Fitting domains are determined by means of orbital Mulliken populations with a threshold $T_{\text{CutMKN}} = 10^{-3}$. This threshold results in an error contribution that is typically about an order of magnitude smaller than the overall total energy deviation from RI-MP2.

Prior to performing the local MP2 calculation, pairs of localized molecular orbitals ij are prescreened using an MP2 energy estimate with a dipole approximation, and the differential overlap integral between orbitals i and j. This procedure has been chosen conservatively and leads to minimal errors.

Residual evaluation can be accelerated significantly by neglecting terms with associated Fock matrix elements F_{ik} and F_{kj} below $F_{\text{Cut}} = 10^{-5}$. Errors resulting from this approximation are typically below 1 μE_h and thus negligible.

Sparsity of the MO and PAO coefficient matrices in atomic orbital basis is exploited to accelerate integral transformations for large systems. Truncation of the coefficients is controlled by a parameter T_{CutC}. Neglect of these coefficients has to be performed very carefully in order to avoid uncontrollable errors. The threshold has been chosen so as to make the errors essentially vanish.

By default, core orbitals are frozen in the MP2 module. However, if core orbitals are subject to an MP2 treatment, it is necessary to use a tighter PNO cutoff for pairs involving at least one core orbital. For this purpose core orbitals and valence orbitals are localized separately. The cutoff for pairs involving core orbitals is given by $T_{\text{CutPNO}} \times T_{\text{ScalePNO\_Core}}$, where $T_{\text{ScalePNO\_Core}} = 10^{-2}$ by default. For more details refer to section *Including (semi)core orbitals in the correlation treatment*.

The UHF-DLPNO-MP2 implementation differs somewhat from the RHF case, particularly regarding construction of PNOs, as described below.

A separate set of PAOs $\tilde{\mu}'_\alpha$ and $\tilde{\mu}'_\beta$ is obtained for each spin case.

For $\alpha\beta$ pairs, separate pair domains of PAOs need to be determined for each spin case. For example, the α pair domain $[i_\alpha j_\beta]_\alpha$ is the union of the domains $[i_\alpha]_\alpha$ and $[j_\beta]_\alpha$. The latter domain $[j_\beta]_\alpha$ is determined by evaluating the spatial differential overlap between j_β and α-spin PAOs $\tilde{\mu}'_\alpha$.

One set of PNOs is needed for each same-spin pair. Opposite-spin pairs require a set of α-PNOs and a set of β-PNOs. In total this results in four types of PNO sets.

Semicanonical amplitudes are obtained as follows, where i, j are spin orbitals and $\tilde{\mu}\tilde{\nu}$ are nonredundant spin PAOs.

$$T^{ij}_{\tilde{\mu}\tilde{\nu}} = -\frac{\langle ij||\tilde{\mu}\tilde{\nu}\rangle}{\varepsilon_{\tilde{\mu}} + \varepsilon_{\tilde{\nu}} - F_{ii} - F_{jj}}$$

In the same-spin case $\langle i_\alpha j_\alpha||\tilde{\mu}_\alpha\tilde{\nu}_\alpha\rangle = \langle ij|\tilde{\mu}\tilde{\nu}\rangle - \langle ij|\tilde{\nu}\tilde{\mu}\rangle$, while in the opposite-spin case $\langle i_\alpha j_\beta||\tilde{\mu}_\alpha\tilde{\nu}_\beta\rangle = \langle ij|\tilde{\mu}\tilde{\nu}\rangle$.

For opposite-spin pairs, α-PNOs and β-PNOs are obtained from diagonalisation of $\mathbf{T}^{ij}\mathbf{T}^{ij\dagger}$ and $\mathbf{T}^{ij\dagger}\mathbf{T}^{ij}$, respectively. For same-spin pairs the pair density is symmetric and only one set of PNOs is needed. PNOs are discarded whenever the absolute value of their natural occupation number is below the threshold T_{CutPNO}.

The following residual equations need to be solved for the cases $R^{i_\alpha j_\alpha}_{\tilde{a}_\alpha\tilde{b}_\alpha}$, $R^{i_\beta j_\beta}_{\tilde{a}_\beta\tilde{b}_\beta}$ and $R^{i_\alpha j_\beta}_{\tilde{a}_\alpha\tilde{b}_\beta}$:

$$\begin{aligned} R^{i_\sigma j_\tau}_{\tilde{a}_\sigma\tilde{b}_\tau} &= \left\langle ij \middle|\middle| \tilde{a}\tilde{b} \right\rangle + \left(\varepsilon_{\tilde{a}} + \varepsilon_{\tilde{b}} - F_{ii} - F_{jj}\right) T^{ij}_{\tilde{a}\tilde{b}} \\ &\quad - \sum_{k_\sigma \neq i_\sigma} \sum_{\tilde{c}_\sigma \tilde{d}_\tau} F_{ik} S^{ij,kj}_{\tilde{a}\tilde{c}} T^{kj}_{\tilde{c}\tilde{d}} S^{kj,ij}_{\tilde{d}\tilde{b}} - \sum_{k_\tau \neq j_\tau} \sum_{\tilde{c}_\sigma \tilde{d}_\tau} F_{kj} S^{ij,ik}_{\tilde{a}\tilde{c}} T^{ik}_{\tilde{c}\tilde{d}} S^{ik,ij}_{\tilde{d}\tilde{b}} = 0 \end{aligned}$$

Most approximations are consistent between the RHF and UHF schemes, with the exception of the PNO truncation. This means that results would match for closed-shell molecules with $T_{\text{CutPNO}} = 0$ (provided both Hartree-Fock solutions are identical), but this is not true whenever the PNO space is truncated. Therefore, UHF-DLPNO-MP2 energies should only be compared to other UHF-DLPNO-MP2 energies, even for closed-shell species.

We found that it is necessary to use tighter PNO thresholds for UHF-DLPNO-MP2. With NormalPNO settings the default value is $T_{\text{CutPNO}} = 10^{-9}$. For an overview of accuracy settings refer to table Table 3.33. As in the RHF implementation, the PNO cutoff for pairs involving core orbitals is scaled with $T_{\text{ScalePNO\_Core}}$.

Input for DLPNO-MP2 requires little specification from the user:

```
# DLPNO-MP2 calculation with standard settings
# sufficient for most purposes
! def2-TZVP def2-TZVP/C DLPNO-MP2 TightSCF

# OR: DLPNO-MP2 with tighter thresholds
# May be interesting for weak interactions, calculations with diffuse basis sets␣
↪etc.
! def2-TZVP def2-TZVP/C DLPNO-MP2 TightPNO TightSCF

%maxcore 2000

*xyz 0 1
... (coordinates)
*
```

Explicit correlation has been implemented in the DLPNO-MP2-F12 methodology for RHF reference wave functions.[350] The available approaches are C (keyword `! DLPNO-MP2-F12`) and the somewhat more approximate D (keyword `! DLPNO-MP2-F12/D`). Approach D is generally recommended as it results in a significant speedup while leading only to small errors relative to approach C. In addition to the MO and correlation fitting sets, a CABS basis set is also required for both F12 approaches as shown below.

```
# DLPNO-MP2-F12 calculation using approach C
! cc-pVDZ-F12 aug-cc-pVDZ/C cc-pVDZ-F12-CABS DLPNO-MP2-F12 TightSCF

# OR: DLPNO-MP2-F12 calculation using approach D (recommended)
! cc-pVDZ-F12 aug-cc-pVDZ/C cc-pVDZ-F12-CABS DLPNO-MP2-F12/D TightSCF
```

Table 3.33: Accuracy settings for DLPNO-MP2.

Setting	T_{CutDO}	T_{CutPNO} (RHF)	T_{CutPNO} (UHF)
LoosePNO	2×10^{-2}	10^{-7}	10^{-8}
NormalPNO	1×10^{-2}	10^{-8}	10^{-9}
TightPNO	5×10^{-3}	10^{-9}	10^{-10}

Options specific to DLPNO-MP2 are listed below.

```
%mp2 DLPNO             false  # Do DLPNO-MP2 (also requires RI true)
     TolE              1e-6   # Energy convergence threshold. Default: TolE of SCF
     TolR              5e-6   # Residual convergence threshold. Default: 5 * TolE
     MaxPNOIter        100    # Maximum number of residual iterations
     MaxLocIter        128    # Maximum number of iterations for orbital␣
↪localization
     LocMet            AHFB   # Localization method
                              # options: FB, PM, IAOIBO, IAOBOYS, NEWBOYS, AHFB
     LocTol            1e-6   # Localization convergence tolerance
                              # Default: 0.1 * TolG from SCF
     DIISStart_PNO     0      # First iteration to invoke DIIS extrapolation
     MaxDIIS_PNO       7      # length of DIIS vector
     Damp1_PNO         0.5    # Damping before DIIS is started
     Damp2_PNO         0.0    # Damping with DIIS
     MP2Shift_PNO      0.2    # level shift in amplitude update (Eh)
# Truncation parameters:
     TCutPNO           1e-8   # PNO occupation number cutoff (RHF)
                       1e-9   # PNO occupation number cutoff (UHF)
     TScalePNO_Core    1e-2   # Core PNO scaling factor
     TCutDO            1e-2   # Differential overlap cutoff for domain selection
     TCutMKN           1e-3   # Mulliken population cutoff for fitting domain␣
↪selection
     FCut              1e-5   # Occupied Fock matrix element cutoff
     TCutPre           1e-6   # Energy threshold for dipole prescreening
```

(continues on next page)

(continued from previous page)

```
     TCutDOij         1e-5   # Maximum differential overlap between screened MOs
     TCutDOPre        3e-2   # Cutoff to select PAOs for domains in prescreening
     TCutC            1e-3   # Cutoff for PAO coefficient truncation
     ScaleTCutC_MO    1.0    # Cutoff for MO truncation: TCutC * ScaleTCutC_MO
     PAOOverlapThresh 1e-8   # Threshold for constructing non-redundant PAOs
end
```

3.9.11 Local MP2 gradients

The analytical gradient has been implemented for the RHF variant of the DLPNO-MP2 method.[351, 352] It is a complete derivative of all components in the DLPNO-MP2 energy, and the results are therefore expected to coincide with numerical derivatives of DLPNO-MP2 (minus the noise).

General remarks:

No gradient is presently implemented for the UHF-DLPNO-MP2 variant.

Spin-component scaled MP2 is supported by the gradient.

Double-hybrid density functionals are supported by the gradient.

Only Foster-Boys localization is presently supported. The default converger is `AHFB` with a convergence tolerance that is automatically bound by a constant factor to the SCF orbital gradient tolerance. Using a different converger is possible, but discouraged, as the orbital localization needs to be sufficiently tightly converged.

When calculating properties without the full nuclear gradient, the relaxed MP2 density should be requested.

A number of points regarding geometry optimizations (not all of them specific to DLPNO-MP2) are worth keeping in mind:

As of 2018, we expect that the DLPNO-MP2 gradient can most beneficially be used for geometry optimizations of systems containing around 70-150 atoms. It may be faster than RI-MP2 even for systems containing 50-60 atoms or less, but the timing difference is probably not going to be very large. Of course, structures containing 200 atoms and above can (and have been) optimized, but this may take long if many geometry steps are required. On the other hand, single point gradient or density calculations can be performed for systems containing many hundred atoms.

DLPNO-MP2 is a substantially more expensive method for geometry optimizations than GGA or hybrid DFT functionals. Therefore, it is generally a good idea to start a geometry optimization with a structure that is already optimized at dispersion-corrected DFT level.

RIJCOSX can be used to accelerate exchange evaluation substantially. However, great care needs to be exercised with the grid settings. Insufficiently large grids may lead, for example, to non-planar distortions of planar molecules. The updated default grids in ORCA 5 (`DefGrid2-3`) should be sufficiently accurate to optimize neutral main group compounds. We therefore recommend these grids for general use with some careful checking in more complicated cases. Even with these grids, the calculation is a lot faster than "regular" Hartree-Fock with basis sets of triple zeta quality (or larger).

Using RIJONX is also possible.

Sufficiently large grids should be used for the exchange-correlation functional of double hybrids. The SCF calculation takes only a fraction of the time that is needed for DLPNO-MP2, and sacrificing quality because of an insufficiently accurate grid is a waste of computer time.

Optimization of large structures is often a challenge for the geometry optimizer. It may help to change the trust radius settings, to modify the settings of the `AddExtraBonds` feature, or to change other settings of the geometry optimizer. Sometimes it may be beneficial to check the geometry optimizer settings with a less demanding electronic structure method.

Finally, problems with a geometry optimization may in some cases indeed be caused by the DLPNO approximations. Using `LoosePNO` for accurate calculations is not recommended anyway, and difficulties with `NormalPNO` settings are possibly rectified by switching to `TightPNO`.

During the development process, a number of difficulties were encountered related to the orbital localization Z-vector equations. Great care was taken to work around these problems and to make the procedures as robust as possible, but a number of settings can be changed. For more information on these aspects, we recommend consulting the full paper on the DLPNO-MP2 gradient [352].

Several different solvers are implemented for the orbital localization Z-vector equations. The default is an iterative conjugate gradient solver. As an alternative, the DIIS-accelerated Jacobi solver can be used, but it tends to be inferior to the conjugate gradient solver. Moreover, a direct solver is available as a fail-safe alternative for smaller systems. As the dimension of the linear equation system is $n(n-1)/2$ for n occupied orbitals, the memory requirement and FLOP count increase as $O(n^4)$ and $O(n^6)$, respectively, and using the direct solver becomes unfeasible for large systems.

Settings for the CPSCF solver are specified the same way as for canonical MP2.

Under specific circumstances, the orbital Hessian of the orbital localization function may have zero or near-zero eigenvalues, which can lead to singular localization Z-vector equations. In particular, it is typically a consequence of continuously degenerate localized orbitals, which may (but do not need to) appear in some molecular symmetries.[353] A typical symptom are natural occupation number above 2 and below 0 for systems that would be expected to have MP2 density eigenvalues between 2 and 0 without the DLPNO approximations.

In order to work around the aforementioned problem, a procedure has been implemented to eliminate singular or near-singular eigenvectors of the localization function orbital Hessian. Vectors with an eigenvalue smaller than `ZLoc_EThresh` (or `ZLoc_EThresh_core` for the core orbitals) are subject to the modified procedure. If the program eliminates any eigenvectors, it might sometimes be a good idea to check if calculated properties are reasonable (or at least to check the natural occupation numbers). Eigenvectors of the Hessian are calculated by Davidson diagonalization by default, but direct diagonalization can be chosen for smaller systems, instead.

Diagonalization of the localization orbital Hessian can be switched off altogether by setting `ZLoc_EThresh` to 0.

If the "Asymmetric localization equation residual norm" exceeds the localization Z-vector equation tolerance (`ZLoc_Tol`), there are typically two plausible reasons: (1) the localized orbitals are not sufficiently tightly converged (too large `LocTol`) or unconverged, or (2) the orbital localization Hessian has got small eigenvalues that were not eliminated.

Usage is as simple as that of RI-MP2. For example, the following input calculates the gradient and the natural orbitals:

```
! DLPNO-MP2 def2-SVP def2-SVP/C TightSCF EnGrad
%MaxCore 512
# With 'EnGrad', specifying 'density relaxed' is unnecessary.
# However, it is needed when calculating properties without the gradient.
%MP2 Density Relaxed
     NatOrbs True
     End
*xyz 0 1
C   0.000   0.000   0.000
O   0.000   0.000   1.162
O   0.000   0.000  -1.162
*
```

The implementation supports spin-component scaling and can be used together with double-hybrid density functionals. The latter are invoked with the name of the functional preceded by "`DLPNO-`". A simple geometry optimization with a double-hybrid density functional is illustrated in the example below:

```
! DLPNO-B2PLYP D3 NormalPNO def2-TZVP def2-TZVP/C Opt
%MaxCore 1000
*xyz 0 1
O   0.000   0.000   0.000
H   0.000   0.000   1.000
H   0.000   1.000   0.000
*
```

For smaller systems, the performance difference between DLPNO-MP2 and RI-MP2 is not particularly large, but very substantial savings in computational time over RI-MP2 can be achieved for systems containing more than approximately 70-80 atoms.

Since MP2 is an expensive method for geometry optimizations, it is generally a good idea to use well-optimized starting structures (calculated, for example, with a dispersion-corrected DFT functional). Moreover, it is highly advisable to employ accurate Grids for RIJCOSX or the exchange-correlation functional (if applicable), as the SCF iterations account only for a fraction of the overall computational cost. If calculating calculating properties without requesting the gradient, `Density Relaxed` needs to be specified in the `%MP2-block`.

Only the Foster-Boys localization scheme is presently supported by the derivatives implementation. The default localizer in DLPNO-MP2 is `AHFB`, and changing this setting is strongly discouraged, since tightly converged localized orbitals are necessary to calculate the gradient.

This is an overview over the options related to the gradient:

```
# Settings specific to the localization equation z-solver
%mp2 ZLoc_Solver         CG       # Use conjugate gradient solver (default)
                         DIR      # Use direct solver
                         JAC      # Use DIIS-accelerated Jacobi solver
     ZLoc_Tol            1.0e-3   # Residual convergence tolerance for the
                                  # localization Z-solver
                                  # Default: same value as Z_Tol for CPSCF
     ZLoc_MaxIter        1024     # Maximum localization Z-solver iterations
     ZLoc_MaxDIIS        10       # Number of DIIS vectors for the Jacobi solver
     ZLoc_Shift          0.2      # Shift for the Jacobi solver
# Options for eliminating (near-)singular eigenvectors of the
# orbital Hessian of the localization function.
     ZLoc_EThresh        3.0e-4   # Eigenvectors with an eigenvalue below
                                  # this threshold are eliminated.
     ZLoc_EThresh_core   3.0e-4   # Same as ZLoc_EThresh, but for the core orbitals.
                                  # Default: identical value as ZLoc_EThresh.
# Options for determining eigenvectors of the localization orbital Hessian.
     ZLoc_UseDavidson    True     # Use Davidson diagonalization.
                                  # If false, use direct diagonalization.
     ZLoc_DVDRoots       32       # Number of Davidson roots to be determined.
     ZLoc_DVDNIter       256      # Number of Davidson iterations.
     ZLoc_DVDTolE        3.0e-10  # Eigenvalue tolerance for the Davidson solver.
                                  # Default: 1e-6 * ZLoc_EThresh
     ZLoc_DVDTolR        1.0e-7   # Residual tolerance for the Davidson solver.
                                  # Default: 0.1 * (ZLoc_Tol)^2
     ZLoc_DVDMaxDim      10       # During Davidson diagonalization, the space of␣
↪trial
                                  # vectors is expanded up to MaxDim * DVDRoots.
# Choice of the PNO processing algorithm.
     DLPNOGrad_Opt       AUTO       # Chooses automatically between RAM and DISK
                                    # (default and recommended)
                         RAM        # Enforce memory-based one-pass algorithm
                         DISK       # Enforce disk-based two-pass algorithm
                         BUFFERED   # Buffered algorithm. Usage is discouraged.
                                    # Experimental, unpredictable I/O performance.
end
```

Local MP2 Second Derivatives and Response Properties

Analytical second derivatives with respect to electric and magnetic fields are implemented for closed-shell DLPNO-MP2 (as well as double-hybrid DFT).[354] Thus, analytic dipole polarizability and NMR shielding tensors (with our without GIAOs) are available. The implementation supports spin-component scaling and double-hybrid functionals. Errors in the calculated properties are well below 0.5% when `NormalPNO` thresholds are used. Refer to section *Local MP2 Second Derivatives and Response Properties* for more information about the DLPNO-MP2 second derivatives implementation, as well as to the sections on electric (*Electrical Properties - Electric Moments and Polarizabilities*) and magnetic (*EPRNMR - keywords for magnetic properties*) properties and CP-SCF settings (*CP-SCF Options*). All considerations and options discussed in sections *Local MP2 calculations* and *Local MP2 gradients* apply here as well, while additional remarks specific to second derivatives are given below.

DLPNO-MP2 response property calculations are expected to be faster than the RI-MP2 equivalents for systems larger than about 70 atoms or 300 correlated electrons.

Using the `NormalPNO` default thresholds, relative errors in the calculated properties, due to the local approximations, are smaller than 0.5%, or 5–10 times smaller than the inherent inaccuracy of MP2 vs a more accurate method like CCSD(T).

DLPNO-MP2 second derivatives are much more sensitive to near-linear dependencies and other numerical issues than the energy or first-order properties. We have made efforts to choose reasonable and robust defaults, however we encourage the user to be critical of the results and to proceed with caution, especially if diffuse basis sets or numerical integration (DFT, COSX) are used. In the latter case, `DefGrid3` is recommended.

In particular, the near-redundancy of PAO domains introduces numerical instabilities in the algorithm. Hence, these should be truncated at `PAOOverlapThresh=1e-5`, which is higher than the default for the energy and gradient. Therefore, the energy and gradient in jobs, which include response property calculations, may deviate from jobs, which do not. The difference is much smaller than the accuracy of the method (vs RI-MP2) but it is still advisable to use the same value of `PAOOverlapThresh` in all calculations, when calculating, e.g., relative energies.

For the same reason, if diffuse basis sets are used, it is advised to set `SThresh=1e-6` in the `%scf` block.

Another instability arises due to small differences between the occupation number of kept and discarded PNOs and may result in very large errors. The smallest difference is printed during the DLPNO-MP2 relaxed density calculation:

Smallest occupation number difference between PNOs and complementary PNOs. Absolute: 3.10e-10 Relative: 3.28e-02

We found that a relative difference under 10^{-3}, which is not uncommon, may be cause for concern. To regularize the unstable equations, a level shift is applied, which is equal to T_{CutPNO} multiplied by $T_{\text{ScalePNO\_LShift}}$. A reasonable value of `TScalePNO_LShift=0.1` is set by default for response property calculations but not for gradient (or energy) calculations, as these were not found to suffer from this issue, so the same considerations as above apply.

The option `DLPNOGrad_Opt=BUFFERED` is not implemented for response properties.

Below is an example for a simple DLPNO-MP2 NMR shielding calculation:

```
! DLPNO-MP2 def2-TZVP def2-TZVP/C TightSCF NMR
# MP2 relaxed density is requested automatically
*xyz 0 1
  H 0 0 0
  F 0 0 0.9
*
```

A summary of the additional options used for DLPNO-MP2 response properties is given below:

```
%mp2
     PAOOverlapThresh 1e-5   # Threshold for constructing non-redundant PAOs
                             # Default is 1e-8 for energy/gradient calculations!
     TScalePNO_LShift 0.1    # Level shift for PNO constraint equations:
                             # TScalePNO_LShift * TCutPNO
                             # Default is 0 for energy/gradient calculations!
end
```

An example input for a DSD-PBEP86 calculation of the NMR shielding and dipole polarizability tensors employing DLPNO-MP2 is given below. Note that the def2-TZVP basis set is not necessarily ideal for either shielding or polarizability.

```
! DLPNO-DSD-PBEP86/2013 D3BJ def2-TZVP def2-TZVP/C TightSCF NoFrozenCore
! RIJCOSX def2/J DefGrid3      # Use RIJCOSX with tighter grid settings
! NMR                         # Request NMR shielding
%elprop Polar 1 end           # Request polarizability
%mp2                          # These settings are default for response properties
   Density Relaxed
   PAOOverlapThresh 1e-5
   TScalePNO_LShift 0.1
```

(continues on next page)