

Struktur des Inputfiles

Groß-Klein-Schreibung spielt bei der Erstellung des Inputfiles keine Rolle!

```
% Link0 Commands  (Programmsteuerung)
# Route Section   (Was soll berechnet werden?)
```

Titel

Ladung Multiplizität
Geometrie

Weitere Daten (abhängig von Angaben in der Route-Section)

Link0 Commands

```
%chk=file.chk      Name für das Checkpointfile; Bsp: „file.chk“
%mem=800mb         Speicher pro Prozessor; Bsp: 800mb
%nprocs=4          Anzahl Prozessoren; Bsp: 4
%kjob=L301         Job nach Link301 beenden; nur für Basissatzinfo!
```

Optimierung

```
# Methode          Theorie-Niveau und Basissatz
opt=GDIIS          Optimierung; Zusatz GDIIS für schnellere Konvergenz
pop=none           keine Populationsanalyse ausgeben
freq              nach beendeter Geometrieoptimierung Frequenzrechnung durchführen
```

Parameter fixieren und variieren

```
# Opt=ModRed
```

Angaben im Inputfile **hinter** der Geometrie:

```
N1 N2 [N3 [N4]] [Wert] F          Parameter fixieren (freeze)
N1 N2 [N3 [N4]] [Wert] S [Anzahl] [Schrittgröße]  Parameter variieren (scan)
```

Neustart

```
opt=restart
```

- Optimierungsverlauf und letzte Geometrie werden aus Checkpoint-File gelesen
- Link0 Commands und Route-Section entsprechend dem neuen Inputfile
- Vorsicht bei Keyword CalcFC: darf hier nicht verwendet werden!

NBO Analyse

(vollständiges Inputfile; hier keine Geometrie nötig!)

```
%mem=256MB        (im Normalfall ausreichend)
%chk=file.chk     (Datei mit erfolgreicher Optimierung)
# Methode         (sollte dieselbe wie bei Optimierung sein)
Pop=NBOread      (liest NBO-Parameter am Ende ein)
Geom=AllCheck    (liest Geometrie und Titel)
Guess=TCheck     (liest Wellenfunktion)
```

```
$nbo reson bndidx $end (stark delokalisierte Struktur erlaubt, Bindungsindizes)
```

TS-Optimierung

opt=(TS,CalcFC,NoEigentest,GDIIS)

TS *sucht nach Sattelpunkt 1. Ordnung*
CalcFC *für korrekte Hess-Matrix mit Frequenzrechnung beginnen*
NoEigentest *unterdrückt Test auf richtige Anzahl negativer Eigenwerte*

opt=(QST2, CalcFC)

- *Sucht TS, der zwei Strukturen verbindet.*
- *Startgeometrie ist arithmetisches Mittel*
- *Atomreihenfolge muß in beiden Strukturen gleich sein*

opt=(QST3, CalcFC)

- *Wie QST2*
- *Startgeometrie ist dritte Struktur*

Struktur des Inputfiles für QST: Titel

 Ladung Multiplizität
 Geometrie Edukt

 Titel2

 Ladung Multiplizität
 Geometrie Produkt

 Titel3 *(nur bei QST3!)*

 Ladung Multiplizität
 Geometrie Guess

Reduktion des numerischen Rauschens

Opt=Tight *Verbesserte Geometrie*
Int(Grid=UltraFine) *Verbesserter Grid für SCF und CPHF*
impliziert CPHF(Grid=SG1)

Auswertung des Outputfiles

Die folgenden Befehle werden an der Unix-Konsole eingegeben. Groß-Klein-Schreibung beachten!

tail file.out *zeigt letzte 10 Zeilen des Outputfiles an*
grep -A4 Converged file.out *Optimierung schon konvergiert? (geht nur unter Linux!)*
grep 'SCF D' file.out *gibt Energien aller SCF-Cyclen aus*
grep -c 'SCF D' file.out *gibt nur die Anzahl der Optimierungsschritte aus*
grep -i zero file.out *gibt Zero Point Energy - Korrektur aus*
grep Free file.out *gibt ΔG aus*
grep imag file.out *gibt Zahl der imaginären Frequenzen aus*

Online-Referenz und Manual

<http://gaussian.icm.edu.pl/keyword.htm> *Liste der keywords*

<http://www.gaussian.com>