

(continued from previous page)

```
AtomName1 X Y Z
...
```

Thus, the structures are simply of the standard XYZ format and they are provided one after the other. There is no need to add any extra character between them. This was the case for ORCA versions older than 6.0.0, where the structures were separated by a ">" sign. The user can still use this format, if preferred, but is not needed anymore. After processing the XYZ file, single point calculations are performed on each structure in sequence and the results are collected at the end of the run in the same kind of trajectory .dat files as produced from trajectory calculations.

In order to aid in using this feature, the relaxed surface scans produce a file called `MyJob.allxyz` that is of the correct format to be re-read in a subsequent run.

## 4.2.6 Surface Scan Keywords

Table 4.4: %geom block input keywords and options for surface scanned geometry optimization.

Keyword	Option / Value	Description
Scan	B 1 2 = v1, v2, N	Scan atom 1 to atom 2 bond distance from v1 to v2 in N steps.
	B 1 2 [v1 v2 ...]	Scan atom 1 to atom 2 bond with custom step values.
	A 1 2 3	Scan an angle using atoms 1,2,3.
	D 1 2 3 4	Scan a dihedral of atoms 1,2,3,4.
Simul_Scan	true	Enable simultaneous multidimensional scans (up to 3 coordinates).
FullScan	true	Forces full surface scan before TS optimization (!ScanTS).

## 4.3 Transition State Searches

### 4.3.1 Transition State Search Theory

As transition state finder we implemented the well-established eigenvector following algorithm using a P-RFO step as implemented by Baker [538]. This algorithm is a quasi-Newton like algorithm.

The Taylor series of the energy, truncated after the quadratic term, is:

$$E = E_0 + g_q^+ \Delta q_q + \frac{1}{2} \Delta q^+ H_q \Delta q \quad (4.16)$$

The Newton-Raphson step to get from the actual point to a stationary point is:

$$\Delta q = -H_q^{-1} g_q = \sum -\frac{V_i^+ g_q V_i}{b_i} \quad (4.17)$$

with  $V_i$  and  $b_i$  as eigenvectors and eigenvalues of the Hessian  $H_q$ . This step leads to the nearest stationary point on the PES. This stationary point can be a minimum or a saddle point, according to the curvature of the PES at the actual point.

With a simple shift of the Hessian eigenvalues  $b_i$  in this equation one can guide the step to a stationary point with the required characteristics (Hessian with exactly one negative eigenvalue). The transition state search is separated into two different optimization problems. The energy is maximized along one Hessian eigenmode and minimized along the remaining  $3N - 7(6)$  eigenmodes. We introduce two different shift parameters  $\lambda_p$  and  $\lambda_n$ , where  $\lambda_p$  is the shift parameter for the eigenmode being maximized and  $\lambda_n$  shifts the Hessian eigenvalues of the modes being minimized.

This method allows us to maximize along any mode, not only the one with smallest eigenvalue. Starting from two different RFO-matrices for the different optimization problems (see description above) we get for  $\lambda_p$  and  $\lambda_n$ :

$$\lambda_p = \frac{1}{2}b_k + \frac{1}{2}\sqrt{b_k^2 + 4F_k^2} \quad \text{and} \quad \sum_{i \neq k} \frac{F_i^2}{\lambda_n - b_i} = \lambda_n \quad (4.18)$$

whereas  $F_i = V_i^+ g$  is the component of  $g$  along the Hessian eigenmode  $V_i$  and  $\lambda_n$  has to get solved iteratively. The solution for  $\lambda_n$  has to be negative and lower than  $b_2$  (or lower than  $b_1$ , if not the lowest mode is being followed). If the Hessian has more than one negative eigenvalue, these properties might not be fulfilled, and the Hessian would have to be modified. In our implementation the Hessian diagonal elements are either shifted or reversed in such a case.

Once the shift parameters are known the P-RFO step  $h$  is calculated as follows:

$$\Delta q_k = -\frac{\bar{F}_k V_k}{b_k - \lambda_p} \quad \text{and} \quad \Delta q_i = -\frac{\bar{F}_i V_i}{b_i - \lambda_n} \quad \text{with} \quad i = 1 \dots n, \quad i \neq k \quad (4.19)$$

$$\Delta q = \sum_{j=1}^n \Delta q_j \quad (4.20)$$

### 4.3.2 Introduction to Transition State Searches

If you provide a good estimate for the structure of the transition state (TS) structure, then you can find the respective transition state with the following keywords OptTS and ScanTS (in this example we take the structure with highest energy of the above relaxed surface scan):

```
! B3LYP SV(P) TightSCF SlowConv OptTS
# performs a TS optimization with the EF-algorithm
# Transition state: H-atom abstraction from CH4 to OH-radical

%geom
  Calc_Hess true # calculation of the exact Hessian
                  # before the first optimization step
end

* int 0 2
  C 0 0 0 0.000000 0.000 0.000
  H 1 0 0 1.285714 0.000 0.000
  H 1 2 0 1.100174 107.375 0.000
  H 1 2 3 1.100975 103.353 119.612
  H 1 2 3 1.100756 105.481 238.889
  O 2 1 3 1.244156 169.257 17.024
  H 6 2 1 0.980342 100.836 10.515
*
```

#### Note

- You need a good guess of the TS structure. Relaxed surface scans can help in almost all cases (see also example above).
- For TS optimization (in contrast to geometry optimization) an exact Hessian, a Hybrid Hessian or a modification of selected second derivatives is necessary.
- Analytic Hessian evaluation is available for HF and SCF methods, including the RI and RIJCOSX approximations and canonical MP2.
- Check the eigenmodes of the optimized structure for the eigenmode with a single imaginary frequency. You can also visualize this eigenmode with `orca_pltvib` (section [Animation of Vibrational Modes](#)) or any other visualization program that reads ORCA output files.

- If the Hessian is calculated during the TS optimization, it is stored as `basename.001.hess`, if it is recalculated several times, then the subsequently calculated Hessians are stored as `basename.002.hess`, `basename.003.hess`, ...
- If you are using the Hybrid Hessian, then you have to check carefully at the beginning of the TS optimization (after the first three to five cycles) whether the algorithm is following the correct mode (see TIP below). If this is not the case you can use the same Hybrid Hessian again via the `inhess read` keyword and try to target a different mode (via the `TS_Mode` keyword, see below).

In the example above the TS mode is of local nature. In such a case you can directly combine the relaxed surface scan with the TS optimization with the

```
! ScanTS
```

command, as used in the following example:

```
! B3LYP SV(P) TightSCF SlowConv
! ScanTS
# perform a relaxed surface scan and TS optimization
# in one calculation
%geom
  scan
    B 1 0 = 2.0, 1.0, 15
  end
end
* int 0 2
C   0   0   0   0.000000   0.000   0.000
H   1   0   0   1.999962   0.000   0.000
H   1   2   0   1.095870  100.445   0.000
H   1   2   3   1.095971   90.180  119.467
H   1   2   3   1.095530   95.161  238.880
O   2   1   3   0.984205  164.404   27.073
H   6   2   1   0.972562  103.807   10.843
*
```

#### Note

- The algorithm performs the relaxed surface scan, aborts the Scan after the maximum is surmounted, chooses the optimized structure with highest energy, calculates the second derivative of the scanned coordinate and finally performs a TS optimization.
- If you do not want the scan to be aborted after the highest point has been reached but be carried out up to the last point, then you have to type:

```
%geom
  fullScan true # do not abort the scan with !ScanTS
end
```

As transition state finder we implemented the quasi-Newton like Hessian mode following algorithm.[531, 533, 534, 535, 536, 537, 538, 540, 541] This algorithm maximizes the energy with respect to one (usually the lowest) eigenmode and minimizes with respect to the remaining  $3N - 7(6)$  eigenmodes of the Hessian.

#### Tip

- You can check at an early stage if the optimization will lead to the “correct” transition state. After the first optimization step you find the following output for the redundant internal coordinates:

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

(Angstroem and degrees)									
Definition				Value		dE/dq	Step	New-Value	comp. (TS_
↪mode)									
-----									
1.	B(H	1,C	0)		1.2857	0.013136	0.0286	1.3143	0.
↪58									
2.	B(H	2,C	0)		1.1002	0.014201	-0.0220	1.0782	
3.	B(H	3,C	0)		1.1010	0.014753	-0.0230	1.0779	
4.	B(H	4,C	0)		1.1008	0.014842	-0.0229	1.0779	
5.	B(O	5,H	1)		1.2442	-0.015421	-0.0488	1.1954	0.
↪80									
6.	B(H	6,O	5)		0.9803	0.025828	-0.0289	0.9514	
7.	A(H	1,C	0,H	2)	107.38	-0.001418	-0.88	106.49	
8.	A(H	1,C	0,H	4)	105.48	-0.002209	-0.46	105.02	
9.	A(H	1,C	0,H	3)	103.35	-0.003406	0.08	103.43	
10.	A(H	2,C	0,H	4)	113.30	0.001833	0.35	113.65	
11.	A(H	3,C	0,H	4)	113.38	0.002116	0.26	113.64	
12.	A(H	2,C	0,H	3)	112.95	0.001923	0.45	113.40	
13.	A(C	0,H	1,O	5)	169.26	-0.002089	4.30	173.56	
14.	A(H	1,O	5,H	6)	100.84	0.003097	-1.41	99.43	
15.	D(O	5,H	1,C	0,H	2)	17.02	0.000135	0.24	17.26
16.	D(O	5,H	1,C	0,H	4)	-104.09	-0.000100	0.52	-103.57
17.	D(O	5,H	1,C	0,H	3)	136.64	0.000004	0.39	137.03
18.	D(H	6,O	5,H	1,C	0)	10.52	0.000078	-0.72	9.79
-----									

Every Hessian eigenmode can be represented by a linear combination of the redundant internal coordinates. In the last column of this list the internal coordinates, that represent a big part of the mode which is followed uphill, are labelled. The numbers reflect their magnitude in the TS eigenvector (fraction of this internal coordinate in the linear combination of the eigenvector of the TS mode). Thus at this point you can already check whether your TS optimization is following the right mode (which is the case in our example, since we are interested in the abstraction of H1 from C0 by O5).

- If you want the algorithm to follow a different mode than the one with lowest eigenvalue, you can either choose the number of the mode or you can give an internal coordinate that should be strongly involved in this mode using the TS\_Mode keyword.

#### Tip

- If you look for a TS of a breaking bond the respective internal coordinate might not be included in the list of redundant internal coordinates due to the bond distance being slightly too large, leading to slow or even no convergence at all. In order to prevent that behavior a region of atoms that are active in the TS search can be defined, consisting of e.g. the two atoms of the breaking bond. During the automatic generation of the internal coordinates the bond radii of these atoms (and their neighbouring atoms) are increased, making it more probable that breaking or forming bonds in the TS are detected as bonds. See keywords TS\_Active\_Atoms and TS\_Active\_Atoms\_Factor in the keywords list below.

## Hessians for Transition State Calculations

For transition state (TS) optimization a simple initial Hessian, which is used for minimization, is not sufficient. In a TS optimization we are looking for a first order saddle point, and thus for a point on the PES where the curvature is negative in the direction of the TS mode (the TS mode is also called transition state vector, the only eigenvector of the Hessian at the TS geometry with a negative eigenvalue). Starting from an initial guess structure the algorithm used in the ORCA TS optimization has to climb uphill with respect to the TS mode, which means that the starting structure has to be near the TS and the initial Hessian has to account for the negative curvature of the PES at that point. The simple force-field Hessians cannot account for this, since they only know harmonic potentials and thus positive curvature.

The most straightforward option in this case would be (after having looked for a promising initial guess structure with the help of a relaxed surface scan) to calculate the exact Hessian before starting the TS optimization. With this Hessian (depending on the quality of the initial guess structure) we know the TS eigenvector with its negative eigenvalue and we have also calculated the exact force constants for all other eigenmodes (which should have positive force constants). For the HF, DFT methods and MP2, the analytic Hessian evaluation is available and is the best choice, for details see section Frequencies (*Vibrational Frequencies*).

When only the gradients are available (most notably the CASSCF), the numerical calculation of the exact Hessian is very time consuming, and one could ask if it is really necessary to calculate the full exact Hessian since the only special thing (compared to the simple force-field Hessians) that we need is the TS mode with a negative eigenvalue.

Here ORCA provides two different possibilities to speed up the Hessian calculation, depending on the nature of the TS mode: the Hybrid Hessian and the calculation of the Hessian value of an internal coordinate. For both possibilities the initial Hessian is based on a force-field Hessian and only parts of it are calculated exactly. If the TS mode is of very local nature, which would be the case when e.g. cleaving or forming a bond, then the exactly calculated part of the Hessian can be the second derivative of only one internal coordinate, the one which is supposed to make up the TS mode (the formed or cleaved bond). If the TS mode is more complicated and more delocalized, as e.g. in a concerted proton transfer reaction, then the hybrid Hessian, a Hessian matrix in which the numerical second derivatives are calculated only for those atoms, which are involved in the TS mode (for more details, see section *Geometry Optimizations*), should be sufficient. If you are dealing with more complicated cases where these two approaches do not succeed, then you still have the possibility to start the TS optimization with a full exact Hessian.

Numerical Frequency calculations are quite expensive. You can first calculate the Hessian at a lower level of theory or with a smaller basis set and use this Hessian as input for a subsequent TS optimization:

```
%geom
  inhess    Read          # this command comes with:
  InHessName "yourHessian.hess" # filename of Hessian input file
end
```

Another possibility to save computational time is to calculate exact Hessian values only for those atoms which are crucial for the TS optimization and to use approximate Hessian values for the rest. This option is very useful for big systems, where only a small part of the molecule changes its geometry during the transition and hence the information of the full exact Hessian is not necessary. With this option the coupling of the selected atoms are calculated exactly and the remaining Hessian matrix is filled up with a model initial Hessian:

```
%geom
  Calc_Hess true
  Hybrid_Hess {0 1 5 6} end # calculates a Hybrid Hessian with
                             # exact calculation for atoms 0, 1, 5 and 6
end
```

For some molecules the PES near the TS can be very far from ideal for a Newton-Raphson step. In such a case ORCA can recalculate the Hessian after a number of steps:

```
%geom
  Recalc_Hess 5 # calculate the Hessian at the beginning
                # and recalculate it after 5,10,15,... steps
end
```

Another solution in that case is to switch on the trust radius update, which reduces the step size if the Newton-Raphson steps behave unexpected and ensures bigger step size if the PES seems to be quite quadratic:

```
%geom
  Trust 0.3 # Trust <0 - use fixed trust radius (default: -0.3 au)
           # Trust >0 - use trust radius update, i.e. 0.3 means:
           # start with trust radius 0.3 and use trust radius update
end
```

## Special Coordinates for Transition State Optimizations

- If you look for a TS of a breaking bond the respective internal coordinate might not be included in the list of redundant internal coordinates (but this would accelerate the convergence). In such a case (and of course in others) you can add coordinates to or remove them from the set of autogenerated redundant internal coordinates (alternatively check the TS\_Active\_Atoms keyword):

```
# add ( A ) or remove ( R ) internal coordinates
%geom
  modify_internal
    { B 10 0 A }      # add a bond between atoms 0 and 10
    { A 8 9 10 R }    # remove the angle defined
                      # by atoms 8, 9 and 10
    { D 7 8 9 10 R }  # remove the dihedral angle defined
                      # by atoms 7, 8, 9 and 10
  end
end
```

### 4.3.3 ScanTS option

For TS modes of rather local nature (involving only one bond or an angle; no concerted movements over multiple atoms) we implemented the ScanTS feature. Here the user can carry out a relaxed surface scan and a TS optimization in one calculation. After the relaxed surface scan the algorithm chooses the optimized structure of the scan with highest energy as initial guess structure and the two neighbouring structures for the calculation of the second derivative of the scanned coordinate (e.g., if scan step number 4 gives the structure with highest energy, then structure `basename.004.xyz` is the initial guess for the TS optimization; the structures `basename.003.xyz` and `basename.005.xyz` are used for the calculation of the second derivative). Before the first step of the subsequent TS optimization the energies and gradients for all three structures are calculated. The gradients are then transformed to internal coordinates. The diagonal Hessian value of the scanned coordinate is then calculated via finite difference of the internal gradients of the two given structures (003 and 005 in our example).

For the construction of the initial Hessian a model force field Hessian is built up (this Hessian has got only diagonal entries and zeros as off-diagonal elements). The exactly calculated diagonal Hessian value replaces the model force field Hessian entry for the respective internal coordinate.

If the user already performed a regular relaxed surface scan without the subsequent TS optimization, then he can nevertheless use these structures for the same procedure. A relaxed surface scan always gives you the xyz-files and gbw-files of the optimized structures of each scan step. A separate TS optimization can be carried out where the structure with highest energy is the starting structure. Additionally the two files with the two adjacent structures (as explained above) have to be provided (via the Hess\_Internal keyword, see below). Furthermore, the internal coordinate, for which the diagonal Hessian value has to be calculated, has to be given (the previously scanned coordinate). This exact Hessian calculation is only possible for one internal coordinate:

```
%geom
  Hess_Internal
    { B 1 0 C }      # previously scanned coordinate
    XYZ1 "ScanName.003.xyz" # the xyz-files of the structures
    XYZ2 "ScanName.005.xyz" # next to the highest energy point
    GBW1 "ScanName.003.gbw" # the gbw-files of the structures
    GBW2 "ScanName.005.gbw" # next to the highest energy
```

(continues on next page)

(continued from previous page)

```

                                # the gbw-files are optional
    end
end

```

Additionally the manipulation of the diagonal Hessian values of the internal Hessian is possible for further internal coordinates, but without an extra calculation. Here the user can just define a value (in Eh/Bohr<sup>2</sup>).

```

Hess_Internal
  {A 3 2 1 D 2.0} # define a diagonal Hessian value of
                  # 2 Eh/Bohr2 for the angle between
                  # atoms 3 2 1
  {B 1 0 D -0.5} # define a diagonal Hessian value of
                  # -0.5 Eh/Bohr2 for the bond between
                  # atoms 1 and 0
end

```

The definition of such Hessian (diagonal) elements is possible for multiple internal coordinates. These just replace the values of the force field model Hessian.

### 4.3.4 Hybrid Hessian

We implemented the calculation of a “Hybrid Hessian” as an alternative to the full Hessian calculation for TS optimization. Here only those parts of the Hessian, that are important for the TS optimization, are calculated exactly. For this calculation we define two kinds of atoms: atoms whose couplings with the other atoms are treated exactly (E) and atoms whose couplings are treated approximately (A).

In a first step an Almloef model Hessian is built up in redundant internal coordinates and transformed to Cartesian coordinates. This Hessian gives the second derivative elements for atom pairs A/A. In a second step the second derivative elements between pairs E/E and E/A are calculated numerically as in a numerical frequency calculation:

$$\frac{\partial^2 E}{\partial i_B \partial j_C} = \frac{\partial^2 E}{\partial j_C \partial i_B} = \frac{g_{j,C}^{i,B} - g_{j,C}^{eq}}{displ.} \quad (4.21)$$

with:

$i, j$	x-, y- or z-direction
$B, C$	pairs of E/E, E/A, A/E
$displ.$	magnitude of displacement
$g_{j,C}^{eq}$	force on atom $C$ in direction $j$ in current geometry
$g_{j,C}^{i,B}$	force on atom $C$ in direction $j$ after displacement of atom $B$ in direction $i$

### 4.3.5 Partial Hessian Vibrational Analysis

We implemented the Partial Hessian Vibrational Analysis (PHVA), as published by Li, Jensen in [531], for the analysis of the nature of stationary points of structures obtained with QM/MM optimizations.

```

# PHVA after a QM/MM optimization in the (dispersion-/PC-) field
# caused by the MM-atoms
! NumFreq
%LJCoefficients "temp.LJ" # file with the Lennard Jones
                        # coefficients for dispersion interaction
                        # obtained from last QM/MM run
%pointcharges "temp.pc" # file with the point charges for

```

(continues on next page)

(continued from previous page)

```

# electrostatic interaction
# obtained from last QM/MM run
#
%freq
  PARTIAL_Hess {0 1 2} # atoms which are "frozen" and which make
                        # the boundary to the MM-system
  end
end

```

**Note**

- This procedure should be used for QM/MM optimized structures only to verify the nature of the stationary point and have an estimate of the ZPE.

Here we shortly describe the procedure: In PHVA we divide the system into two parts  $B$  (of size  $n$  atoms) and  $C$  (size  $N - n$ ). Let the atom set  $B$  belong to the region where the chemical change is localized. The Partial Hessian matrix is built up as follows:

$$\begin{pmatrix} K_{BB} & 0 \\ 0 & K_{CC}^\varepsilon \end{pmatrix} \quad (4.22)$$

With:

$$K_{BB} : x, y, z \text{ direction}$$

$$K_{CC}^\varepsilon = \begin{pmatrix} \varepsilon & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \varepsilon \end{pmatrix}, \varepsilon = 10^{-8} \text{ au}, \quad (4.23)$$

this corresponds to using near-infinite masses for the atoms in  $C$ .

With this procedure we get the following eigenvalue structure:

- Six zero eigenvalues with modes corresponding to translational and rotational motion of the entire molecule.
- $3(N - n) - 6$  small (less than  $1 \text{ cm}^{-1}$ ) eigenvalues with modes corresponding mainly to internal motion within region  $C$ .
- Three eigenvalues (typically less than  $10 \text{ cm}^{-1}$ ) with modes corresponding mainly to motion of region  $C$  relative to region  $B$ .
- $(3n - 3)$  eigenvalues with modes corresponding mainly to relative motion of  $B$  and  $C$  as well as internal motion within region  $B$ .

### 4.3.6 Transition State Theory Keywords

Table 4.5: %geom block input keywords and options for transition state geometry optimization

Keyword	Option / Value	Description
OptTS		Transition structure optimization. Included as !OptTS
Scan		Must include !ScanTS for transition state scan.
	B 1 2 = v1, v2, N	Scan atom 1 to atom 2 bond distance from v1 to v2 in N steps
	B 1 2 [v1 v2 ...]	Scan atom 1 to atom 2 bond with custom step values.
	A 1 2 3	Scan an angle using atoms 1,2,3.
	D 1 2 3 4	Scan a dihedral of atoms 1,2,3,4.
Simul_Scan	true	Enable simultaneous multidimensional scans (up to 3 coordinates)
FullScan	true	Forces full surface scan before TS optimization (!ScanTS)



Table 4.5 – continued from previous page

Keyword	Option / Value	Description
TS_search	EF	Enable TS search using Eigenvector Following.
TS_Mode	{M 0}	Follow eigenvector 0 (lowest mode).
	{B 0 1}	Follow bond between atoms 0 and 1.
	{A 2 1 0}	Follow angle between atoms 2, 1, and 0.
	{D 3 2 1 0}	Follow dihedral between atoms 3, 2, 1, and 0.
TS_Active_Atoms	{ 1 2 3 }	Atoms that are involved in TS, e.g. for proton transfer the
TS_Active_Atoms_Factor	1.5	Factor by which the cutoff for bonds is increased for the al
inhess	unit	Start optimization with a unit matrix Hessian.
InHessName	Read	Read Hessian from external file. Requires InHessName.
	"filename.hess"	Specifies file containing the Hessian (e.g., .hess, .opt,
	Lindh	Use Lindh's model Hessian (only for redundant coordinate
	Almloef	Use Almloef's model Hessian.
	Schlegel	Use Schlegel's model Hessian.
	Swart	Use Swart and Bickelhaupt's model Hessian.
	XTB0	Use GFN0-xTB Hessian.
	XTB1	Use GFN1-xTB Hessian.
	XTB2	Use GFN2-xTB Hessian.
	GFNFF	Use GFN-FF Hessian.
Calc_Hess	true	Calculate numerical Hessian at the beginning of optimizati
Recalc_Hess	5	Recalculate Hessian every 5 cycles.
Hybrid_Hess	{0 1 5 6} end	Calculate exact Hessian for specified atoms, use model for
NumHess	true	Request numerical Hessian.
Hess_Internal	{A 3 2 1 D 2.0}	Manually set Hessian value ( $E_h/\text{Bohr}^2$ ) for angle between a
	reset 5	Reset modified Hessian values after 5 cycles.
	{B 1 0 C}	Coordinates for Hessian refinement after a relaxed surface
	XYZ1 "ScanName.003.xyz"	Structure file before maximum in scan.
	XYZ2 "ScanName.005.xyz"	Structure file after maximum in scan.
	GBW1 "ScanName.003.gbw"	Optional GBW file before max in scan.
	GBW2 "ScanName.005.xyz"	Optional GBW file after max in scan.
Update	Powell	Use Powell update for Hessian.
	Bofill	Use Bofill update (default for TS optimization).
	BFGS	Use BFGS update (default for geometry optimization).
HESS_Modification	Shift_Diag	Shift diagonal elements of Hessian (default).
	EV_Reverse	Reverse sign of diagonal elements.
HESS_MinEV	0.0001	Minimum allowed Hessian eigenvalue.
NResetHess	20	Rebuild model Hessian after 20 steps (BFGS only).
NStepsInResetHess	5	Number of previous steps used to inform new model Hessi
Trust	-0.3	Use fixed trust radius of 0.3 au (negative sign indicates fixe
modify_internal	{B 10 0 A}	Add bond between atoms 0 and 10.
	{A 8 9 10 R}	Remove angle between atoms 8, 9, and 10.
	{D 7 8 9 10 R}	Remove dihedral between atoms 7, 8, 9, and 10.

## 4.4 Intrinsic Reaction Coordinate

The Intrinsic Reaction Coordinate (IRC) is a special form of a minimum energy path, connecting a transition state (TS) with its downhill-nearest intermediates. A method determining the IRC is thus useful to determine whether a transition state is directly connected to a given reactant and/or a product. The implementation in ORCA follows the method suggested by Morokuma and coworkers.[542]

The IRC method follows the gradient of the nuclear coordinates. As the gradient is negligible at a TS, first an initial displacement from the TS structure has to be carried out, based on the eigenmodes of the Hessian, in order to get to a region with a non-negligible gradient. For the initial displacement the eigenvector of the eigenmode with the lowest frequency (hessMode=0) is normalized and then scaled by Scale\_Displ\_SD (which by default is chosen such that an

energy change of Init\_Displ\_DE can be expected). Two initial displacements, forward and backward, are taken by adding the resulting displacement vector (multiplied with +1 and -1, respectively) to the initial structure. If the user requests the downhill direction (e.g. from a previous unconverged IRC run), it is assumed that the gradient is nonzero and thus no initial displacement is carried out.

After the initial displacement, the iterations of the IRC method begin. Each iteration consists of two main steps, each of which consists again of multiple SP and gradient runs:

1. Initial steepest descent (SD) step:
  1. The gradient (grad0) of the starting geometry (G0) is normalized, scaled by Scale\_Displ\_SD, and the resulting displacement vector (SD1) is applied to G0.
  2. Optional (if SD\_ParabolicFit is true): If SD1 increases the energy, a linear search is taken along the direction of the displacement vector:
    1. The displacement vector SD1 is scaled by 0.5 ( $SD2 = 0.5 \times SD1$ ) and again added to G0.
    2. A parabolic fit for finding the displacement vector (SD3) which leads to minimal energy is carried out using the three SP energies (G0, geometry after SD1 and after SD2 step). SD3 has the same direction as SD1 and SD2, but can have a different length.
    3. The keyword Interpolate\_only controls whether the length of SD3 has to be in between 0 and the length of SD1. If that is the case, the maximum length is determined by SD1, the minimum length is zero.
  3. At the resulting geometry G1 (G0+SD1 or G0+SD3) the gradient is calculated (grad1).
2. Optional (if Do\_SD\_Corr is true): Correction to the steepest descent step:
  1. Based on grad0 and grad1 a vector is computed which represents a correction to the first SD (SD1 or SD3) step. This correction brings the geometry closer to the IRC.
  2. This vector is normalized, scaled by Scale\_Displ\_SD\_Corr times the length of SD1 or SD3, and the resulting displacement vector (SDC1) is applied to G1.
  3. Optional (if SD\_Corr\_ParabolicFit is true):
    1. If the energy increases after applying step SDC1, SDC1 is scaled by 0.5 ( $SDC2 = 0.5 \times SDC1$ ), if the energy decreases, SDC1 is scaled by 2 ( $SDC2 = 2 \times SDC1$ ). SDC2 is then added to G1.
    2. A parabolic fit for finding the displacement vector (SDC3) which leads to minimal energy is carried out using the three SP energies (G1, geometry after SDC1 and after SDC2 step). SDC3 has the same direction as SDC1 and SDC2, but can have a different length.
    3. The keyword Interpolate\_only controls whether the length of SDC3 has to be in between 0 and the length of SDC1. If that is the case, the maximum length is determined by SDC1, the minimum length is zero.
  4. At the resulting geometry G2 (G1+SDC1 or G1+SDC3) the gradient is calculated (grad2).
3. The gradient at the new geometry is checked for convergence.
4. Optional (if Adapt\_Scale\_Displ is true):
  1. If the resulting overall step size is smaller than 0.5 times Scale\_Displ\_SD, Scale\_Displ\_SD is multiplied by 0.5.
  2. If the resulting overall step size is larger than 2 times Scale\_Displ\_SD, Scale\_Displ\_SD is multiplied by 2.
  3. Scale\_Displ\_SD may not become smaller than 1/16 times the initial Scale\_Displ\_SD and not larger than 4 times the initial Scale\_Displ\_SD.

### 4.4.1 Basic Usage

The following keywords are available:

```
! B3LYP SV(P) TightSCF KDIIS SOSCF Freq IRC
* xyz 0 2
C -0.000 0.001 -0.000
H 1.290 0.005 -0.006
H -0.330 1.050 -0.002
H -0.252 -0.532 -0.929
H -0.286 -0.545 0.911
O 2.499 0.220 0.065
H 2.509 1.085 0.525
*
```

#### **Note**

- The same method and basis set as used for optimization and frequency calculation should be used for the IRC run.
- The IRC keyword can be requested without, but also together with OptTS, ScanTS, NEB-TS, AnFreq and NumFreq keywords.
- In its default settings the IRC code checks whether a Hessian was computed before the IRC run. If that is not the case, and if no Hessian is defined via the %irc block, a new Hessian is computed at the beginning of the IRC run.
- A final trajectory (\_IRC\_Full\_trj.xyz) is generated which contains both directions, forward and backward, by starting from one endpoint and going to the other endpoint, visualizing the entire IRC. Forward (\_IRC\_F\_trj.xyz and \_IRC\_F.xyz) and backward (\_IRC\_B\_trj.xyz and \_IRC\_B.xyz) trajectories and xyz files contain the IRC and the last geometry of that respective run.

### 4.4.2 Keywords

```
! IRC
%irc
  MaxIter      20
  PrintLevel 1
  Direction    both # both - default
                  # forward
                  # backward
                  # down
# Initial displacement
  InitHess     read # by default ORCA uses the Hessian from AnFreq or NumFreq, or
                  # computes a new one
                  # read      - reads the Hessian that is defined via Hess_Filename
                  # calc_anfreq - computes the analytic Hessian
                  # calc_numfreq - computes the numeric Hessian
  Hess_Filename "h2o.hess" # input Hessian for initial displacement, must be used
                          # together with InitHess = read
  hessMode     0 # Hessian mode that is used for the initial displacement. Default_
↳ 0
  Init_Displ_DE # DE (default) - energy difference
                  # length      - step size
  Scale_Init_Displ 0.1 # step size for initial displacement from TS. Default 0.1_
↳ a.u.
  DE_Init_Displ 2.0 # energy difference that is expected for initial_
↳ displacement
                  # based on provided Hessian (Default: 2 mEh)
```

(continues on next page)

(continued from previous page)

```

# Steps
Follow_CoordType cartesian # default and only option
Scale_Displ_SD      0.15 # Scaling factor for scaling the 1st SD step
Adapt_Scale_Displ   true  # modify Scale_Displ_SD when the step size becomes
                          # smaller or larger
SD_ParabolicFit      true  # Do a parabolic fit for finding an optimal SD step
                          # length
Interpolate_only     true  # Only allow interpolation for parabolic fit, not
                          # extrapolation
Do_SD_Corr           true  # Apply a correction to the 1st SD step
Scale_Displ_SD_Corr  0.333 # Scaling factor for scaling the correction step to
                          # the SD step. It is multiplied by the length of the
                          # final 1st SD step
SD_Corr_ParabolicFit true  # Do a parabolic fit for finding an optimal_
↪correction
                          # step length
# Convergence thresholds - similar to LooseOpt
TolRMSG  5.e-4      # RMS gradient (a.u.)
TolMaxG   2.e-3      # Max. element of gradient (a.u.)
# Output options
Monitor_Internals # Up to three internal coordinates can be defined
  {B 0 1}          # for which the values are printed during the IRC run.
  {B 1 5}          # Possible are (B)onds, (A)ngles, (D)ihedrals and (I)mpropers
end
end

```

## NOTE

- For direction=down (downhill) no initial displacement is necessary, and thus no Hessian is needed.

## 4.5 Nudged Elastic Band Method

The Nudged Elastic Band (NEB) method is used to find a minimum energy path (MEP) connecting given reactant and product state minima on the energy surface. An initial path is generated and represented by a discrete set of configurations of the atoms, referred to as images of the system. The number of images is specified by the user and has to be large enough to obtain sufficient resolution of the path. The implementation in ORCA is described in detail in the article by Ásgeirsson et. al.[543] and in section *Nudged Elastic Band Method* along with the input options. The most common use of the NEB method is to find the highest energy saddle point on the potential energy surface specifying the transition state for a given initial and final state. Rigorous convergence to a first-order saddle point can be obtained with the climbing image NEB (CI-NEB), where the highest energy image is pushed uphill in energy along the tangent to the path while relaxing downhill in orthogonal directions. Another method for finding a first-order saddle point is the NEB-TS which uses the CI-NEB method with a loose tolerance to begin with and then switches over to the OptTS method to converge on the saddle point. This combination can be a good choice for calculations of complex reactions where the ScanTS method fails or where 2D relaxed surface scans are necessary to find a good initial guess structure for the OptTS method. The zoomNEB variants are a good choice in case of very complex transition states with long tails. For more detailed information on the various NEB variants implemented in ORCA please consult section *Nudged Elastic Band Method*.

In their simplest form NEB, NEB-CI and NEB-TS only require the reactant and product state configurations (one via the xyz block, and the other one via the keyword neb\_end\_xyzfile):

```

!NEB-TS # or !NEB or !NEB-CI or !ZOOM-NEB-TS or !ZOOM-NEB-CI
      # or !Fast-NEB-TS (corresponds to IDPP-TS defined in the NEB-TS_
↪manuscript)
      # or !Loose-NEB-TS (corresponds to default NEB-TS in the NEB-TS manuscript)
%neb
  neb_end_xyzfile "final.xyz"
end

```

Below is an example of an NEB-TS run involving an intramolecular proton transfer within acetic acid. The simplest input is

```
!XTB NEB-TS
%neb
  neb_end_xyzfile "final.xyz"
end

*xyz 0 1
C      0.416168      0.038758      -0.014077
C      0.041816      0.011798       1.439610
O      1.524458      0.176600      -0.453888
O      -0.654209     -0.127881      -0.803857
H      -0.391037     -0.126036      -1.737478
H      -0.913438      0.507022      1.585301
H      -0.057787     -1.026455      1.750845
H      0.819515      0.485425      2.030252
*
```

Where the final.xyz structure contains the corresponding structure with the proton on the other oxygen.

The initial path is reasonable and the CI calculation can be switched on after five NEB iterations.

Starting iterations:

Optim.	Iteration	HEI	E(HEI)-E(0)	max( Fp )	RMS(Fp)	dS
Switch-on CI threshold						
LBFGS	0	4	0.081017	0.073897	0.018915	3.2882
LBFGS	1	5	0.070244	0.056668	0.013913	3.2770
LBFGS	2	5	0.062934	0.038972	0.008763	3.3376
LBFGS	3	5	0.057358	0.032076	0.006535	3.3950
LBFGS	4	4	0.053260	0.019015	0.003599	3.4826

Image 4 will be converted to a climbing image in the next iteration (max(|Fp|) < 0.0200)

Optim.	Iteration	CI	E(CI)-E(0)	max( Fp )	RMS(Fp)	dS	max( FCI )
Convergence thresholds							
				0.020000	0.010000		0.002000

The CI run converges after another couple of iterations:

```
*****H U R R A Y*****
***      THE NEB OPTIMIZATION HAS CONVERGED      ***
*****
```

Subsequently a summary of the MEP is printed:

```
-----
                        PATH SUMMARY
-----
All forces in Eh/Bohr.
```

Image	Dist. (Ang.)	E(Eh)	dE(kcal/mol)	max( Fp )	RMS(Fp)
0	0.000	-14.45993	0.00	0.00011	0.00004
1	0.426	-14.44891	6.91	0.00092	0.00033
2	0.652	-14.42864	19.63	0.00084	0.00038
3	0.805	-14.41132	30.50	0.00075	0.00027
4	0.932	-14.40562	34.08	0.00057	0.00018 <= CI
5	1.044	-14.41047	31.03	0.00057	0.00024
6	1.153	-14.42200	23.80	0.00103	0.00034
7	1.280	-14.43666	14.60	0.00098	0.00037

(continues on next page)

(continued from previous page)

8	1.476	-14.45106	5.56	0.00106	0.00033
9	1.869	-14.45988	0.03	0.00013	0.00006

Additionally, detailed information on the highest energy image (or the CI) is printed:

-----					
INFORMATION ABOUT SADDLE POINT					
-----					
Climbing image	....	4			
Energy	....	-14.40561577	Eh		
Max. abs. force	....	9.5976e-04	Eh/Bohr		
-----					
SADDLE POINT (ANGSTROEM)					
-----					
C	0.040867	0.007347	-0.497635		
C	-0.075595	0.017879	0.979075		
O	1.122340	0.126074	-1.145534		
O	-0.928470	-0.137946	-1.298318		
H	0.165808	-0.021676	-2.055704		
H	-0.996979	0.514720	1.271668		
H	-0.116377	-1.013504	1.327873		
H	0.788406	0.507105	1.418575		
-----					
FORCES (Eh/Bohr)					
-----					
C	-0.000646	-0.000111	0.000086		
...					
-----					
UNIT TANGENT					
-----					
C	-0.246569	-0.031821	-0.019359		
...					
=> Unit tangent is an approximation to the TS mode at the saddle point					
Next a TS optimization is performed on the CI from the NEB run.					

Finally, a TS optimization is started, after which the MEP information is updated by including the TS structure:

-----					
PATH SUMMARY FOR NEB-TS					
-----					
All forces in Eh/Bohr. Global forces for TS.					
Image	E (Eh)	dE (kcal/mol)	max( Fp )	RMS(Fp)	
0	-14.45993	0.00	0.00011	0.00004	
1	-14.44891	6.91	0.00092	0.00033	
2	-14.42864	19.63	0.00084	0.00038	
3	-14.41132	30.50	0.00075	0.00027	
4	-14.40562	34.08	0.00057	0.00018	<= CI
TS	-14.40562	34.08	0.00033	0.00013	<= TS
5	-14.41047	31.03	0.00057	0.00024	
6	-14.42200	23.80	0.00103	0.00034	
7	-14.43666	14.60	0.00098	0.00037	
8	-14.45106	5.56	0.00106	0.00033	
9	-14.45988	0.03	0.00013	0.00006	

Note that here both TS and CI are printed for comparison.

## 4.6 Nudged Elastic Band Method

The Nudged Elastic Band (NEB)[544, 545, 546] method is used to find a minimum energy path (MEP) connecting two local energy minima on the potential energy surface (PES) and thereby an estimate of the activation energy for the transition. The two minima are referred to as the reactant and product in the following discussion. The path can have one or more maxima, each one corresponding to a first order saddle point on the energy surface. The NEB method offers an advantage over eigenvector-following methods in that it is guaranteed to find saddle points that connect the given reactant and product states. The minimum energy path is often used to represent the reaction coordinate of the transition between the two states. The methods and implementation outlined below are discussed further in Ref. [543].

The user needs to specify the reactant and product configurations. The reactant energy minimum is inserted into the regular ORCA input file while the product should be in a separate .xyz file (keyword: *neb\_end\_xyzfile*). The reactant and product configurations should be optimized *a priori* by relaxing to energy minima on the PES, see section *Geometry Optimizations*. This can also be achieved via the ‘preopt\_ends’ keyword. It is important to carefully prepare the reactant and product such that the position (or index) of the atoms is the same in the two configuration files, i.e. there should be one-to-one mapping between the reactant and product configurations.

The discretized path between the two minima is represented by a set of  $M$  system configurations that are referred to as images, i.e.  $\mathbf{R} = [r_0, r_1, \dots, r_{M-1}]$ . The number of intermediate images between the end points (i.e. reactant and product) is specified by the user. The general rule of thumb is to include  $M = 7 - 12$ , or  $5 - 10$  intermediate images per energy maximum in order to obtain a high enough resolution of the path and the saddle point. However, calculations can often converge and give accurate results with fewer images but complex paths with multiple maxima or long tails may require more images. During an NEB calculation the intermediate images are iteratively shifted towards the MEP using the component of the atomic force that is perpendicular to the current path as estimated from the tangent to the path at each image, while the endpoint images are typically kept fixed. In each step of the iterative process, the energy and atomic forces of each intermediate image need to be computed. One of the main advantages of the NEB method is that the calculations of the images are carried out in parallel, where the electronic structure computations can be distributed over multiple processors (see discussion below for more details on the parallelization). While the CPU time is proportional to the number of images, the number of iterations needed for convergence to the MEP can become smaller when more images are included in the discrete representation of the path.

The tangent to the path at each image,  $\hat{\tau}_i$ , can be estimated in two ways (keyword: *tangent*), either by the original method [545] or by the more numerically stable improved [546] estimate (default option). In the former, the tangent at an image is taken to be a linear combination of the two line segments that connect to image  $i$ ,

$$\tau_i = \frac{r_{i+1} - r_i}{|r_{i+1} - r_i|} + \frac{r_i - r_{i-1}}{|r_i - r_{i-1}|}$$

In the improved tangent estimate, the line segment from the current image to the adjacent image with higher energy is used, i.e.

$$\tau_i = \begin{cases} \tau^+ & \text{if } E_{i+1} > E_i > E_{i-1} \\ \tau^- & \text{if } E_{i+1} < E_i < E_{i-1} \end{cases}$$

where  $\tau^+ = r_{i+1} - r_i$  and  $\tau^- = r_i - r_{i-1}$ . With the exception of image  $i$  being at an energy extremum along the path, then an energy-weighted average of the two line segments to adjacent images is used,

$$\tau_i = \begin{cases} \tau^+ \Delta V_i^{\max} + \tau^- \Delta V_i^{\min} & \text{if } E_{i+1} > E_{i-1} \\ \tau^+ \Delta V_i^{\min} + \tau^- \Delta V_i^{\max} & \text{if } E_{i+1} < E_{i-1} \end{cases}$$

where

$$\begin{aligned} \Delta V_i^{\max} &= \max(|E_{i+1} - E_i|, |E_{i-1} - E_i|) \\ \Delta V_i^{\min} &= \min(|E_{i+1} - E_i|, |E_{i-1} - E_i|) \end{aligned}$$

Then the tangent is normalized,  $\hat{\tau}_i = \tau_i / |\tau_i|$ . With an accurate estimate of the tangent, the perpendicular component of the atom force is computed by,

$$F_i^\perp = F_i - (F_i \cdot \hat{\tau}_i) \hat{\tau}_i$$

In the free-end NEB method, the endpoint images are optimized simultaneously along with the intermediate images, i.e.  $M$  movable images are used. Three different variants of the free-end NEB method (keywords: *free\_end*, *free\_end\_type*) are included in ORCA: where (i) the endpoints are restrained to move along the same (or separate) energy isocontour[547, 548], (ii) according to the atomic force acting perpendicular to the path and (iii) the full atomic force. For the first variant, the ‘reactant’ and ‘product’ endpoint images, can be restrained to move along two different energy contours,  $E_0$  (given by keyword: *free\_end\_ec*) and  $E_1$  (given by keyword: *free\_end\_ec\_end*), to keep the path bounded. Then, if the images drift away from the isocontour because of curvature of the energy surface, an harmonic restraint term is used to pull the image back to the contour.[548] The stiffness of the harmonic restraint is given by a user-supplied parameter (keyword: *free\_end\_kappa*). It is important to carefully select the energy values and the strength of the harmonic restraint, otherwise the path may become kinked. For the second variant, the endpoint images are unbounded but displaced directly towards the MEP. This is typically acceptable if the endpoint images are already in vicinity to the MEP and is less prone to kinks developing along the path. For the third variant, the endpoint images become optimized to the reactant and product energy minimum.

## 4.6.1 Spring forces

In order to control the distribution of the images along the path, spring forces are included to act between adjacent images, tangential to the path[545],

$$F_i^{\text{sp},\parallel} = (k_i^{\text{sp}}|r_{i+1} - r_i| - k_{i-1}^{\text{sp}}|r_i - r_{i-1}|) \hat{\tau}_i$$

The magnitude of the spring forces (i.e. stiffness) is controlled by spring constants,  $k^{\text{sp}}$ . The typical values of the spring constant (keyword: *springconst*) can be taken to be within the range of 0.01 Eh/Bohr<sup>2</sup> to 1.0 Eh/Bohr<sup>2</sup>. If the spring constants are chosen to be the same for all pairs of adjacent images, the images will be equally distributed along the path. However, it is also possible to choose energy-weighted spring constants (keyword: *energy\_weighted*) so as to increase the density of images in the higher energy regions[543, 549]. In an energy-weighted NEB calculation the spring constants are scaled according to the relative energy of the images, from a lower-bound value (keyword: *springconst*) to an upper-bound value (keyword: *springconst2*), by

$$k_i^{\text{sp}} = \begin{cases} (1 - \alpha_i)k_u + \alpha_i k_l, & \text{if } E_i > E_{\text{ref}} \\ k_l, & \text{otherwise} \end{cases}$$

$$\alpha_i = \frac{E_{\text{max}} - E_i}{E_{\text{max}} - E_{\text{ref}}}$$

where  $k_u$  and  $k_l$  are the upper- and lower-bound values for the spring constant.  $E_{\text{max}}$  is the current estimate of the maximum energy along the path,  $E_i$  is the higher energy image of the pair of images connected by line segment  $i$  and  $E_{\text{ref}}$  is the energy of the higher energy minimum. The energy-weighted springs will typically serve to improve the tangent estimate in the barrier region and hence stabilize the calculations. The inclusion of energy-weighted springs can be important in reactions where the energy barrier is narrow and/or the pathway is characterized by a long ‘energy tail’, e.g., in rearrangements or dissociation reactions. The choice of spring constants will affect the behavior of a calculation, especially the number of iterations needed to reach convergence. Other formulations for spring forces are also available since ORCA 4.2 (keyword: *springtype*). These are referred to as the original[545] and ideal[550] spring forces. The original spring forces are estimated by a spring acting on each degree of freedom between a pair of adjacent images. For ideal springs each image is assigned an ideal position along the path based on a linear interpolation of the current location of the images and the individual images interact with the ideal locations via spring forces. The ideal springs are currently not implemented to work with energy weighted spring constants.

While only the component of the spring force parallel to the path is included in an NEB calculation (by default) the user can choose to include a fraction of the spring force acting perpendicular to the path to stabilize calculations (keyword: *perpspring*). Since, the perpendicular component of the spring force can serve to straighten out the path and is useful for complex pathways with multiple energy extrema and low resolution of the path. The inclusion of the perpendicular component of the spring force is always accompanied by a switching function that is used to scale it according to (i) convergence to the MEP (the ‘tan’ function)[551] (ii) the angle between adjacent images (the ‘cos’ function)[545] or a combination of both. The inclusion of the perpendicular spring force can help to reduce the number of iterations and eliminate kinks on the path. However, it is important to stress that by inclusion of the perpendicular spring forces, the images may not converge on the MEP. Alternatively, the user can choose to use the modified DNEB method[551, 552].



### 4.6.2 Optimization and convergence of the NEB method

The effective force used in a standard NEB calculations is the sum of the atomic force component perpendicular to the path and the spring force component parallel to the path,

$$F_i^{\text{NEB}} = F_i^{\perp} + F_i^{\text{sp},\parallel}$$

The path can be brought to the MEP by moving according to the effective force, as is shown in Fig. 4.2.

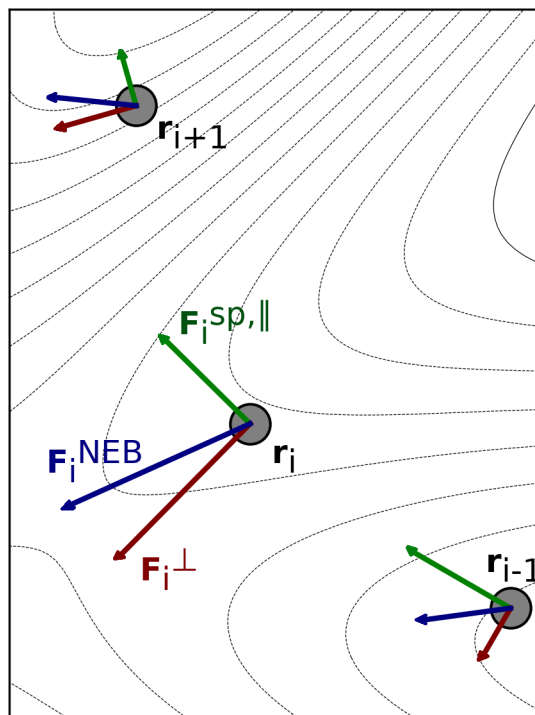


Fig. 4.2: Visualization of the effective force,  $F_i^{\text{NEB}}$  and its two components:  $F_i^{\perp}$  and  $F_i^{\text{sp},\parallel}$  for three images along an intermediate path in a NEB optimization. The figure is taken from Ref. [553]

This can be achieved by using any of the three optimization methods implemented (keyword: *opt\_method*): velocity projection optimization (VPO)[545], fast inertial relaxation engine (FIRE)[554] and L-BFGS[555]. VPO and FIRE are more robust for regions that are far from the MEP, while L-BFGS converges faster when the images are close to the MEP. FIRE and VPO both have a local and global implementation (keyword: *local*). In the local variant, all of the images are treated individually when taking an optimization step, while in the latter the whole band is treated as a single point. A constant ‘trust-radius’ is used for all optimization methods, where if the magnitude of the maximum Cartesian component of an optimization step exceeds a user-supplied threshold (keyword: *maxmove*), the whole displacement is scaled down accordingly. The number of steps stored in the L-BFGS optimization for the construction of the approximate Hessian matrix can be adjusted by the user (keyword: *lbfgs\_memory*). For a more conservative optimization with L-BFGS, the memory can also be erased if a large step is attempted (keyword: *lbfgs\_restart\_on\_maxmove*).

The convergence of the intermediate images is gauged from the maximum Cartesian component of the perpendicular atom force as well as the root-mean-square, i.e.  $\max(|\mathbf{F}^{\perp}|)$ ,  $\text{RMS}(\mathbf{F}^{\perp})$ . The atomic force on the images perpendicular to the path vanishes as the images are located on the MEP. A typical value of the tolerance for the maximum component of the atomic force perpendicular to the path is  $1 \cdot 10^{-3}$  Eh/Bohr. Typically the tolerance for the root-mean-square value is chosen to be smaller by a factor of 1/2 or 1/3. Sometimes a tighter tolerance for the maximum component of the force is needed, for example  $5 \cdot 10^{-4}$  Eh/Bohr or even  $2 \cdot 10^{-4}$  Eh/Bohr. The maximum number of optimization steps allowed is set by the keyword ‘maxiter’.

The configuration of each image after each iteration is written to a ‘\_trj.xyz’ file (see file: *basename\_MEP\_ALL\_trj.xyz*). This file is useful for troubleshooting non-convergent calculations.

### 4.6.3 Climbing image NEB

In order to make the highest energy image converge more accurately to the (highest) energy maximum along the MEP, the climbing image variant of the NEB method (CI-NEB) can be used. [549] In the CI-NEB method, the effective force  $F^{\text{NEB}}$  acting on the climbing image (i.e.  $i = \text{ci}$ ) is transformed to:

$$F_{\text{ci}}^{\text{NEB}} = F_{\text{ci}} - 2 (F_{\text{ci}} \cdot \hat{\tau}_{\text{ci}}) \hat{\tau}_{\text{ci}}$$

where the climbing image is pushed up-hill along the path and relaxed down-hill perpendicular to the path. That is, the energy is maximized with respect to one degree of freedom corresponding to the direction of the tangent while the energy is minimized with respect to all other degrees of freedom. The effective force on the climbing image does not include any spring force and the density of images then becomes different on either side of the climbing image. As long as the tangent estimate is accurate enough the climbing image will converge rigorously to the point of highest energy along the path. An illustration of how the climbing image NEB method works is shown in Fig. 4.3 for a simple two-dimensional energy function.

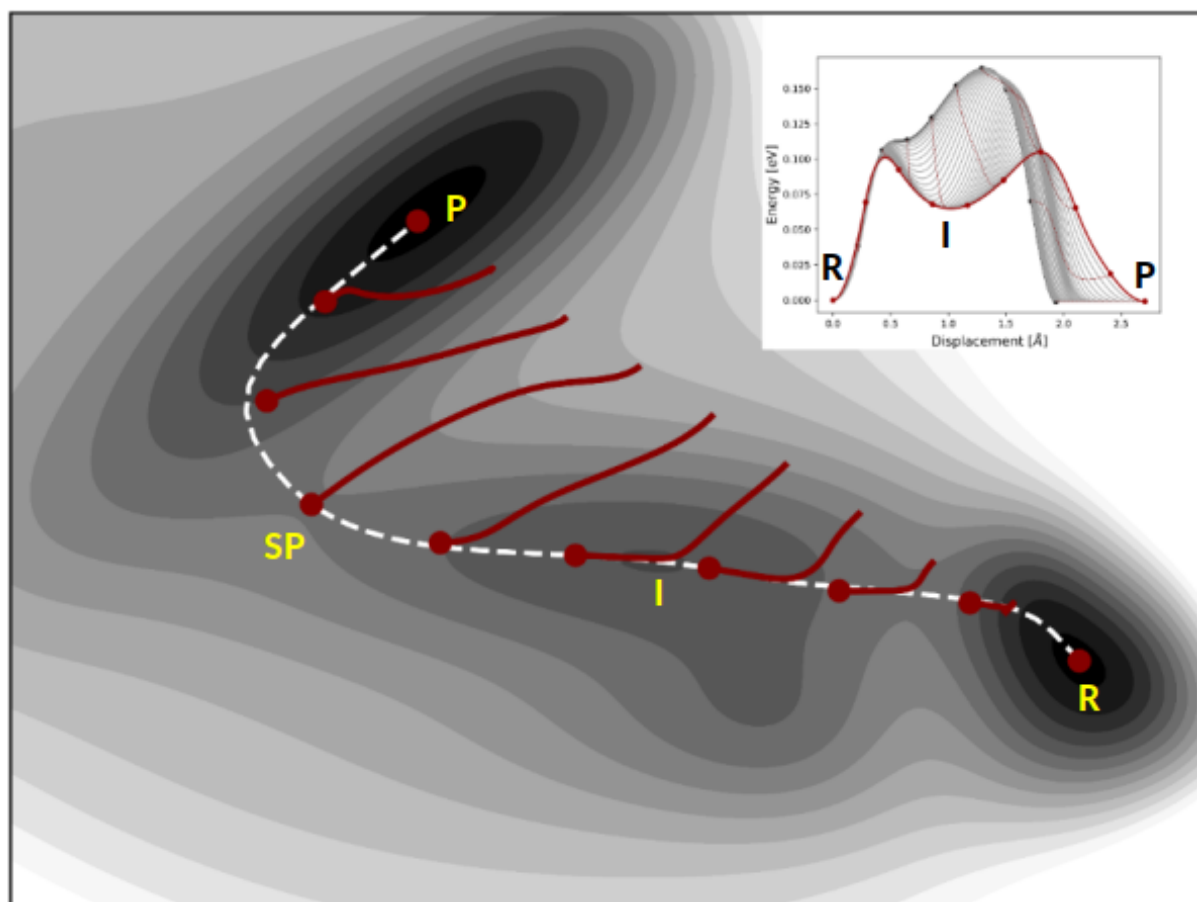


Fig. 4.3: Illustration of how the CI-NEB method works on a two-dimensional Müller-Brown energy surface,  $E(x, y)$ . [556] The calculation is started from a linear interpolation between the reactant (**R**) and product (**P**) energy minima, using  $M = 10$ . The images are displaced in the orthogonal direction to the path (red curves), until they converge to the minimum energy path (white dashed curve). The climbing image accurately locates the higher energy first-order saddle point along the path (denoted by **SP**). The optimization profile is shown as an inset. In such a plot the interpolated energy along the path is plotted as a function of displacement, for each (or selected) optimization step. The figure is taken from Ref. [553]

It can be useful to start the climbing image after the magnitude of the atomic forces perpendicular to the path drops below a given user supplied threshold (keyword: `tol_turn_on_ci`). Then, the highest energy image along the current path is converted to a climbing image. It is usually most efficient to initiate the climbing image early on or even from the start of the NEB calculation. This applies when using the VPO optimization method. For L-BFGS it is recommended to start CI-NEB when the path has partially converged to the MEP, e.g., around 0.01-0.02 Eh/Bohr.

If there are two or more maxima in the energy along the MEP it is possible that the image near the highest maximum is not chosen as the climbing image at an early stage of the NEB calculation. Then, later on the choice of the climbing image can be switched automatically. Also, for barrierless reactions, the climbing image is not turned on. The atom coordinates of the climbing image (in a CI-NEB calculation) or the highest energy image (in an NEB calculation) are written to files ‘\_NEB-CI\_converged.xyz’ and ‘\_NEB-HEI\_converged.xyz’, respectively, when a calculation has successfully completed.

The convergence of a CI-NEB calculation can either be gauged by monitoring the forces on all images or only on the climbing image, (keyword: *convtype*). The latter option may be used when the objective of the calculation is to locate the highest energy saddle point connecting the reactant and product states and can save significant computational effort. To gauge for convergence on the climbing image, both the root-mean-square and magnitude of the maximum (Cartesian) component of the atom force are monitored, i.e.  $\max(|\mathbf{F}_{ci}|)$ ,  $\text{RMS}(\mathbf{F}_{ci})$ . When gauging the convergence of all images in a CI-NEB calculation it is typically acceptable to converge the regular images more loosely than the climbing image. By default, the tolerance for the regular images is a factor of 10 larger than that of the climbing image. This scaling of the tolerances is a parameter that can be set by the user (keyword: *tol\_scale*). Typically for an acceptable convergence to the saddle point, the tolerance threshold for the maximum magnitude of an atomic force component of the climbing image can be set to  $5 \cdot 10^{-4}$  Eh/Bohr.

#### 4.6.4 Generation of the initial path

One of the most important aspects of any NEB or CI-NEB calculation is the generation of the initial path between the reactant and product energy minima (keyword: *interpolation*). The recommended method is the image-dependent pair potential (IDPP) method [557]. An alternative and simpler method is linear interpolation in Cartesian coordinates. In either case, the user should always inspect the initial path (see file: *basename\_initial\_path\_trj.xyz*) to make certain that it is acceptable.

By default, the endpoint structures are scanned for covalently linked fragments. If two fragments are detected in either of the endpoint structures, the structure is automatically prepared such that the distance between the fragments is no larger than a maximum distance defaulted to 3.5 Ångström.

The linear interpolation in Cartesian coordinates may result in overlap of atoms leading to large, initial, atomic forces and even divergence in the SCF cycles. The IDPP method solves these problems by interpolating pairwise distances between neighboring atoms in the reactant and product energy minimum. Then, a path is generated to match those distances as closely as possible. Since there are many more pairwise distances than atom coordinates, the initial path is found by minimizing the objective function,

$$S_{\kappa}^{\text{IDPP}}(r_i) = \sum_A \sum_{B>A}^M w(d_{AB}) (d_{AB}^{\kappa} - d_{AB})^2 \quad (4.24)$$

where  $d_{AB}$  is the pairwise distance between atoms  $A$  and  $B$  for intermediate image  $i$ .  $d_{AB}^{\kappa}$  is the ideal interpolated distance between atoms  $A$  and  $B$  of the same image.  $w$  is a weight function to give shorter bond distances more weight and make unnecessary bond-breaking unfavorable. The weight function is given as  $w = (d_{AB})^{-4}$ . The IDPP path avoids the overlap of atoms and can also generate a path that is closer to the MEP than the linear interpolation in Cartesian coordinates [557]. The IDPP path is obtained from an NEB calculation using the IDPP objective function starting from a linear interpolation of the Cartesian coordinates between the endpoint structures, but this calculation requires little computational effort since it does not require any electronic structure computations. Note that it is possible that the initial path breaks covalent bonds and that it can be far from the optimal MEP. If this is the case, the default algorithm detects it and automatically regenerates a different IDPP initial path. This is repeated until either an initial path without unnecessary bond changes is obtained or all initial path options are exhausted, at which point the best initial path is identified. In any case, the user should always inspect the initial path before starting an NEB calculation. The user can adjust the settings of the IDPP calculations using the ‘idpp’ related keywords, but the default values should suffice for most applications. Note that the units of the IDPP are in Ångströms instead of atomic units. The matrix of ideal interpolated distance between the atoms  $A$  and  $B$  at image  $i$ ,  $d_{AB, i}^{\kappa}$ , is most simply obtained by linear interpolation as

$$d_{AB, i}^{\kappa} = d_{AB}^{\text{initial}} + P_i * (d_{AB}^{\text{final}} - d_{AB}^{\text{initial}}),$$

where  $P_i = \frac{i}{n_{im}-1}$  is the position of image  $i$  on the path between 0, corresponding to the initial image, and 1, corresponding to the final image, and  $n_{im}$  is the total number of images.

As this linear distance interpolation may not be ideal, one can alternatively choose a bilinear distance interpolation scheme that takes atomic bonds into account. Bonds between the atoms  $A$  and  $B$  in the initial and final images are identified if their distance  $d_{AB} < t_{bond} = t_c (R_A^{cov} + R_B^{cov})$ , where  $t_c$  is an input factor impacting at what distances a bond is detected by default taken to be 1.3, and  $R_X^{cov}$  is the covalent radius of atom  $X$ . If a bond is detected in one of the endpoint structures but not the other, two linear distance interpolations are performed to obtain  $d_{AB,i}^\kappa$ . The first interpolates the distance between the endpoint structure in which the atoms are bonded and  $t_{bond}$ , the second interpolates the distance between  $t_{bond}$  and the endpoint structure in which the bond is broken. The two linear distance interpolations are joined at the path position  $P_{join}$  defaulted to 0.5, corresponding to the midpoint on the path between the endpoint images. If the atoms are bonded in the initial structure but not in the final structure, the interpolation is

$$d_{AB,i}^\kappa = \begin{cases} d_{AB}^{initial} + \frac{P_i(t_{bond} - d_{AB}^{initial})}{P_{join}}, & \text{if } P_i < P_{join} \\ d_{AB}^{final} - \frac{(1-P_i)(d_{AB}^{final} - t_{bond})}{(1.0-P_{join})}, & \text{if } P_i \geq P_{join} \end{cases}$$

If the atoms are bonded in the final structure but not in the initial structure, the interpolation is

$$d_{AB,i}^\kappa = \begin{cases} d_{AB}^{initial} - \frac{P_i(d_{AB}^{initial} - t_{bond})}{1-P_{join}}, & \text{if } P_i < 1 - P_{join} \\ d_{AB}^{final} + \frac{(1-P_i)(t_{bond} - d_{AB}^{final})}{P_{join}}, & \text{if } P_i \geq 1 - P_{join} \end{cases}$$

This bilinear interpolation leads to a slower increase of the atom distance in the bonded region compared to the non-bonded region in most cases, emphasizing the bond-breaking process and thus the region where the energy is changed most by a change in the bond distance. If the change in the bond distance in the non-bonded region is slower than the change in the bonded region, the implementation falls back to the regular linear distance interpolation.

In challenging cases, even the IDPP NEB calculation starting from a linear interpolation of the Cartesian coordinates between the endpoint structures may not provide a reasonable initial path. This may happen when the linear interpolation path has atoms come very close to each other in intermediate images, which leads to bond breaking even after NEB optimization using the IDPP objective function. In such cases, a reasonable initial path can often be obtained by avoiding the linear interpolation by sequentially adding images to the path starting from both endpoint structures instead. This method is referred to as sequential IDPP (S-IDPP).<sup>[558]</sup> The two sets of images close to each endpoint structure are separated by a larger distance than the images in each set by choosing a smaller spring constant than between the images in each set where equal spring constants are used. The images closest to the center are converged in an NEB calculation and an image added at an appropriate distance from the converged image between the two images closest to the center. This process is repeated until the requested amount of images has been added. The tangent of the images closest to the center follows the improved tangent definition and always treats them as extrema on the path, i.e. an average of the normalized distance vectors to the adjacent images is used. The number of images used to form the S-IDPP initial path can be different from the number of images used in the subsequent NEB calculation involving electronic structure calculations. It can be beneficial to use an even number of images in the S-IDPP computation since no image is placed at the center of the initial path then and moving to the requested odd number of images in an additional IDPP NEB calculation afterwards. This option is used by default, but can be deactivated. For systems involving very heavy rotations of large groups, the method becomes more robust when twice the amount of images is used in the S-IDPP calculation. Half the images are then removed automatically for the subsequent NEB calculation involving electronic structure calculations. This option is available, but not used by default.

The initial path generated with the chosen settings is checked for unnecessary bond breaking or bond formation. Bonds between the atoms  $A$  and  $B$  are identified if their distance  $d_{AB} < t_{bond} = t_c (R_A^{cov} + R_B^{cov})$ , where  $t_c$  is an input factor impacting at what distances a bond is detected by default taken to be 1.3, and  $R_X^{cov}$  is the covalent radius of atom  $X$ . A change in the connectivity of atoms  $A$  and  $B$  is considered unnecessary if their connectivity does not change between the endpoint structures. If it does change between the endpoint structures, the connectivity is allowed to change exactly once, but any further connectivity changes along the initial path are considered unnecessary. If any unnecessary connectivity changes occur on the initial path, the flexibility of S-IDPP is used to generate more initial paths with different settings systematically, until one without unnecessary connectivity changes is obtained. The parameter space for this hyperparameter search consists of all combinations of the settings *SIDPP\_Reparam*, *SIDPP\_Energy\_Weighted\_Tangent*, *SIDPP\_Even\_NIm*, *SIDPP\_Double\_NIm*, *SIDPP\_Ideal\_Springconst*, and four values (0.05, 0.01, 0.005, and 0.001) for *SIDPP\_Tol\_MaxF*, leading to a total of 128 options in addition to the chosen settings. Once an initial path without unnecessary connectivity changes is found, the hyperparameter search is terminated. In the unlikely scenario that none of the generated initial paths show no unnecessary connectivity changes, the initial path with the fewest is taken by default, but the user may prefer to have the calculation abort to inspect the initial path, which may be set with the keyword *SIDPP\_HyperSearch\_StrictConnectivity*.