

Note

In certain cases, for instance rotations of large sterically limited groups in a molecule, the reaction may only be possible by breaking and reforming a bond along the MEP. You may not always be looking for an initial path completely free of unnecessary connectivity changes!

The user may have a preconceived notion of the saddle point configuration or have an estimate of the path from a calculation carried out at a lower level of theory. The initial path can be generated in such a way as to include an intermediate configuration as one of the images using the 'NEB_TS_XYZFile' keyword. Since this image will be optimized along with the other intermediate images during the NEB calculation the guess does not have to be accurate.

If inspection of the initial path reveals problems, e.g., unnecessary bond breaking, it is often a good idea to insert a reasonable configuration into the initial interpolation to avoid such problems. Moreover, if an NEB calculation is unable to converge to the MEP (or saddle point) with the given maximum number of iterations, the user can restart the calculation from the 'allxyz' file (see file: `basename_MEP.allxyz`) which is written to the disk after each iteration during the optimization. Note, when starting an NEB calculation from an output from a previous CI-NEB calculation and vice-versa the band may require a few iterations to adjust the distribution of images to achieve the desired distribution, depending on the selected spring type and the choice of NEB method.

If the system can be modeled reasonably well using the GFN-xTB method, another possible choice is the generation of an initial path on XTB level (keywords 'XTB0', 'XTB1' or 'XTB2' for GFN0-xTB[321] GFN-xTB[320] or GFN2-xTB[161]). In this case the initial path on IDPP level is refined using an NEB calculation on the chosen XTB level. If this NEB run is successful, the entire MEP on XTB level is used as the initial path. If the NEB run on XTB level is not successful, the initial path on IDPP level is used instead.

Another keyword that makes use of the XTB method is the 'XTBTS' keyword ('XTB0TS', 'XTB1TS' or 'XTB2TS'). In this case the initial path on IDPP level is refined using an NEB-CI calculation on XTB level. If the NEB-CI run is successful, the resulting CI structure is chosen as TS guess structure, and the final initial path is generated using an IDPP path from reactant to TS guess and from TS guess to product. If the NEB-CI run on XTB level is not successful, the initial path on IDPP level is used instead.

4.6.5 Removal of translational and rotational degrees of freedom

For NEB and CI-NEB calculations of molecular systems it is important to project out the six (or five in the case of linear molecules) degrees of freedom corresponding to the global rotation and translation of the system. This can be done either at the start of a calculation or for each optimization step (keyword: *quatern*). For the latter, the center-of-geometry of each image is translated to origin and a rotational matrix is constructed using the quaternion approach[559] to minimize the root-mean-square deviation between any two adjacent images. Depending on whether a fixed center is used (keyword: *fix_center*), the images are either kept in that place or transferred back to their original position. This procedure is repeated for any pair of adjacent images in each step of the optimization. Also, the net effective NEB force is set to zero (keyword: *remove_extern_force*).[560]

4.6.6 Reparametrization of the path

In some cases it may be beneficial to enable redistribution of the images along the path every N iterations (i.e. analogous to the string method[561]) (keyword: *reparam*). The path is then interpolated using either a linear or cubic polynomial fitted to both the coordinates and the tangent to the path, and the images are redistributed evenly along the interpolated path. Both N and the type of interpolation are specified by the user (keyword: *reparam_type*). The cubic interpolation method should be better in calculations where the resolution of the path and hence the estimate of the tangent is good, while the linear interpolation is generally more robust as it is not as dependent on the tangent.

4.6.7 Useful output

After each iteration, the energy profile along the path is obtained by making a piecewise cubic polynomial interpolation using both the energy and the component of the atom force tangential to the path[546]. The interpolation can reveal important information about the MEP in locations between the intermediate images. The interpolation is written to the file 'interp' (see file: `basename.interp`) in each step of the optimization. Moreover, as NEB and CI-NEB calculations can be quite computationally demanding and in order to properly analyze what may have gone wrong in such a calculation, the user can inspect the '.log' file, which includes information about the type of calculation, energy, length of the path, spring forces, atomic forces etc.

4.6.8 Important warning messages

Some tests are carried out during the optimization in order to detect problems on the fly. The angle between the two straight lines going through an image and its neighbors on each side is calculated. If the angle becomes large e.g. exceeding 90° the estimate of the tangent has likely become inaccurate and a better resolution of the path is required. If the angle is close to 180° the ordering of the images may have become incorrect. Especially in the latter case, it may be a good choice of action to terminate the calculation and include a larger number of intermediate images in the subsequent calculation. Some information from the calculation, e.g. a guess for the saddle point, could be incorporated in the new calculation.

Another issue is the identification of an intermediate minimum along the path. If an intermediate minimum is observed in M subsequent iterations (M is supplied by the user) a warning is issued to the user that a possible intermediate minimum may exist along the path. It is a good idea to check the status of the calculation, the path and the convergence behaviour. If the calculation appears to be proceeding normally and heading for convergence, the best course of action is to allow the calculation to finish. However, it is in general better to carry out separate NEB calculations for segments of the MEP on either side of the intermediate minimum. Especially, if the intermediate minimum is deep w.r.t to the reactant and product state energy minima.

In such cases, the image closest to the apparent intermediate minimum is selected and a structural minimization is carried out. The resulting configuration is then used as the initial state or final state in the subsequent CI-NEB calculations.

4.6.9 Parallel execution

If the number of processes (NProcs) specified in the input is larger than 1, NEB will automatically start up in multi-process mode:

NProcs <= NImages

NProcs processes will handle NProcs images independently with 1 process per image. Choose $\text{NProcs} = X * \text{NImages}$ (e.g. $X = 1$ or 0.5)

NProcs > NImages

NProcs processes will handle NImage images, each image being treated by ($\text{NProcs} / \text{NImages}$) processes. If you want to dedicate more than 1 process to each single image-calculation, choose $\text{NProcs} = X * \text{NImages}$ (e.g. $X = 2, 3, 4, \dots$).

Note: If in the second case multiple compute-nodes are involved, the user will need to define the ORCA-specific environment variable `RSH_COMMAND`, which tells the NEB driver how to connect to the individual nodes (set it to either 'rsh' or 'ssh'). However, this may not work with all queueing systems.

If the energy and force calculations are fast (e.g. with semiempirical methods), there is no gain in using multiple processes per image. Starting up and finalizing MPI may consume more time than the gain from parallel processing.

4.6.10 zoomNEB

A preliminary version of the Zoom-NEB (Z-NEB) method has been included in this implementation, where the objective of the method is to locate a saddle point more accurately with a better resolution compared to CI-NEB calculations. The Z-NEB method is an automatic two-step procedure, where in the first step a CI-NEB calculation is carried out to obtain a rough convergence towards the MEP. Then, the region surrounding the highest energy maximum along the path is identified and a new set of images is distributed along this region. In the second step, a free-end CI-NEB calculation is started from this new path. In this calculation, the endpoints are optimized according to the atom force acting perpendicular to the path. This will ensure that the endpoints of the second CI-NEB calculation will converge to the MEP, as well. Furthermore, to maintain the parallelization of the CI-NEB method, the same number of movable images are used in the first and second CI-NEB calculations.

4.6.11 NEB-TS

Probably the most efficient saddle point search methods are obtained when double-ended methods (e.g. NEB) are combined with single-ended methods (e.g., eigenvector-following). In the current implementation, a combination of EW-CI-NEB and EF P-RFO (eigenvector-following partitioned rational function optimization) methods is presented and referred to as the NEB-TS method [543].

In NEB-TS, the EW-CI-NEB method is used to partially converge to the MEP and hence saddle point, i.e., the optimization of the images along the MEP is halted once the climbing image is converged to a prescribed tolerance. Then, the climbing image is used to provide a subsequent eigenvector-following calculation with an accurate initial guess configuration, as is shown in Fig. 4.4 and the tangent at the CI is used to select the eigenvector to be followed. The tangent estimate should already provide an accurate approximation to the unstable mode at the saddle point. The initial Hessian matrix needed for the eigenvector-following calculation can either be computed analytically (if available) or taken as the Almlöf empirical Hessian matrix [8]. If the Almlöf Hessian matrix is used, the curvature at the CI is estimated by using a finite difference approximation (i.e. using the atom force acting on the neighboring images to CI) and used to scale the corresponding eigenvalue of the selected eigenvector, allowing the eigenvector-following calculation to be started from a Hessian matrix of the correct form. The typical ‘%geom’ block can be used to modify the settings of the eigenvector-following calculation.

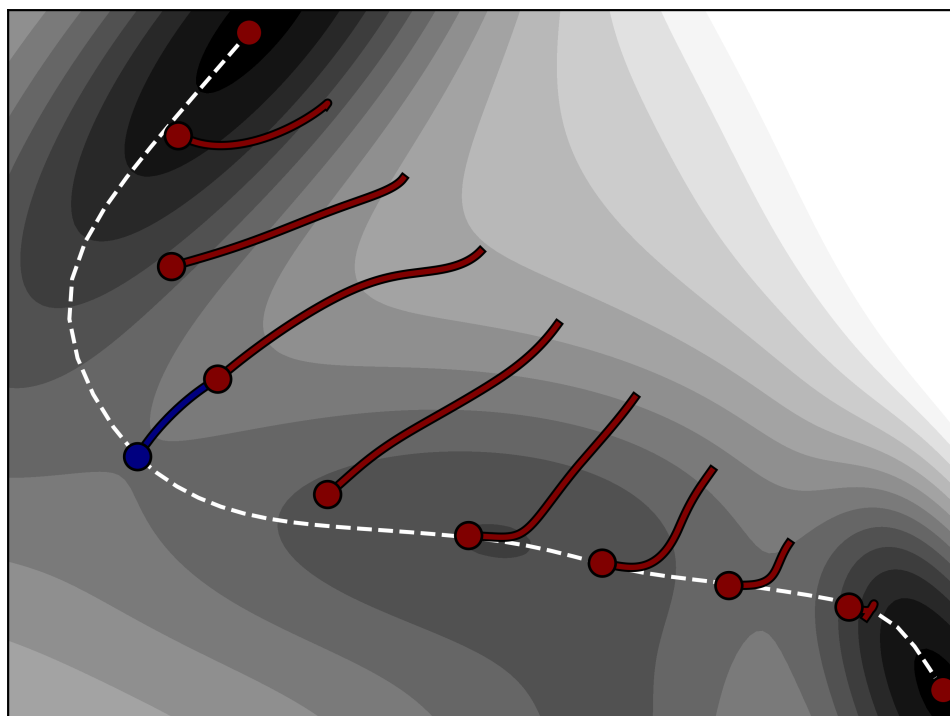


Fig. 4.4: Illustration of how the NEB-TS method works on a two-dimensional Müller-Brown energy surface, $E(x, y)$ [556]. The calculation is started from a linear interpolation between the reactant and product energy minima, using $M = 10$. The images are displaced in the orthogonal direction to the path (red curves) using the CI-NEB algorithm, until a rough convergence to the minimum energy path (white dashed curve) is obtained. The climbing image then provides an approximate saddle point configuration that can be used to start eigenvector-following partitioned rational function optimization to accurately (and swiftly) identify the (higher energy) first-order saddle point. The figure is taken from Ref. [543]

4.6.12 FAST-NEB-TS and LOOSE-NEB-TS

Since our first NEB-TS implementation, we investigated a lot more settings and variants, see [543]. Based on those findings, two new algorithms and convergence threshold settings have been implemented into ORCA. FAST-NEB-TS corresponds to the IDPP-TS in the paper, and LOOSE-NEB-TS corresponds to the actual NEB-TS defaults, which are defined in the paper. Both features, FAST- and LOOSE-NEB-TS, show slightly lower robustness, but need significantly fewer NEB cycles.

4.6.13 NEB / NEB-TS and TD-DFT

The NEB and NEB-TS algorithm now also works in combination with TD-DFT. The input:

```
! NEB-TS
%neb
  product "product.xyz"
end
%tddft
  NRoots 1
  IRoot 1
end
```

not only computes the MEP and TS of the first excited state, but it also prints out (after NEB convergence) the excited state as well as ground state energies over the MEP:

Image	E0	Root 1
0	-67.203	0.000
1	-66.443	1.517
2	-54.382	5.774
3	-42.483	10.473
4	-32.798	14.323
5	-28.056	16.108
6	-34.102	13.915
7	-48.125	8.249
8	-64.188	2.212
9	-67.066	0.018

You can even request an NEB-TS calculation on the ground state PES, and at the same time gain information on the excited PES via the input:

```
! NEB-TS
%neb
  product "product.xyz"
end
%tddft
  NRoots 1
  IRoot 0
end
```

During this NEB calculation, only ground state energies and gradients are computed. Only after NEB convergence, the additional excited state energies are computed on each image, in order to yield the ordering of the states on the MEP:

Image	E0	Root 1
0	0.000	121.545
1	6.281	128.198
2	21.434	143.835
3	35.437	151.581
4	43.223	155.579
5	43.235	155.501
6	35.472	151.495
7	21.498	143.715
8	6.395	128.060
9	0.166	121.425

4.6.14 Summary of Keywords

The following keywords are available:

```
! NEB                # NEB calculation
! NEB-CI             # Climbing Image NEB calculation
! NEB-TS             # NEB calculation plus subsequent TS optimization
! FAST-NEB-TS        # NEB calculation with one iteration only plus subsequent TS opt.
! LOOSE-NEB-TS       # NEB calculation with default convergence criteria from NEB-TS_
→paper
! TIGHT-NEB-TS       # NEB calculation with tighter convergence criteria plus
                     # subsequent TS optimization
! ZOOM-NEB           # NEB calculation plus zoomed NEB calculation
! ZOOM-NEB-CI        # Climbing Image NEB plus zoomed climbing image NEB calculation
! ZOOM-NEB-TS        # NEB calculation plus zoomed NEB calculation plus subsequent
                     # TS optimization
```

(continues on next page)

(continued from previous page)

```

! NEB-IDPP      # IDPP (Initial Path) NEB calculation - for estimation of path
                # length
%neb
  Product "product.xyz" # product structure. Input is mandatory.
  NImages  8           # default 8. Number of images without fixed endpoints,
                        # for free_end total number of images
  PrintLevel  1        # default 1. Normal printout. Use 0 for no printout, higher
                        # numbers (<=4) for more detailed printout.

  TS "TSGuess.xyz"     # Provide guess for the TS structure. Images
                        # are interpolated between reactant and TS guess
                        # and between TS guess and product.
  NEB_TS_Image  3       # default -1. Number of the image the TS guess is used for.
                        # If not defined (= -1), the image which gives lowest RMSD
                        # for all image distances is used.

# Restart option: After each iteration the NEB method stores all image
# structures in an .allxyz file. In case of an abort this file can be used
# for a restart. File should contain the structures for all images.
  Restart_ALLXYZFile "NEB1.allxyz" # use the trajectory from file if filename is
                                   # provided

# Alternatively, NEB can be started on user-prepared wavefunctions for each image.
# The names of these wavefunction files should consist of a user-chosen basename
# and the extension '_imN.gbw', where N is the image number.
# The basename should be provided in the input, ORCA will add extension '_imN.gbw'
  Restart_GBW_BaseName "NEB2"      # use the wavefunctions from file NEB2_imN.gbw

# Check SCF convergence: If true, SCF convergence is checked for and
# calculation aborts if:
# -any of the images does not show SCF convergence in four subsequent cycles.
# -any of the images does not show SCF convergence in two subsequent cycles
# after the gradient is converged.
  CheckSCFConv true                # default true

# PDB file input format:
  Product_PDBFile "product.pdb"   # Product structure in pdb format. If this is
                                   # given, xyz does not need to be given.
  TS_PDBFile      "TSGuess.pdb"   # TS guess structure in pdb format. If this is
                                   # given, xyz does not need to be given.
  Free_End        false           # Use free-end NEB. In this case the NImages
                                   # corresponds to the total number of images.
  PreOpt          false           # do optimization of reactant and product in
                                   # internal coordinates before NEB starts
  NSteps_FoundIntermediate 30     # Number of steps the intermediate has to be
                                   # present
  AbortIf_FoundIntermediate false # If an intermediate is found abort the run.
  NPTS_Interpol   10              # Number of abscissa in cubic polynomial
                                   # interpolation
  Interpolation   IDPP            # Method to generate the images based on the
                                   # reactant, product (and potentially TS guess)
                                   # linear
                                   # IDPP
                                   # XTB0TS - TS on GFN0-xTB level
                                   # XTB0   - entire path on GFN0-xTB level
                                   # XTB1TS - TS on GFN1-xTB level
                                   # XTB1   - entire path on GFN1-xTB level
                                   # XTB2TS - TS on GFN2-xTB level
                                   # XTB2   - entire path on GFN2-xTB level
  Prepare_Frags   true            # Analyze endpoint structures for fragments.
                                   # If two fragments are detected in an

```

(continues on next page)

(continued from previous page)

```

# endpoint structure, reduce distance to
# maximum distance given by Max_Frag_Dist
Max_Frag_Dist      3.5      # Maximum allowed fragment distance. If they
# are farther apart, reduce the distance to
# this value (Ang.)
Bond_Cutoff        1.2      # Factor to multiply sum of covalent radii of
# two atoms by. If the distance is smaller
# than the result, the atoms are considered
# bonded.

# The formulation used to estimate the tangent to the path
Tangent improved    # improved (default)
                    # original

# The type of the spring interaction parallel to the path. Original springs apply
# spring interaction between each degree of freedom of adjacent images, while
# 'image' springs apply a spring interaction between the images
# Spring type
SpringType image    # image / distance (default)
                    # dof / original
                    # ideal
SpringConst 0.01    # The spring constant used to scale the spring
# forces parallel to the path. If energy-
↪weighted
                    # springs are used. This parameter gives the
                    # lower bound value of the spring constant
SpringConst2 0.1    # If energy-weighted spring forces are used.
                    # This parameters give the value for the upper
                    # bound value of the spring constant.
Energy_Weighted true # Employ energy-weighted springs. When
                    # energy-weighted springs are used, the
                    # images tend to accumulate in higher energy
                    # regions of the path.

# The type of the spring interaction perpendicular to the path. The perpendicular
# spring is introduced via a scaling function: cos, tan, costan, which all use
# the spring component perpendicular to the path.
# DNEB is the doubly nudged elastic band method.
PerpSpring no       # no (default)
                    # cos
                    # tan
                    # cosTan
                    # DNEB

LLT_Cos true        # Enables the cos-type spring force
                    # acting perpendicular to the band.

# Translational and rotational degrees of freedom
Quatern always     # no,
                    # startonly
                    # always (default)

# Fix_center specifies whether the centroid of each image should be
# constrained to the origin of the coordinate system or to the center
# of each image individually.
Fix_center True

# Fix_center specifies whether the centroid of each image should be
# constrained to the origin of the coordinate system or to the center
# of each image individually.
Remove_extern_Force True # Removes the net effective NEB force before
                        # translation of the path

```

(continues on next page)

(continued from previous page)

```

# Options for Free-End NEB
Free_End_Type      Perp      # Type of optimization of endpoints in free-end
                                # NEB.
                                # contour - constrain endpoints to a fixed
                                # contour with energy EC, see below
                                # perp - allow end points to move according to
                                # perp. spring force
                                # full - allow to move according to full force,
                                # i.e. relax to energy minimum
Free_End_EC        # Energy contour value for image 0 - needed for
                                # free_end_type = contour
Free_End_EC_End    # Energy contour value for image N - needed for
                                # free_end_type = contour
Free_End_Kappa     # harmonic restraint term - needed for
                                # free_end_type = contour

# Monitor convergence for all images or only the CI.
# Convergence type
ConvType all        # all (default)
                    # CIOOnly

CI false            # Do Climbing image NEB

NEB_TS false        # Do CI NEB and subsequent TS opt.

# Convergence tolerance. In Eh / Bohr (except Tol_Scale ).
Tol_MaxFP_I        1.e-3 # Default. The convergence tolerance for the
                        # maximum component of the atomic force
                        # perpendicular to the path.
Tol_RMSFP_I         5.e-4 # Default. The convergence tolerance for the rms
                        # atomic force perpendicular to the path. Only
                        # applies to regular images.
Tol_MaxF_CI         2.e-3 # The convergence tolerance for the maximum
                        # component of the atomic force acting on the CI.
                        # Only applies to (ZOOM-)NEB-CI/-TS calculations.
                        # Default is 5.e-4 (-CI) and 2.e-3 (-TS)
Tol_RMSF_CI         1.e-3 # The convergence tolerance for the rms atomic
                        # force acting on the CI. Only applies to (ZOOM-)NEB-
↪ CI.
                        # Default is 2.5e-4 (-CI) and 1.e-3 (-TS)
Tol_Turn_On_CI      2.e-2 # Thresholds for max. atomic force for switching on
                        # CI in (ZOOM-)NEB-CI and (ZOOM-)NEB-TS.
                        # Defaults: 0.02 for LBFGS, 0.2 for VPO and FIRE
Tol_Scale           10.0 # For convergence type 'all' the user can scale
                        # the convergence tolerance of the regular images
                        # relative to the CI values using this
                        # multiplicative factor. Only applies to (ZOOM-)NEB-CI
                        # and (ZOOM-)NEB-TS calculations.

# Interpolation and redistribution of the path is performed every 'reparam'
# iterations. The type of interpolation is set by reparam_type.
Reparam_type linear # Cubic
                    # Linear (default)
Reparam 0           # No. of iterations after which the path should be
                    # reparametrized
                    # 0 (default) means: reparametrization is off
Tol_Reparam 0.0     # User-defined threshold at which the path should be
                    # reparametrized
                    # 0.0 (default) means: reparametrization is off

```

(continues on next page)

(continued from previous page)

```

# The optimization method used to converge the band on the MEP / saddle point.
# The L-BFGS is more aggressive and efficient, but also more error-prone.
# VPO is conservative and robust.
Opt_Method LBFGS      # LBFGS      (default)
                      # VPO
                      # FIRE
                      # BFGS

# Options Optim. Method
Maxmove  0.1          # maximum component allowed per step. Default is 0.1 (LBFGS)
                      # and 0.2 (VPO / FIRE)
Stepsize 1.0          # multiplicative factor to scale the size of the step in each
                      # optimization cycle.
                      # Default is 1.0 (LBFGS) and 0.5 (VPO / FIRE)
MaxIter  500          # Maximum number of iterations. 500 for LBFGS, 1000 for VPO /
↳FIRE.
Local    false        # Use local optimization.
                      # Default is false for NEB, but true for (ZOOM-)NEB-CI/-TS.

# Options LBFGS
LBFGS_Mem 20          # the number of previous steps to be kept in memory and
↳used
                      # to construct the approximate Hessian matrix.
LBFGS_DR 1.e-3        # Size of the finite difference step taken at the
                      # initialization of L-BFGS
LBFGS_Restart_On_Maxmove true # Re-initialize L-BFGS for the next step when
                      # the 'max-move' limit is reached.
LBFGS_Reparam_On_Restart false # Re-parametrize when L-BFGS is re-initialized
LBFGS_Precondition true  # If true, then after initialization, the
↳curvature
                      # along direction of the force is estimated and
                      # used to determine the first step

# FIRE parameters
FIRE_INITIAL_DAMP 0.1 # Initial value for the damping factor
FIRE_DAMP_DECR" 0.99 # Decrease of the damping factor
FIRE_STEP_INCR" 1.1  # Factor to increase the stepsize
FIRE_STEP_DECR" 0.5  # Factor to decrease the stepsize
FIRE_MAX_STEP" 5.0   # Default is 10 x Stepsize
FIRE_RETENTION" 5    # Retention before starting acceleration

# Options Zoom
Tol_Turn_On_Zoom 0.1  # use ZOOM-NEB(-CI/TS)
Zoom_Offset 1        # if manual selection is chosen, how many
                      # images away from CI should be chosen
Zoom_Auto true       # automatically select zoom region
Zoom_Alpha 0.5       # determines how much of the barrier
                      # zoom-auto should select
Zoom_Interpolation  # linear (default)
                      # cubic
Zoom_PrintFullTrj  # print full trajectory including fixed region during
↳Zoom

# Set of parameters to adjust the IDPP pre-optimization when generating the
↳initial
# path.
# Options IDPP
IDPP_NMax 7000        # maximum number of cycles allowed in IDPP
IDPP_Tol_MaxF 0.01    # tolerance on the maximum component of the
                      # atomic force perpendicular to the path.
                      # For S-IDPP, this setting is used in the

```

(continues on next page)

(continued from previous page)

```

# final IDPP optimization after all images
# have been added to the path
IDPP_ksp 1.0 # spring constant used to scale the spring
# force parallel to the path.
IDPP_Alpha 0.01 # multiplicative factor to scale the size
# of the step in each opt. cycle
IDPP_MaxMove 0.05 # maximum component allowed per step
IDPP_Debug false # will print out the convergence of IDPP
# and also the optimization trajectory and
# the log file for the IDPP run.
IDPP_Quatern true # Whether quaternions should be used in the
# IDPP optimization
# Interpolation scheme between the endpoint images to use for the pairwise atom
# distances to obtain the ideal distances. Bilinear identifies bonded atoms in
# one endpoint structure that are not bonded in the other and leads to a smaller
# change in the bond distance in the bonded region than in the non-bonded one.
IDPP_Dist_Interpolation # Linear (default)
# Bilinear
IDPP_Bilinear_Partition 0.5 # Path position at which to join the two
# linear distance interpolations. The
↪default
SIDPP false # corresponds to the midpoint of the path.
# Whether the IDPP optimization should use a
# Cartesian linear interpolation initial
# guess or sequentially add images to the
# path
SIDPP_Tol_MaxF 0.01 # Tolerance on the maximum component of the
# atomic force perpendicular to the path
# acting on the images closest to the
↪center
# of the path. This setting defines when
# new images are added to the path
SIDPP_Reparam true # Whether to reparameterize the path once
# after all images have been added to it
# (uses Reparam_type setting) before the
# final IDPP optimization of the full path
SIDPP_Energy_Weighted_Tangent false # Whether to use an energy weight in the
# tangent definition of the images closest
# to the center of the path
SIDPP_Even_NIm true # Whether to perform S-IDPP with one image
# less than requested until all images
# have been added to the path and then
# add the last image afterwards. This
# setting increases robustness of the
# method since no image is placed in the
# center of the path during image addition
SIDPP_Double_NIm false # Whether to perform S-IDPP with twice the
# amount of images and then remove half of
# them automatically. This setting
# increases robustness of the method for
# systems involving very heavy rotations of
# large groups
SIDPP_Ideal_Springconst false # Whether the spring constant between the
# images closest to the center of the path
# should be scaled according to the ratio
# of the number of images that have been
# placed on the path already and the
# requested number of images. This is a
# more aggressive setting pulling the
# images closest to the center of the path
# together more.

```

(continues on next page)

(continued from previous page)

```

SIDPP_HyperSearch          true  # Check connectivity of initial path
                              #   and perform hypersearch for S-IDPP
                              #   initial path with minimal
                              #   unnecessary connectivity changes?

SIDPP_HyperSearch_StrictConnectivity
                              false # Abort if no initial path with no
                              #   unnecessary connectivity changes
                              #   is found? Otherwise, use the one
                              #   with the fewest

# Extra Output options
Monitor_Internals # Up to three internal coordinates can be defined
  {B 0 1}          # for which the values are printed during the NEB run.
  {B 1 5}          # Possible are (B)onds, (A)ngles, (D)ihedrals and (I)mpropers
end
end

```

Output files:

- Configuration and trajectory files:
 - `basename_initial_path_trj.xyz`: The initial path generated at the start of the NEB run and after minimization of RMSD between the reactant and product states.
 - `basename_MEP_trj.xyz`: The final converged MEP trajectory.
 - `basename_MEP_ALL_trj.xyz`: The configurations of each image are appended to this file for each step of the NEB optimization.
 - `basename_trj.xyz`: The trajectory of TS optimization.
 - `basename_MEP.allxyz`: Restart file that includes the configuration of each image from the last iteration of an NEB or NEB-CI iteration.
 - `basename_NEB-CI_converged.xyz`: The configuration of the climbing image after a successful NEB-CI calculation.
 - `basename_NEB-HEI_converged.xyz`: The configuration of the highest energy image after a successful NEB calculation.
 - `basename.xyz`: The configuration of the optimized saddle point using the TS optimization.
- Log files:
 - `basename.interp`: The interpolated energy profile of the path for each iteration during the NEB/NEB-CI optimization.
 - `basename.interp.final`: The energy profile for the converged path of an NEB/NEB-CI optimization.
 - `basename.log`: A general log file containing essential information regarding the run e.g., energy, forces and step size.

4.7 Vibrational Frequencies

Vibrational frequency calculations are available through analytical differentiation of the SCF energy as well as one- or two-sided numerical differentiation of analytical gradients, i.e. for Hartree-Fock and DFT models. For methods without analytical gradient, a numerically calculated gradient can be used (keyword NumGrad) for numerical frequencies. Please note that this will be a very time-consuming calculation.

At present, analytical Hessians can be calculated for SCF only. However, there are some additional restrictions. Analytical Hessians cannot be performed for

- Double-Hybrid functionals
- RI-JK approximation

Note that there is a separate section of the manual dedicated to vibrational spectroscopy (see section [Vibrational Spectroscopy](#)) that contains further examples and describes the calculation of anharmonic effects, Raman spectra, NRVS and things like isotope shifts or visualizing normal modes.

The use of vibrational frequency calculations is fairly simple:

```
# any Hartree-Fock or DFT model can be used here
! BP def2-TZVP

# Tight SCF convergence is advisable to minimize the numerical
# noise in the frequencies.
! TightSCF

# perform a geometry optimization first
! Opt

# Run an analytical or numerical frequency calculation afterwards
! AnFreq # or just ``! Freq''
# numerical:
! NumFreq

# details of the numerical frequency calculation
%freq CentralDiff true # use central-differences (this is the default)
Increment 0.005 # increment in bohr for the
# differentiation (default 0.005)
end

! bohrs
* xyz 0 1
O -1.396288 -0.075107 0.052125
O 1.396289 -0.016261 -0.089970
H -1.775703 1.309756 -1.111179
H 1.775687 0.140443 1.711854
*
```

At the end of the frequency job you get an output like this:

```
-----
VIBRATIONAL FREQUENCIES
-----

0:      0.00 cm**--1
1:      0.00 cm**--1
2:      0.00 cm**--1
3:      0.00 cm**--1
```

(continues on next page)

(continued from previous page)

```

4:      0.00 cm**−1
5:      0.00 cm**−1
6:     311.78 cm**−1
7:     887.65 cm**−1
8:    1225.38 cm**−1
9:    1394.81 cm**−1
10:   3624.88 cm**−1
11:   3635.73 cm**−1

```

This output consists of the calculated vibrational frequencies, the vibrational modes and the thermochemical properties at 298.15 K. In the example above there are six frequencies which are identically zero. These frequencies correspond to the rotations and translations of the molecule. They have been projected out of the Hessian before the calculation of the frequencies and thus, the zero values do not tell you anything about the quality of the Hessian that has been diagonalized. The projection can be turned off by `PROJECTTR FALSE` under `%FREQ`, so that the frequencies of the translations and rotations can deviate from zero and the deviations represent a metric of the numerical error of the Hessian calculation. This is done automatically when there is e.g. an external electric field that makes the exact translational and/or rotational modes have non-zero frequencies (see section *Finite Electric Fields*). However, in normal cases where the molecule is expected to obey both translational and rotational invariance, it is **strongly discouraged** to turn off `PROJECTTR` when calculating thermochemical quantities (especially entropies and Gibbs free energies). This is because when the frequencies of translational and rotational modes exceed `CutOffFreq` (which is 1 cm^{-1} by default), their contributions to the partition function will be calculated using the formulas for vibrations. As a result, the calculated entropy is inaccurate (due to treating translations and rotations as vibrations), is sensitive to numerical noise, and in particular exhibits a finite jump when the (theoretically zero) frequencies of the translational and rotational modes cross `CutOffFreq`. Therefore, the only case where the user needs to turn off `PROJECTTR` manually is when the exact Hessian is expected to have zero translational and rotational frequencies, and one wants to check how much the translational and rotational eigenvalues of the actually computed Hessian deviate from zero. The thermochemical quantities from such a calculation are less reliable and should not be used; even if they differ considerably from the results with `PROJECTTR TRUE`, this does not necessarily mean that the latter are unreliable.

Without `PROJECTTR FALSE`, the reliability of the calculated frequencies has to be judged by comparison of calculations with different convergence criteria, increments, integration grids etc. The numerical error in the frequencies may reach 50 cm^{-1} but should be considerably smaller in most cases. Significant negative frequencies indicate saddle points of the energy hypersurface and prove that the optimization has not resulted in an energy minimum.

OBS: By default, the Hessian is made translation invariant by applying the “acoustic sum rule” ([562]), which reduces the effect of noise from numerical integration coming from DFT or COSX, except for the Partial and Hybrid Hessians where it does not make sense. It can be set to false by using `TRANSINVAR FALSE` under `%FREQ`.

4.7.1 Mass dependencies

Of course the calculated frequencies depend on the masses used for each atom. While this can be influenced later through the `orca_vib` routine (see Section *Isotope Shifts* for more detail) and individually for each atom in the geometry input, one might prefer using a set of precise atomic masses rather than the set of atomic weights (which are set as default). This can be achieved through the `!Mass2016` keyword, which triggers Orca to use those atomic masses representing either the most abundant isotope or the most stable isotope (if all isotopes are unstable) of a certain element (e.g. the mass of ^{35}Cl for chlorine or the mass of ^{98}Tc).

Note

The calculation of numerical frequencies puts rather high demands on both computer time and accuracy. In order to get reliable frequencies make sure that:

- Your SCF is tightly converged. A convergence accuracy of at least 10^{-7} Eh in the total energy and 10^{-6} in the density is desirable.
- Grids of at least `DEFGRID2` (default) are used.

- The use of two-sided (i.e. central) differences increases the computation time by a factor of two but gives more accurate and reliable results.
- Small auxiliary basis sets like DGauss/J or DeMon/J may not result in fully converged frequencies (up to 40 cm⁻¹ difference compared to frequencies calculated without RI). The def2/J universal auxiliary basis sets of Weigend that are now the default in ORCA (or the SARC/J for scalar relativistic calculations) are thought to give sufficiently reliable results.
- Possibly, the convergence criteria of the geometry optimization need to be tightened in order to get fully converged results.
- If you can afford it, decrease the numerical increment to 0.001 Bohr or so. This puts even higher demands on the convergence characteristics of the SCF calculation but should also give more accurate numerical second derivatives. If the increment is too small or too high inaccurate results are expected.

The calculation of analytical frequencies is memory-consuming. To control memory consumption the `%maxcore` parameter must be set. For example, `%maxcore 8192` - use 8 Gb of memory per processor for the calculation. The user should provide the value according to the computer available memory. The batching based on `%maxcore` parameter will be introduced automatically to overcome probable memory shortage.

Numerical frequency calculations are restartable (but analytical frequency calculations are not). If the numerical frequencies job died for one reason or another you can simply continue from where it stopped as in the following example:

```
! STO-3G NumFreq
%freq Restart true # restart an old calculation
                    # this requires .res.* files to be present
end
* int 0 1
  C 0 0 0 0.0000 0 0
  C 1 0 0 1.2160 0 0
  H 1 2 0 1.083 180 0
  H 2 1 3 1.083 180 0
*
```

Note

- You must not change the level of theory, basis set or any other detail of the calculation. Any change will produce an inconsistent, essentially meaningless Hessian.
- The geometry at which the Hessian is calculated must be identical. If you followed a geometry optimization by a frequency run then you must restart the numerical frequency calculation from the optimized geometry.
- Numerical frequencies can be performed in multi-process mode. Please see section *Parallel and Multi-Process Modules* (“Hints on the use of parallel ORCA”) for more information.
- The restart of Numerical frequencies will take off from the result files produced during the preceding run (`BaseName.res.%5d.Type`, with `Type` being `Dipoles`, `Gradients` - and if requested `Ramans` or `Nacmes`). Please make sure that all these local result files get copied to your compute directory. If restart is set and no local files to be found, ORCA will restart from scratch. If ORCA finds a Hessian file on disk, it will only repeat the subsequent analysis.
- The Hessian can be transformed to redundant internal coordinates. More information can be found in section *Printing Hessian in Internal Coordinates*.

4.7.2 Scaling factors for computed frequencies

When comparing computed vibrational frequencies to experiment, effects that are not regarded in the harmonic approximation are often corrected for by scaling the computed harmonic frequencies.

First of all, a standard frequency calculation has to be carried out. After that, a .hess file can be found on disk. This file contains a line with the frequency scaling factor, that can be changed manually:

```
$frequency_scale_factor
0.90      <<<---- you change this to whatever you want
```

then, running `orca_vib` is required:

```
orca_vib myjob_scaled_freq.hess
```

The program reads the Hessian, diagonalizes it and applies the scaling factor. Whatever scaling factor was used in the actual input that generated the Hessian is irrelevant since the Hessian is re-diagonalized. To avoid confusion, we recommend that if the goal is to play with the scaling factor, leave the scaling factor in the input at 1.0. Nothing bad happens if you don't though.

4.7.3 Restarting Numerical Frequency calculations

To restart a numerical frequencies calculation, use:

```
%FREQ
  restart true
END
```

and ORCA will look for `basename.res.{ }` files in the same folder where the calculation is being run, check for what has already been done and restart where it is needed.

4.7.4 Frequencies keyword list

In the ORCA program package, the calculation of frequencies through the numerical or analytical Hessian is done via the `orca_numfreq` module and the combination of the `orca_propint`, `orca_scfresp`, and `orca_prop` modules, respectively.

The parameters to control these frequency calculations can be specified in the `%freq`-block. Relevant simple input keywords are given in [Table 4.6](#).

```
%freq

# Flags to switch frequencies calculation on/off

NumFreq   false      # numerical frequencies (available for all methods)
AnFreq    false      # analytical frequencies (available for HF, DFT)
                        # (One of these options has to be set to true,
                        #  to request a freq calculation)

ScalFreq   1.0        # Scaling factor for frequencies (default = 1.0)
                        # NOTE: Scaling is applied to the frequencies after they
are                                                #
                                                #   calculated. SCALED frequencies will be stored in
the                                              #
                                                #   .hess file and printed in the output file.
                                                #   In the .hess file you have access to the
frequency                                       #
                                                #   scaling factor (see below).
```

(continues on next page)

(continued from previous page)

```

# Flags to control NumFreq calculation:

CentralDiff  true      # use central differences [f(x+h)-f(x-h)]/2h - or -
                        # use one-sided differences [f(x+h)-f(x)]/h
Restart      false     # restart a (numerical) frequency calculation
DX           0.005     # increment h
Increment    0.005     # increment h

Hybrid_Hess  {...} end # calculate (numerical) Hybrid Hessian
Partial_Hess [...] end # calculate (numerical) Partial Hessian

# Flags to control subsequent vibrational analysis:

QuasiRRHO    true      # Evaluate Vibrational Entropy with
                        # Quasi-Rigid Rotor Harmonic Oscillator
QRRHOREfFreq 25        # reference frequency used in the QuasiRRHO in cm-1
                        # default value is 100 from original paper
CutOffFreq   1.0       # Threshold for frequencies to be considered
                        # in spectra, thermochemistry and printout (cm-1)
Temp         298.15     # run the thermochemistry calculations at user-defined
                        # temperatures (max 16 temperatures, separated by ',')
T            290, 292, 295 # same as Temp

XTBVPT2      True      # use XTB for the VPT2 correction of the IR
Delq         0.5       # the displacement in dimensionless coordinates used
                        # during the VPT2

TransInvar   True      # enforce translation invariance after calculating the
↪Hessian?
NumHessTransInvar True # enforce translation invariance while calculating the
↪Hessian?
                        # NOTE: TransInvar does not reduce the computational cost
↪of the
                        # Hessian, and can be applied to analytic as well as
↪numeric
                        # Hessians. NumHessTransInvar reduces the number of
↪gradients
                        # that need to be calculated by 6 (for two-sided
↪differentiation)
                        # or 3 (for one-sided differentiation) for numeric
↪Hessians, but
                        # has no effect for analytic Hessians.
ProjectTR    True      # project out translation and rotation degrees of freedom
                        # in frequency calculation and thermochemistry analysis?

end

```

Table 4.6: Simple input keywords for vibrational frequency calculation

Keyword	Description
Freq	Perform an analytic frequency calculation (Alias: AnFreq)
NumFreq	Perform a numerical frequency calculation
Mass2016	Use the latest (2016) atomic masses of the most abundant or most stable isotopes instead of atomic weights

4.8 Thermochemistry

The second thing that you get automatically as the result of a frequency calculation is a thermochemical analysis based on ideal gas statistical mechanics. This can be used to study heats of formation, dissociation energies and similar thermochemical properties. To correct for the breakdown of the harmonic oscillator approximation for low frequencies, entropic contributions to the free energies are computed, by default, using the Quasi-RRHO approach of Grimme.[280] To switch-off the Quasi-RRHO method and use the RRHO method, use:

```
%freq  QuasiRRHO false
      CutOffFreq 35 # in cm-1
      end
```

Where the CutOffFreq parameter controls the cut-off for the low frequencies mode (excluded from the calculation of the thermochemical properties). Note that the default CutOffFreq is 1 (cm^{-1}) when Quasi-RRHO is on, since Quasi-RRHO behaves much more reasonably for low frequencies than RRHO does. In particular, the entropy contribution calculated by Quasi-RRHO approaches a constant value when the vibrational frequency approaches zero, while the RRHO contribution diverges.

The Quasi-RRHO method smoothly interpolates between the entropy formulas of a harmonic oscillator and a rigid rotor, such that high frequency vibrations behave like harmonic vibrations, and low frequency vibrations behave like rotations with the same frequency. The frequency at which the entropy contribution is a half-half mixture of rotation and vibration is called the “reference frequency” ω_0 of the Quasi-RRHO method, accessible via the `QRRHOREfFreq` keyword in `%freq` (see [Vibrational Frequencies](#)). The default value (100 cm^{-1}) is consistent with the original Quasi-RRHO paper[280], but other papers may choose different values, such as 50 cm^{-1} . Meanwhile, ORCA’s Quasi-RRHO implementation deviates from the original paper in the choice of “average molecular moment of inertia” B_{av} ; while in the original paper it is chosen as a fixed value $10^{-44} \text{ kg} \cdot \text{m}^2$, in ORCA it is given as the isotropically averaged moment of inertia of the actual molecule at hand. This is theoretically more justified than using the same moment of inertia for molecules of different sizes, although the resulting difference in the Gibbs free energies is rather small, usually within 0.1 kcal/mol.

Note that the rotational contribution to the entropy is calculated using the expressions given by Herzberg [563] including the symmetry number obtained from the order of the point group.¹ While this is a good approximation, one might want to modify the symmetry number or use a different expression [564]. For this purpose, the rotational constants (in cm^{-1}) of the molecule are also given in the thermochemistry output.

For example let us calculate a number for the oxygen-oxygen dissociation energy in the H_2O_2 molecule. First run the following jobs:

```
# Calculate a value for the O-O bond strength in H2O2
! B3LYP DEF2-TZVP OPT FREQ BOHRS

* xyz 0 1
O      -1.396288    -0.075107    0.052125
O       1.396289    -0.016261   -0.089970
H      -1.775703     1.309756   -1.111179
H       1.775687     0.140443     1.711854
*
```

```
# Now the OH radical job
! B3LYP DEF2-TZVP OPT FREQ BOHRS

* xyz 0 2
O      -1.396288    -0.075107    0.052125
H      -1.775703     1.309756   -1.111179
*
```

¹ the corresponding equation for the partition function (assuming sufficiently high temperatures) of a linear molecule is $Q_{\text{int}} = \frac{kT}{\sigma h c B}$ and for non-linear molecules $Q_{\text{int}} = \frac{1}{\sigma} \sqrt{\frac{\pi}{ABC}} \left(\frac{kT}{hc}\right)^3$. A, B and C are the corresponding rotational constants, σ is the symmetry number. If you want to choose a different symmetry number, ORCA also provides a table with the values for this entropy contribution for other symmetry numbers. Herzberg reports the following symmetry numbers for the point groups C_1, C_i, C_s : 1; C_2, C_{2v}, C_{2h} : 2; C_3, C_{3v}, C_{3h} : 3; C_4, C_{4v}, C_{4h} : 4; C_6, C_{6v}, C_{6h} : 6; $D_2, D_{2d}, D_{2h} = V_h$: 4; D_3, D_{3d}, D_{3h} : 6; D_4, D_{4d}, D_{4h} : 8; D_6, D_{6d}, D_{6h} : 12; S_6 : 3; $C_{\infty v}$: 1; $D_{\infty h}$: 2; T, T_d : 12; O_h : 24.

The first job gives you the following output following the frequency calculation:

```
-----
THERMOCHEMISTRY AT 298.15K
-----
```

```
Temperature      ... 298.15 K
Pressure         ... 1.00 atm
Total Mass       ... 34.01 AMU
```

Throughout the following assumptions are being made:

- (1) The electronic state is orbitally nondegenerate
- (2) There are no thermally accessible electronically excited states
- (3) Hindered rotations indicated by low frequency modes are not treated as such but are treated as vibrations and this may cause some error
- (4) All equations used are the standard statistical mechanics equations for an ideal gas
- (5) All vibrations are strictly harmonic

```
freq.    370.67 E(vib)    ...    0.21
freq.    947.27 E(vib)    ...    0.03
freq.   1313.46 E(vib)    ...    0.01
freq.   1440.24 E(vib)    ...    0.00
freq.   3739.49 E(vib)    ...    0.00
freq.   3739.86 E(vib)    ...    0.00
```

```
-----
INNER ENERGY
-----
```

The inner energy is: $U = E(\text{el}) + E(\text{ZPE}) + E(\text{vib}) + E(\text{rot}) + E(\text{trans})$

$E(\text{el})$ - is the total energy from the electronic structure calculation
 $= E(\text{kin-el}) + E(\text{nuc-el}) + E(\text{el-el}) + E(\text{nuc-nuc})$

$E(\text{ZPE})$ - the zero temperature vibrational energy from the frequency calculation

$E(\text{vib})$ - the finite temperature correction to $E(\text{ZPE})$ due to population of excited vibrational states

$E(\text{rot})$ - is the rotational thermal energy

$E(\text{trans})$ - is the translational thermal energy

Summary of contributions to the inner energy U :

Electronic energy	...	-151.55083691 Eh	
Zero point energy	...	0.02631509 Eh	16.51 kcal/mol
Thermal vibrational correction	...	0.00040105 Eh	0.25 kcal/mol
Thermal rotational correction	...	0.00141627 Eh	0.89 kcal/mol
Thermal translational correction	...	0.00141627 Eh	0.89 kcal/mol

```
-----
Total thermal energy                                -151.52128823 Eh
```

Summary of corrections to the electronic energy:

(perhaps to be used in another calculation)

Total thermal correction	0.00323359 Eh	2.03 kcal/mol
Non-thermal (ZPE) correction	0.02631509 Eh	16.51 kcal/mol

Total correction	0.02954868 Eh	18.54 kcal/mol

```
-----
ENTHALPY
-----
```

(continues on next page)

(continued from previous page)

The enthalpy is $H = U + k_B T$
 k_B is Boltzmann's constant

Total free energy	...	-151.52129054 Eh	
Thermal Enthalpy correction	...	0.00094421 Eh	0.59 kcal/mol

Total Enthalpy	...	-151.52034633 Eh	
----------------	-----	------------------	--

Note: Rotational entropy computed according to Herzberg
 Infrared and Raman Spectra, Chapter V,1, Van Nostrand Reinhold, 1945
 Point Group: C2, Symmetry Number: 2
 Rotational constants in cm⁻¹: 10.087644 0.882994 0.851333

Vibrational entropy computed according to the QRRHO of S. Grimme
 Chem.Eur.J. 2012 18 9955

 ENTROPY

The entropy contributions are $T \cdot S = T \cdot (S(\text{el}) + S(\text{vib}) + S(\text{rot}) + S(\text{trans}))$

S(el) - electronic entropy
 S(vib) - vibrational entropy
 S(rot) - rotational entropy
 S(trans) - translational entropy

The entropies will be listed as multiplied by the temperature to get
 units of energy

Electronic entropy	...	0.00000000 Eh	0.00 kcal/mol
Vibrational entropy	...	0.00059250 Eh	0.37 kcal/mol
Rotational entropy	...	0.00789993 Eh	4.96 kcal/mol
Translational entropy	...	0.01734394 Eh	10.88 kcal/mol

Final entropy term	...	0.02583637 Eh	16.21 kcal/mol
--------------------	-----	---------------	----------------

In case the symmetry of your molecule has not been determined correctly
 or in case you have a reason to use a different symmetry number we print
 out the resulting rotational entropy values for sn=1,12 :

sn= 1 S(rot)=	0.00855439 Eh	5.37 kcal/mol
sn= 2 S(rot)=	0.00789993 Eh	4.96 kcal/mol
sn= 3 S(rot)=	0.00751710 Eh	4.72 kcal/mol
sn= 4 S(rot)=	0.00724548 Eh	4.55 kcal/mol
sn= 5 S(rot)=	0.00703479 Eh	4.41 kcal/mol
sn= 6 S(rot)=	0.00686265 Eh	4.31 kcal/mol
sn= 7 S(rot)=	0.00671710 Eh	4.22 kcal/mol
sn= 8 S(rot)=	0.00659102 Eh	4.14 kcal/mol
sn= 9 S(rot)=	0.00647981 Eh	4.07 kcal/mol
sn=10 S(rot)=	0.00638033 Eh	4.00 kcal/mol
sn=11 S(rot)=	0.00629034 Eh	3.95 kcal/mol
sn=12 S(rot)=	0.00620819 Eh	3.90 kcal/mol

 GIBBS FREE ENERGY

The Gibbs free energy is $G = H - T \cdot S$

(continues on next page)

(continued from previous page)

```

Total enthalpy          ...   -151.52034633 Eh
Total entropy correction ...   -0.02583637 Eh   -16.21 kcal/mol
-----
Final Gibbs free energy ...   -151.54618270 Eh

For completeness - the Gibbs free energy minus the electronic energy
G-E(el)                ...         0.00465413 Eh       2.92 kcal/mol

```

And similarly for the OH-radical job.

----- INNER ENERGY -----

The inner energy is: $U = E(\text{el}) + E(\text{ZPE}) + E(\text{vib}) + E(\text{rot}) + E(\text{trans})$

$E(\text{el})$ - is the total energy from the electronic structure calculation
 $= E(\text{kin-el}) + E(\text{nuc-el}) + E(\text{el-el}) + E(\text{nuc-nuc})$

$E(\text{ZPE})$ - the zero temperature vibrational energy from the frequency calculation

$E(\text{vib})$ - the finite temperature correction to $E(\text{ZPE})$ due to population of excited vibrational states

$E(\text{rot})$ - is the rotational thermal energy

$E(\text{trans})$ - is the translational thermal energy

Summary of contributions to the inner energy U:

```

Electronic energy          ...   -75.73492538 Eh
Zero point energy          ...     0.00837287 Eh       5.25 kcal/mol
Thermal vibrational correction ...     0.00000000 Eh       0.00 kcal/mol
Thermal rotational correction ...     0.00094418 Eh       0.59 kcal/mol
Thermal translational correction ...     0.00141627 Eh       0.89 kcal/mol
-----
Total thermal energy          ...   -75.72419205 Eh

```

Summary of corrections to the electronic energy:
(perhaps to be used in another calculation)

```

Total thermal correction          0.00236045 Eh       1.48 kcal/mol
Non-thermal (ZPE) correction      0.00837287 Eh       5.25 kcal/mol
-----
Total correction                  0.01073332 Eh       6.74 kcal/mol

```

----- ENTHALPY -----

The enthalpy is $H = U + k_B \cdot T$

k_B is Boltzmann's constant

```

Total free energy          ...   -75.72419205 Eh
Thermal Enthalpy correction ...     0.00094421 Eh       0.59 kcal/mol
-----
Total Enthalpy              ...   -75.72324785 Eh

```

Note: Rotational entropy computed according to Herzberg
Infrared and Raman Spectra, Chapter V,1, Van Nostrand Reinhold, 1945
Point Group: C2v, Symmetry Number: 1
Rotational constants in cm-1: 0.000000 18.628159 18.628159

Vibrational entropy computed according to the QRRHO of S. Grimme
Chem.Eur.J. 2012 18 9955

(continues on next page)